ESCUELA POLITÉCNICA NACIONAL

DOCTORADO EN INGENIERÍA ELÉCTRICA

(MENCIÓN EN SISTEMAS DE CONTROL)

FORMALIZACIÓN DE CONTROLADORES DIFUSOS LAMDA

TRABAJO DE TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE DOCTOR EN INGENIERÍA ELÉCTRICA

LUIS ALBERTO MORALES ESCOBAR

DIRECTOR:

DR. JOSE LISANDRO AGUILAR CASTRO

CODIRECTORES:

DR. JORGE ANDRES ROSALES ACOSTA

DR. GEOVANNY DANILO CHÁVEZ GARCÍA

Quito, julio de 2021

ESCUELA POLITÉCNICA NACIONAL

DOCTORADO EN INGENIERÍA ELÉCTRICA

(MENCIÓN EN SISTEMAS DE CONTROL)

FORMALIZATION OF FUZZY LAMDA CONTROLLERS

DOCTORAL THESIS FOR AWARDING THE DEGREE OF DOCTOR OF ELECTRICAL ENGINEERING

LUIS ALBERTO MORALES ESCOBAR

ADVISOR:

DR. JOSE LISANDRO AGUILAR CASTRO

CO-ADVISORS:

DR. JORGE ANDRES ROSALES ACOSTA

DR. GEOVANNY DANILO CHÁVEZ GARCÍA

Quito, june 2021

ESCUELA POLITÉCNICA NACIONAL

DOCTORADO EN INGENIERÍA ELÉCTRICA

(MENCIÓN EN SISTEMAS DE CONTROL)

FORMALIZACIÓN DE CONTROLADORES DIFUSOS LAMDA

LUIS ALBERTO MORALES ESCOBAR

Director: Dr. José Lisandro Aguilar Castro

Codirector: Dr. Jorge Andrés Rosales Acosta

Dr. Geovanny Danilo Chávez García

Tribunal de Defensa: Nombre 1 Apellido 1

Institución / Departamento / País

Nombre 2 Apellido 2

Institución / Departamento / País

Nombre 3 Apellido 3

Institución / Departamento / País

Catalogación:
240 páginas; 21 x 15 cm
Línea de Investigación: Control y Sistemas, Robótica y Automatización, Inteligencia Artificial
Departamento de Automatización y Control Industrial
Primera Edición. Fecha de Catalogación: Febrero/2021

FORMALIZACIÓN DE CONTROLADORES DIFUSOS LAMDA

LUIS ALBERTO MORALES ESCOBAR

Tesis De Doctorado
Programa de Doctorado en Ingeniería Eléctrica
Mención: Sistemas de Control

ISBN: XXX-XXX-XXX-X
eISBN: XXX-XXX-XXX-XXX-X

ESCUELA POLITÉCNICA NACIONAL, 2021

AVAL

Certificamos que el presente trabajo fue desarrollado por Luis Alberto Morales Escobar, bajo nuestra supervisión.

_____

DR. JOSE LISANDRO AGUILAR CASTRO
DIRECTOR DEL TRABAJO DE TITULACIÓN

_____

DR. JORGE ANDRES ROSALES ACOSTA
CODIRECTOR DEL TRABAJO DE TITULACIÓN

_____

DR. GEOVANNY DANILO CHÁVEZ GARCÍA
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Luis Alberto Morales Escobar declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado, posgrado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Luis Alberto Morales Escobar

DEDICATORIA

Dedico este trabajo a mi esposa Tamara, a mis padres Luis y Mary, a mis hermanas Sandra y Lourdes, a mis sobrinos Cami y Nico y a mi cuñado Eddy, por motivarme e inspirarme a cumplir esta nueva etapa académica.

AGRADECIMIENTO

A Dios, a mi familia, profesores, colegas y amigos que han aportado a la elaboración de este trabajo, a todos ellos gracias por siempre estar presentes.

De manera muy especial agradezco al Profesor Jose Aguilar ya que gracias a sus conocimientos, apoyo y guía se ha podido concluir este trabajo de manera satisfactoria.

CONTENTS

AVAL i
DECLARACIÓN DE AUTORÍA ii
DEDICATORIA iii
AGRADECIMIENTO iv
CONTENTS v
LIST OF FIGURES ix
LIST OF TABLES xv
LIST OF MAIN ABBREVIATIONS xvii
ABSTRACT xix
RESUMEN xxi
1. INTRODUCTION 1
1.1 Research motivation and justification 5
1.2 Hyphotesis 6
1.3 Main and Supplementary Objectives 6
1.3.1 Main Objective 6
1.3.2 Supplementary Objectives 6
1.4 Scopes of the research 7
1.5 Main Contributions 9
1.6 List of Publications 10
1.6.1 Journal Papers 10
1.6.2 Conference Proceddings 12
1.6.3 Papers during the doctoral schooling 12
1.7 Thesis Outline 14
2. THEORETICAL FRAMEWORK 16
2.1 State of the Art of LAMDA 17

2.2 LAMDA fundamentals 21
2.2.1 Marginal Adequacy Degree (MAD) 22
2.2.1.1 Fuzzy Binomial Function 22
2.2.1.2 Gaussian Function 23
2.2.2 Global Adequacy Degree (GAD) 23
2.3 Artificial Intelligence in the field of Control Systems ... 26
2.4 Z-Numbers 28
2.4.1 Z-numbers formalization 30
2.4.2 Total Utility of Z-numbers (TU) 32
3. PROPOSED LAMDA EXTENSIONS 34
3.1 Extension of LAMDA in the classification context 34
3.1.1 Extension 1: Adaptable $\boldsymbol{GADNIC}$ 38
3.1.2 Extension 2: Higher Adequacy Degree (HAD) .. 40
3.1.3 General procedure of LAMDA-HAD 42
3.2 Extension of LAMDA in the clustering context 43
3.2.1 Robust Distance 44
3.2.2 Automatic merge algorithm 46
3.2.3 General procedure of LAMDA-RD 51
4. LAMDA IN THE FIELD OF CONTROL SYSTEMS 54
4.1 Rule-based LAMDA 57
4.2 LAMDA Sliding-Mode Control (LSMC) 59
4.2.1 System description and fundamentals of SMC .. 59
4.2.2 The proposed approach of LSMC 62
4.2.3 Stability Analysis 69
4.2.4 LSMC scaling gains offline calibration 72
4.2.4.1 Heuristic calibration 72

4.2.4.2 Particle Swarm Optimization (PSO) 74
4.3 Adaptive LAMDA 77
4.3.1 Adaptive LAMDA Model 78
4.3.1.1 Hybrid Learning Algorithm 81
4.3.1.2 Adaptive LAMDA Control 87
4.3.1.3 Feedback Control with Adaptive LAMDA 89
4.3.1.4 Convergence of the learning algorithm 90
4.4 LSMC based on Z-numbers (ZLSMC) 92
4.4.1 Formalization of ZLSMC 94
5. EXPERIMENTATION AND RESULTS 103
5.1 LAMDA in classification and clustering tests 103
5.1.1 LAMDA-HAD in classification process 103
5.1.1.1 LAMDA-HAD validation 103
5.1.1.2 Testing algorithms for the identification of new classes 110
5.1.2 LAMDA-RD in clustering process 113
5.1.2.1 Comparison of LAMDA-RD with other clustering algorithms 115
5.1.2.2 Performance comparison of LAMDA-RD and other online clustering algorithms 123
5.2 Tests of LAMDA as controller 128
5.2.1 LSMC experiments 129
5.2.1.1 Case Study 1 130
5.2.1.2 Case Study 2 139
5.2.1.2.1 Sensitivity Analysis 148
5.2.1.3 Comparative Analysis of LSMC and ZLSMC 151

5.2.1.3.1 CSTR Process 151
5.2.1.3.2 Mixing Tank Process 153
5.2.2 Adaptive LAMDA experiments 154
5.2.2.1 Case Study 1 156
5.2.2.2 Case Study 2 169
5.2.2.3 Case Study 3 175
5.2.2.3.1 Robot Model 175
5.2.2.3.2 Kinematic Controller 177
5.2.2.3.3 Dynamic Controller 178
5.2.3 General comparative analysis among LAMDA controllers 189
5.2.3.1 Rule-based LAMDA applied to the dynamic model 189
5.2.3.2 LSMC and ZLSMC applied to the dynamic model 192
5.3 Computational Complexity 204
5.3.1 Memory Usage 204
5.3.2 Computation Time 205
*NT corresponds to not tested. 206
5.3.3 Number of Operations 206
6. DISCUSSION 209
6.1 Classification Context 209
6.2 Clustering Context 211
6.3 Control Systems Context 213
7. CONCLUSIONS AND FUTURE WORK 217
7.1 Conclusions 217
7.2 Future Work 220

8. REFERENCES 222

APPENDIX A: Graphical representation of the datasets used in classification and clustering A-1

APPENDIX B: LAMDA-HAD EVALUATION B-1

B.1. Comparison between LAMDA, LAMDA (Adaptable $\boldsymbol{GADNIC}$) and LAMDA-HAD in classification benchmarks B-1

B.2. ROC CURVES OF THE TESTED CLASSIFIERS B-3

APPENDIX C: LAMDA-RD parameter calibration C-1

APPENDIX D: Computational Complexity of LAMDA-HAD and LAMDA-RD D-1

APPENDIX E: Demonstration of the stability of the learning algorithm E-1

LIST OF FIGURES

Figure 2.1. Structure of LAMDA algorithm 25

Figure 2.2. Membership functions of the Z-number $Z = (Az, Rz)$ with $Az = Gauss(0.1,0.25)$, $Rz = Gauss(0.6,0.07)$ 31

Figure 3.1. Iris Benchmark data and classes 35

Figure 3.2. GADs in each class for Iris dataset 36

Figure 3.3. Classification results of the original LAMDA algorithm 37

Figure 3.4. $MGAD$ obtained for each GAD in the different classes 39

Figure 3.5. Mean value $GADNIC$ for each class in the Iris dataset 40

Figure 3.6. LAMDA-HAD structure 42

Figure 3.7. New sample assigned to (a) new cluster, (b) pre-existing cluster. 47

Figure 3.8. Graphical example to assign a new sample to a cluster, when Proposition 3 is not met, (a) the sample creates a new cluster, (b) the sample is assigned to a pre-existing cluster. 49

Figure 3.9. Graphical example to assign a new sample to a cluster, if Proposition 3 is met, (a) the algorithm merges the individual to the neighbor cluster, (b) the algorithm merges the cluster where the individual was assigned with the neighbor cluster 51

Figure 3.10. LAMDA-RD algorithm 52

Figure 4.1. Scheme of the proposed LAMDA controller 59

Figure 4.2. Block diagram of the LSMC. 69

Figure 4.3. Offline optimization of the scaling gains scheme 74

Figure 4.4. Offline optimization of the scaling gains scheme 76

Figure 4.5. Offline optimization of the scaling gains scheme 76

Figure 4.6. Adaptive scheme for LAMDA 79

Figure 4.7. Hybrid Learning Scheme 82

Figure 4.8. (a) Block diagram of the training phase of the inverse control method, (b) Block diagram of the application phase of the inverse control method 88

Figure 4.9. Block diagram of the online inverse learning control with Adaptive LAMDA 89

Figure 4.10. Membership functions for: a) MADs of st, b) reliability of $s(t)$ 97

Figure 4.11. a) Membership functions for: a) MADs of st, b) reliability of $s(t)$, c) MADs of st, d) reliability of $s(t)$ 100

Figure 4.12. Block diagram of the ZLMC 102

Figure 5.1. Database partitioning for new classes identification 110

Figure 5.2. Classification results of new classes identification..112

Figure 5.3. Methodology used for the comparison of LAMDA-RD with other approaches 116

Figure 5.4. Methodology used for the comparison of the different online clustering algorithms 125

Figure 5.5. Tests performed with s1 dataset; (a) Original partition, clusters generated by (b) LAMDA-RD (15 clusters), (c) LAMDA-TP (15 clusters), and (d) ADDclustering (4 clusters), see the detailed statistics in Table 5.9. 128

Figure 5.6. Studied process (CSTR) 130

Figure 5.7. Classes and rules for uc based on $s(t)$ for the CSTR 134

Figure 5.8. Classes and rules for ud based on $s(t)$ and $s(t)$for the CSTR 135

Figure 5.9. (a) Comparative outlet temperature of the CSTR, (b) applied control actions 137

Figure 5.10. Comparative values of performance indexes of the controllers applied to the mixing tank 137

Figure 5.11. (a) Comparative outlet temperature of the CSTR under disturbances, (b) applied control actions 139

Figure 5.12. Studied process (Mixing Tank) 140

Figure 5.13. K, τ, and $t0$ variations as $m(t)$ function for the mixing tank 143

Figure 5.14. Classes and rules for discontinuous control action ud based on $s(t)$ and $s(t)$ for the mixing tank 144

Figure 5.15. Classes and rules for continuous control action uc based on $s(t)$ for the mixing tank 144

Figure 5.16. (a) Change of $W1$, (b) change of dead time $t0$ 146

Figure 5.17. (a) Comparative outlet temperature of the mixing tank, (b) applied control actions 146

Figure 5.18. Comparative values of performance indexes of the controllers applied to the mixing tank 148

Figure 5.19. Controllers with different number of classes (a) outlet temperature of the process (b) applied control actions 150

Figure 5.20. (a) Comparative outlet temperature of the CSTR, (b) applied control actions of LSMC and ZLSMC 152

Figure 5.21. a) Comparative outlet temperature of the CSTR under disturbances, (b) applied control actions of LSMC and ZLSMC 152

Figure 5.22. a) Comparative outlet temperature of the mixing tank, (b) applied control actions of LSMC and ZLSMC 153

Figure 5.23. Implementation scheme of the online learning adaptive controllers.....155

Figure 5.24. Block diagram of a simple HVAC system 156

Figure 5.25. Adaptive control structure for the HVAC system ...161

Figure 5.26. (a) Heat disturbance signal, (b) Moisture disturbance signal applied to robustness analysis 163

Figure 5.27. Comparative results with temperature disturbance: (a) control action $u1$, (b) Humidity Ratio $W3$ 163

Figure 5.28. Comparative results with temperature disturbance: (a) control action $u2$, (b) Temperature $T3$ 164

Figure 5.29. Comparative results with moisture disturbance: (a) control action $u1$, (b) Humidity Ratio $W3$.....166

Figure 5.30. Comparative results with relative humidity disturbance: (a) control action $u2$, (b) Temperature $T3$.....166

Figure 5.31. Comparison of learning algorithms a) adjust in the training stage with actual output data b) error.....171

Figure 5.32. Adaptive control structure for the Mixing Tank.....172

Figure 5.33. (a) Comparative response of the system (temperature) (b) Applied control actions 174

Figure 5.34. Parameters of the unicycle-like mobile robot 176

Figure 5.35. Comparison of learning algorithms in mobile robot (a) linear velocity, (b) angular velocity 179

Figure 5.36. Adaptive control structure for a mobile robot.....180

Figure 5.37. V-REP scene showing the Pioneer 3DX robot 181

Figure 5.38. (a) Circular trajectory followed by the robot, (b) instantaneous quadratic error of the robot position, speeds of the robot and control actions (c) linear and (d) angular velocity 184

Figure 5.39. (a) Lenmiscate trajectory followed by the robot, (b) instantaneous quadratic error of the robot position, speeds of the robot and control actions (c) linear and (d) angular velocity 185

Figure 5.40. (a) Square trajectory followed by the mobile robot, (b) instantaneous quadratic error of the robot position, speeds of the robot and control actions (c) linear and (d) angular velocity 186

Figure 5.41. Step response (a) Linear and, (b) Angular velocity 190

Figure 5.42. Defined classes and outputs for the linear and angular velocities 191

Figure 5.43. Control scheme for trajectory tracking of a mobile robot using LAMDA-PID 192

Figure 5.44. Classes and rules for continuous control action $\boldsymbol{uc}$ based on $\boldsymbol{s(t)}$ for the linear velocity of the mobile robot.....194

Figure 5.45. Classes and rules for discontinuous control action ud based on $s(t)$ and $s(t)$ for the linear velocity of the mobile robot.....194

Figure 5.46. Adaptive LAMDA structure for a mobile robot 195

Figure 5.47. Linear velocity for the Lenmiscate trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA 197

Figure 5.48. Angular velocity for the Lenmiscate trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA 197

Figure 5.49. (a) Lenmiscate trajectory performed by the different methods, (b) trajectory distance error 198

Figure 5.50. Linear velocity for the Square trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA 199

Figure 5.51. Angular velocity for the Square trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA 200

Figure 5.52. (a) Lenmiscate trajectory performed by the different methods, (b) trajectory distance error 200

Figure 5.53. Quantitative comparison (a) IAE, (b) ISE 202

Figure 5.54. Quantitative comparison for the different controllers based on ISCO for the: a) Linear velocity, b) Angular velocity ..203

Figure A.1. Two-dimensional graphs of the classification datasets using the t-Distributed Stochastic Neighbor Embedding A-2

Figure A.2. Two-dimensional graphs of the clustering datasets using the t-Distributed Stochastic Neighbor Embedding A-4

Figure B.1. Comparison between LAMDA, LAMDA (Adaptable $GADNIC$) and LAMDA-HAD in classification benchmarks B-3

Figure B.2. Comparison of the micro-average ROC for the datasets with different classifiers: B-5

Figure C.1. Obtained results for $R15$, in function of dnb and Dt: (a) PC. (b) top view of PC C-5

Figure C.2. Obtained results for $R15$, as function of dnb and Dt: (a) number of clusters "m". (b) top view of the number of clusters C-5

Figure C.3. Illustrative example of: (a) low value of Dt. (b) high value of Dt C-6

Figure C.4. Recommended calibration of $Dt\ and\ dnb$ C-7

Figure C.5. Performance Coefficient for different values of dnb and Dt (marked in red the best value) C-8

LIST OF TABLES

Table 2.1. Aggregation functions for LAMDA [64] 24

Table 4.1. Rule table of LSMC for $s(t)$.....65

Table 4.2. Rule table of LSMC for $s(t)$ and $s(t)$ with $bX(t), t > 0$ 67

Table 4.3. Rule table of LSMC for $s(t)$ and $s(t)$ with $bX(t), t < 0$ 68

Table 4.4. Rule table of ZLSMC for $s(t)$ 97

Table 4.5. Rule table of ZLSMC for $s(t)$ and $s(t)$, $bX(t), t > 0$ .100

Table 5.1. Datasets used to validate LAMDA-HAD.....104

Table 5.2. Average F-measure (%) of LAMDA-HAD and other classification algorithms.....107

Table 5.3. Average accuracy (%) of LAMDA-HAD and other classification algorithms.....108

Table 5.4. Relative error between the highest value metrics and LAMDA-HAD metrics.....109

Table 5.5. Datasets used to test the clustering algorithms [107] 114

Table 5.6. SC for different clustering algorithms 117

Table 5.7. $WBindex$ for different clustering algorithms 118

Table 5.8. PC metric for different clustering algorithms.....120

Table 5.9. RI metric for different clustering algorithms 122

Table 5.10. Relative error between the highest RI and LAMDA-RD metrics.....123

Table 5.11. Performance metrics of online clustering algorithms 126

Table 5.12. Steady-state values of the CSTR 132

Table 5.13. Steady-state values of the mixing tank.....142

Table 5.14. Comparative ISE values among LSMC and ZLSMC 154

Table 5.15. Numerical values for system parameters.....158

Table 5.16. Numerical vales for system parameters at the operating point 159

Table 5.17. Numerical values for IAE for the HVAC experiments 169

Table 5.18. IAE of the controllers applied to the mixing tank process.....174

Table 5.19. Numerical values for IAE for the mobile robot experiments.....188

Table 5.20. Computation Time (seconds) of LSMC ($l = 2$) 206

Table 5.21. Computation Time in seconds (s) of Adaptive LAMDA 206

Table 5.22. Arithmetic complexity of the fuzzy control algorithms 207

Table 5.23. Arithmetic complexity in terms of Asymptotic Notation 207

Table B.1. Average AUC and Standard Deviation of LAMDA-HAD and other classification algorithms..... B-6

Table D.1. Arithmetic complexity (number of operations) of LAMDA, LAMDA-HAD and LAMDA-RD D-2

Table D.2. Arithmetic complexity in terms of Big-O of classification algorithms.....D-3

Table D.3. Arithmetic complexity in terms of Big-O of clustering algorithms.....D-4

LIST OF MAIN ABBREVIATIONS

ADD	Autonomous Data-Driven
AHT	Agglomerative Hierarchical Tree
AIC	Adaptive Inverse Control
ANFIS	Adaptive Neuro-Fuzzy Inference Systems
ANN	Artificial Neural Networks
CMAD	Cauchy Marginal Adequacy Degree
CSTR	Continuous Stirred Tank Reactor
DBSCAN	Density Based Spatial Clustering
DNB	Distance between Neighbors
DT	Density Threshold
DT	Decision Trees
FCM	Fuzzy C-Means
FDI	Fault Location and Isolation
FL	Fuzzy Logic
FLC	Fuzzy Logic Control
FOPDT	First Order Plus Dead Time
GAD	Global Adequacy Degree
GD	Gradient Descent
HAD	Higher Adequacy Degree
HVAC	Heating Ventilation Air Conditioning
IAE	Integral Absolute Error
IGAD	Intuitionistic Global Adequacy Degree
ISCO	Integral Square Control Output
ISE	Integral Square Error
KM	K-Means
KMD	K-Medoids
KNN	K-Nearest Neighbors

LAMDA	Learning Algorithm for Multivariable Data Analysis
LDA	Linear Discriminant Analysis
LSMC	LAMDA Sliding Mode Control
MAD	Marginal Adequacy Degree
MGAD	Average GAD
MIMO	Multiple-Input Multiple-Output)
NBC	Naive Bayes Classifier
NGAD	Normalized GAD
NIC	Non-Informative Class
NN	Feedforward Neural Networks
PC	Performance Coefficient
PID	Proportional Integral Derivative
PSO	Particle Swarm Optimization
RD	Robust Distance
RF	Random Forest
RI	Rand Index
RLS	Recursive Least Squares
RMAD	Robust Marginal Adequacy Degree
SALSA	Situation Assessment using LAMDA
SC	Silhouette Coefficient
SISO	Single Input Single Output
SMC	Sliding Mode Control
SPC	Spectral Clustering
SSB	Sum-of-Squares Between Clusters
SSW	Sum-of-Squares Within Clusters
SVM	Support Vector Machines
TSK	Takagi Sugeno Kang
TU	Total Utility
ZLSMC	LSMC based on Z-numbers

ABSTRACT

This document presents a comprehensive analysis of the LAMDA (Learning Algorithm for Multivariable Data Analysis) method, an artificial intelligence tool that can work on supervised and unsupervised learning tasks. Initially, the study focuses on identifying the weaknesses of the algorithm to propose extensions to improve its performance. We propose LAMDA-HAD for classification and LAMDA-RD for clustering. The two extensions improve the performance of the original LAMDA since they correct the assignment of objects to the class or group that they really belong to. Additionally, in the case of unsupervised learning, the groups can be merged automatically when necessary, improving the quality of the clustering with a more efficient process of information discovery. The proposals have been formalized and validated with different benchmarks that have made it possible to study the performance of the proposed extensions and carry out a comparative analysis against other well-known algorithms.

In a second stage, we formalize LAMDA in the context of control systems taking advantage of its capacity to detect functional states of systems. LAMDA working as controller requires an initial phase of class definition corresponding to the functional states of the system (training), and a second phase of definition of fuzzy rules to obtain the control action. In addition, because it is mainly a classification algorithm, in this work we formalize an inference method based on LAMDA to compute the controller output that allows taking the system from the current to the desired functional state. To demonstrate the stability and robustness of the control algorithm, LAMDA is combined with the concept of Sliding-

Mode Control (SMC). The novelty of this proposal is that LAMDA is used to compute the continuous and discontinuous control actions of the SMC to obtain a chattering free control action. This controller, called LSMC, can be applied to SISO systems with variable dynamics and model uncertainties. Additionally, the concepts of Z-numbers have been added to the LSMC scheme to handle reliability criteria in the algorithm, in order to improve the performance of the controller.

Finally, since LSMC requires the expert knowledge for the design, a different controller-based on LAMDA is proposed, this is the adaptive LAMDA which is presented as an addition to this work. The novelty of this proposal is that for the first time LAMDA is used for fuzzy modeling and control of complex systems, which is a great advantage if the model is not available, is partially known, or variable. Adaptive LAMDA consists of a training stage to establish the initial parameters of the controller, and an application stage in which the control strategy is calculated and updated through online learning that evaluates the closed-loop system.

The proposed LAMDA controllers have been validated in different case studies, for example, in the control of industrial processes with variable dynamics or in the field of robotics for trajectory tracking control whose results have been analyzed and compared with other similar control techniques, showing that the proposed methods are capable of performing a precise control that improves the performance of the overall system.

KEYWORDS: LAMDA, Intelligent Control, Classification, Clustering, SMC.

RESUMEN

En este documento se presenta un amplio análisis del método LAMDA, herramienta de la inteligencia artificial que puede trabajar en tareas de aprendizaje supervisado y no supervisado. Inicialmente, el estudio se centra en identificar los puntos débiles del algoritmo para plantear extensiones que mejoren su desempeño. Nosotros proponemos LAMDA-HAD para clasificación y LAMDA-RD para agrupamiento. Estas dos extensiones mejoran el desempeño de LAMDA original ya que corrigen la asignación de objetos a la clase o grupo que realmente pertenecen y adicionalmente, en el caso de aprendizaje no supervisado, los grupos puedan unirse automáticamente cuando es necesario, mejorando la calidad del agrupamiento con un proceso de descubrimiento de información más eficiente. Estas propuestas han sido formalizadas y validadas con diferentes "benchmarks" que han permitido estudiar el rendimiento de las extensiones propuestas y su respectivo análisis comparativo

En una segunda etapa, nosotros formalizamos LAMDA en sistemas de control, aprovechando su capacidad de detección de estados funcionales. Para que LAMDA funcione como controlador, requiere una fase de definición de clases que son los estados funcionales del sistema y la definición de reglas difusas para obtener la acción de control. Además, al ser un algoritmo de clasificación, en este trabajo es formalizado un método de inferencia que permita calcular la salida del controlador que permita llevar el sistema del estado funcional actual al deseado. Para demostrar la estabilidad y robustez de la propuesta del algoritmo de control, LAMDA es combinado con el concepto

de Control en Modos Deslizantes (SMC). Lo novedoso de esta propuesta es que LAMDA es usado para calcular las acciones de control continua y discontinua del SMC para obtener una acción de control libre de “chattering”. Este controlador, llamado LSMC, puede ser aplicado a sistemas SISO con dinámica variable e incertidumbres en el modelado. Adicionalmente se ha añadido los conceptos de números Z al esquema LSMC para manejar criterios de confiabilidad, para mejorar el desempeño del controlador.

Finalmente, ya que LSMC requiere el conocimiento del experto para el diseño, se ha planteado un esquema de LAMDA adaptativo. Lo novedoso de esta propuesta es que por primera vez se utiliza LAMDA para modelado difuso y control de sistemas complejos, lo cual es una gran ventaja si el modelo no está disponible, es parcialmente conocido o variable. LAMDA adaptativo consta de una etapa de entrenamiento para establecer los parámetros iniciales del controlador, y una etapa de aplicación en la que se calcula y actualiza la estrategia de control mediante un aprendizaje en línea que evalúa el sistema en lazo cerrado.

Los controladores LAMDA han sido validados en diferentes casos de estudio, por ejemplo, en el control de procesos industriales de dinámica variable o en el campo de la robótica para control de seguimiento de trayectorias cuyos resultados han sido analizados y comparados con otras técnicas de control similares, demostrando que los métodos propuestos son capaces de realizar un control preciso que mejora el desempeño del sistema en general.

PALABRAS CLAVE: LAMDA, Control Inteligente, Clasificación, Agrupamiento, SMC

1. INTRODUCTION

In recent years, control theory has proposed several control strategies, as well as different methodologies, for the design of control systems based on the available information. Most control methodologies are based on a common foundation, such as having information about the behavior of the plant, either in the form of an analytical model, plant knowledge, and mathematical approximations among others.

At the end of the last century the idea of "Intelligent Control" has been proposed, which is originated with the aim of applying artificial intelligence to control systems. The main reasons why it has been necessary to design controllers based on techniques inspired by intelligent faculties are [1]:

- In process control, automation means that the control system replaces the human operator, which requires a great capacity in managing the knowledge of the process. In this sense, intelligent control offers innovative solutions, since it allows us proposing methodologies to automatically perform some tasks that are typically performed by humans, based on a priori knowledge.
- The control of systems with highly non-linear characteristics is a research field under development, in which there are requirements that cannot be met with conventional control theory. Mainly, due to the complexity that exists in modeling [2,3], the controller design requires a great effort if an analytical treatment is carried out, so a more heuristic treatment can be considered.
- Due to the existence of increasingly complex systems with a high degree of uncertainty, which generally cannot

be modeled in a strict mathematical way, the artificial intelligence through its learning methods is a very good option to obtain approximations that facilitate the design of control systems that operate under uncertain conditions.

Specifically, the design of "intelligent controllers" has been proposed with the aim of simulating the "intelligent" abilities of living beings, in particular, human reasoning [1]. To do this, controllers must include characteristics such as knowledge representation, learning, and reasoning under uncertainty.

With current computers, powerful in their functionalities, artificial intelligence techniques have been used to achieve some of these objectives, giving as initial results, intelligent autonomous controllers [4]. Thus, it is of great importance to continue with the analysis of new techniques of artificial intelligence, capable of improving the performance of the initial intelligent controllers, viable in their implementation, in order to be a useful alternative applicable to solve control problems in systems with a high degree of uncertainty.

In general, in the field of control systems, information of the process to be controlled is required. The value of the information depends on its usefulness for decision-making. For example, information that indicates that a person is sick encourages them to take a control action, in this case, take a medicine. Information that does not generate an action is meaningless, while many actions are meaningless without the underlying information to justify them [5]. In this way, the link between information and control can be clearly seen. On the other hand, the information of the changes in physical

systems can be recorded by sensors (data collection). By linking the recorded data, with the control actions, then knowledge is generated. For example, the human body temperature must be at 37 °C; if an individual has a higher temperature, then the individual is considered sick. The link to this information (the conclusion) is based on our prior knowledge. The process of extracting knowledge from the data is known as Data Mining [5].

Feldbaum's dual control concept states that control must not only lead, but must also learn [6]. Based on this, an intelligent control system is one that comprises a serie of techniques, fundamentally taken from artificial intelligence, with which it is intended to solve unapproachable control problems by classical methods [4]. For this, the control system bases its operation on cognitive activities such as learning, inferring, optimizing, reasoning and decision making, which often derive from unpredictable behaviors of the systems to be controlled. Thus, an intelligent controller, based on artificial intelligence techniques, can be autonomous, non-linear, and adaptive [3].

The main reason for the development of intelligent controllers is the existence of incomplete or inadequate representations of the plants and imprecise specifications. In particular, environments with uncertainty can make the controller, and therefore the plant, unstable, which would be extremely risky in all processes. Currently, intelligent control methodologies are very varied, and include fuzzy logic [7] , expert systems [8], artificial neural networks [9], evolutionary programming [10], among others. These methods have been used in control tasks, fault detection, plant supervision, and other areas of application.

One of the most used intelligent techniques in the field of control is fuzzy logic. Within the fuzzy control, there are two widely used applications: modeling and control. Compared to traditional control, fuzzy methods have three practical advantages [1]. Firstly, the mathematical model of the process to be controlled is not required. Second a non-linear controller developed empirically without mathematical complications can be obtained, and finally, fuzzy methods can learn about the model of the plant to compute the control action.

The machine learning method called LAMDA (Learning Algorithm for Multivariable Data Analysis) [11], is a non-iterative technique based on fuzzy logic. It has ability to create new classes after the training stage, using a threshold known as the Non-Informative Class (NIC). LAMDA has the ability to work in the context of classification and clustering [12], tasks that have mainly focused on the detection of functional states of systems. To learn, it uses probability density functions for the similarity analysis, and fuzzy aggregation functions [13] to determine the current functional state of the system.

Starting from the detection of functional system states, it is proposed a novel approach in this work to convert LAMDA into a controller. For that, it is necessary to establish an inference mechanism that allows it to take the system from a current functional state to one desired, which is defined through the descriptors that are characterized by the designer for the controller implementation. Systems with variable dynamics and model uncertainties, in which conventional controllers decrease in performance terms are the field of application of the control based on LAMDA.

1.1 Research motivation and justification

Intelligent controllers have the ability to present a good performance (in terms of energy consumption, robustness disturbance rejection and system response) in systems in which obtaining their mathematical model is complex. The complexity of the models may occur due to highly non-linear characteristics, to the scarce information available from the system, or due to uncertainties. In this sense, the development of new proposals or methodologies that allow us controlling systems with these characteristics always represents a research area with theoretical and practical projection.

Since the conception of this research work, it has been proposed to take advantage of the characteristics of LAMDA for the formalization of new controllers. In this sense, LAMDA have the advantage of learning and identifying the functional states of a system, which could be used to take a corrective action to bring the plant to a desired state. Being a fuzzy algorithm, we propose to establish a control guided by the operational states and to propose the control law that guides the system to the desired state considering uncertain information. The operation of the controller in uncertain systems has led us to explore extensions of fuzzy logic such as Z-numbers, in order to address the reliability criteria that could help to handle the uncertainties inherent to the system. With the aforementioned arguments, it is proposed to investigate and formalize the design and implementation of controllers based on LAMDA, with the ability to adapt and learn to characterize the functional states of the system, to take corrective action.

1.2 Hyphotesis

It is possible to propose an intelligent controller[1] based on artificial intelligence techniques related to pattern recognition; in this case, functional states of the system to be controlled, with the LAMDA algorithm. The controller can be designed without the need to have the plant model in detail, which can have non-linear characteristics.

1.3 Main and Supplementary Objectives

1.3.1 Main Objective

According to the above discussion, the main objective of this research work is:

"To formalize controllers based on the LAMDA fuzzy model".

1.3.2 Supplementary Objectives

- To formalize and implement extensions to the LAMDA model, which improve performance in machine learning tasks corresponding to classification.
- To formalize and implement extensions to the LAMDA model, which improve performance in machine learning tasks corresponding to clustering.
- To determine the most suitable LAMDA model for the implementation of intelligent controllers.

[1] Note that in the research plan the development of an intelligent controller based on LAMDA has been proposed, however throughout the research, several proposals of LAMDA-based controllers have been developed which are detailed in this document

- To propose a methodology for the design of LAMDA controllers, and analyze their stability, robustness and disturbance rejection.
- To optimize the LAMDA controller[2] parameters using evolutionary optimization heuristics (offline).
- To evaluate the proposed LAMDA controller[2] in processes with non-linear characteristics and contrast the results with fuzzy controllers with similar characteristics.
- To explore the feasibility of implementing Z number theory in the LAMDA controller[2] to deal with ambiguities, observing if it contributes to improving the performance of the controller and if it is applicable in terms of machine time consumption.

1.4 Scopes of the research

The research proposes the achievement of several scopes addressed by different approaches. As mentioned in the introduction, LAMDA is applied in the field of classification and clustering tasks, in which it is proposed to improve its performance with different extensions. Also, the algorithm is used in the field of control systems, establishing the bases for its design, and its formalization in terms of stability and robustness. The detailed scopes reached in this work are:

- Extensions to the LAMDA model have been developed in order to improve its performance in classification tasks.

[2] The supplementary objectives refers to a LAMDA controller, however, as mentioned above, several proposals have been developed and the methodologies described have been applied to the controller that we consider to be the most complete in terms of stability and robustness.

The formalized extensions are implemented for using as a supervised learning algorithm, and are tested in different types of benchmarks.

- Extensions to the LAMDA model have been developed, in order to improve its performance in clustering tasks. The formalized extensions are implemented for using as an unsupervised learning algorithm, and are tested in different types of benchmarks.
- A LAMDA model has been proposed for the conception of an intelligent controller.
- A methodology for the design of LAMDA controllers has been proposed, to be applied it in different systems.
- Once the LAMDA controller model has been proposed, an evolutionary optimization algorithm has been used for its calibration to improve its performance.
- Possible enhancements to the LAMDA controller have been explored to handle ambiguous contexts, based on the concept of Z-numbers.
- The complete model of the controller has been established, laying the foundations for possible future developments on it, so that it can be applied to any type of automatic system.
- The controller has been implemented and tested in non-linear processes.

1.5 Main Contributions

This work presents different proposals of LAMDA-based extensions for classification /clustering tasks and controllers, whose main contributions and novelties are the following:

- The theoretical formalization of the improvements to LAMDA in classification and clustering tasks.
- The theoretical formalization of controllers based on classes or functional states, concepts taken from the LAMDA theory, which have a number of fixed layers (intrinsic feature). Therefore, the design is more straightforward than methods where the number of internal layers must be calibrated [53]. Also, the characterization of controllers using the class criteria leads to a quick convergence to a desired output based on definition of rules or online learning.
- The design of a robust LSMC controller where the continuous and discontinuous control actions of SMC schemes are computed using the LAMDA method to obtain a chattering-free control action.
- The stability analysis of LSMC controllers is addressed in order to guarantee the convergence of the system output towards the desired output.
- The implementation of LAMDA as an identifier for modeling and controlling process is proposed (called Adaptive LAMDA) for the first time, handling the concept of self-adjustment of the internal parameters.
- The proposed learning process of Adaptive LAMDA is based on a hybrid learning, which allows a quick convergence to the desired output, improving the

learning time and preventing that solutions be trapped in local minima. This is a great advantage over learning methods that only work with gradient descent, which is generally slow [14].

- A stability analysis of the learning algorithm is proposed to guarantee a rapid convergence of the estimated output towards the desired output.
- The computational complexity of the main proposed controllers is analyzed in terms of spatial and temporal complexity.
- The feasibility analysis of the implementation of Z-numbers applied to the LAMDA controller is studied.
- The definition of Z-rules based on different reliability values to modify the control action considering the deviation between the system output and the desired reference is proposed.
- A formal validation of all these improvements, in case studies of non-linear characteristics, and contrast of results with other methods, is carried out.

1.6 List of Publications

1.6.1 Journal Papers

- L. Morales, J. Aguilar, O. Camacho, and A. Rosales, “An Intelligent Sliding Mode Controller Based on LAMDA for a Class of SISO Uncertain Systems,” Information Sciences, vol. 567, pp.75-99, Aug. 2021.
- L. Morales, M. Herrera, O. Camacho, P. Leica, and J. Aguilar, “LAMDA Control Approaches Applied to

Trajectory Tracking for Mobile Robots," IEEE Access, vol. 9, pp. 37179–37195, 2021.

- L. Morales, J. Aguilar, A. Rosales, D. Chávez, and P. Leica, "Modeling and control of nonlinear systems using an Adaptive LAMDA approach," Applied Soft Computing, vol. 95, 2020.
- L. Morales, H. Lozada, J. Aguilar, and E. Camargo, "Applicability of LAMDA as classification model in the oil production," Artificial Intelligence Review, vol. 53, no. 3, pp. 2207–2236, 2020.
- L. Morales and J. Aguilar, "An Automatic Merge Technique to Improve the Clustering Quality Performed by LAMDA," IEEE Access, vol. 8, pp. 162917–162944, 2020.
- L. Morales, J. Aguilar, D. Chávez, and C. Isaza, "LAMDA-HAD, an Extension to the LAMDA Classifier in the Context of Supervised Learning," International Journal of Information Technology & Decision Making., vol. 19, no. 01, pp. 283–316, Jan. 2020.
- L. Morales, J. Aguilar, A. Garces-Jimenez, J. A. Gutierrez De Mesa, and J. M. Gomez-Pulido, "Advanced Fuzzy-Logic-Based Context-Driven Control for HVAC Management Systems in Buildings," IEEE Access, vol. 8, pp. 16111–16126, 2020.
- L. Morales, C. A. Ouedraogo, J. Aguilar, C. Chassot, S. Medjiah, and K. Drira, "Experimental comparison of the diagnostic capabilities of classification and clustering algorithms for the QoS management in an autonomic IoT

platform," Serv. Oriented Comput. Appl., vol. 13, no. 3, 2019.

1.6.2 Conference Proceddings

- L. Morales, D. Pozo, J. Aguilar, and A. Rosales, "Adaptive LAMDA applied to identify and regulate a process with variable dead time," in 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020, vol. 2020-July, pp. 1–8.
- L. Morales, J. Aguilar, A. Rosales, J. Gutiérrez, and D. Chavez, "An intelligent controller based on LAMDA," in Proceedings of the IEEE 4th Colombian Conference on Automatic Control, 2019, pp. 1–6.

1.6.3 Papers during the doctoral schooling

- L. Morales, D. Pozo." Mean-Variance Mapping Optimization for tuning scaling gains of a fuzzy control applied to a Cart-Inverted Pendulum." Revista Ibérica de Sistemas e Tecnologias de Informação, (E19), 2019.
- D. Pozo, L. Morales, D. Maldonado, J. Aguilar. "A Novel Methodology to obtain Optimal PI Controller Gains using Multi-gene Genetic Programming for FOPTD Systems", in: 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM). IEEE, 2018, pp. 1–6.
- J. Campos, S. Jaramillo, L. Morales, O. Camacho, D. Chávez. "PD + I Fuzzy Controller optimized by PSO applied to a variable dead time process", in: IEEE ETCM, 2018, pp. 1–6.
- J. Campos, S. Jaramillo, L. Morales, O. Camacho, D. Chávez, D. Pozo. "PSO Tuning for Fuzzy PD + I

Controller Applied to a Mobile Robot Trajectory Control", in: IEEE INCISCOS, 2018, pp. 62–68.

- L. Morales, O. Camacho, P. Leica, D. Chávez "A Sliding-Mode Controller from a Reduced System Model: Ball and Plate System Experimental Application, in: 14th International Conference on Informatics in Control, Automation and Robotics. SCITEPRESS - Science and Technology Publications, 2017, pp. 590–597.
- L. Morales, J. Cepeda. "Feature Extraction from sEMG of Forearm Muscles, Performance Analysis of Neural Networks and Support Vector Machines for Movement Classification", in: Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics. SCITEPRESS - Science and Technology Publications, 2017, pp. 254–261.
- L. Morales, D. Pozo. "An experimental comparative analysis among different classifiers applied to identify hand movements based on sEMG", in: 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). IEEE, 2017, pp. 1–6.
- L. Morales, O. Camacho, P. Leica, D. Chávez. "Sliding-Mode control based on a model reference applied to a non-linear ball and plate system with time delay". 2017 IEEE 3rd Colomb. Conf. Autom. Control, 2017, 1–6.
- M. Herrera, L. Morales, A. Rosales, Y. Garcia and O. Camacho, "Processes with variable dead time: Comparison of hybrid control schemes based on internal model," 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), 2017.

1.7 Thesis Outline

The remainder of this thesis is organized as follows:

In Chapter 2 a theoretical framework of LAMDA is presented, detailing its fundamentals used for assigning objects to classes/clusters, learning method, internal parameters, and structure of the algorithm. Aditionally, a review of the state of the art of the algorithm is made, detailing the relevant literature in classification and clustering contexts, and the recent research advances related with LAMDA.

Chapter 3 presents the extensions of the algorithm in each context raised in the objectives of this thesis. Initially, it is formalized the extension of LAMDA focused on supervised learning, through the approach of two improvements to get a better performance. First, it is proposed to calculate as many Non-Informative Classes (NICs) as the number of classes, obtaining an adaptive NIC. Then, it is proposed to compute the Higher Adequacy Degree (HAD), which improves the assignment of objects to their respective class, reducing misclassification. In the clustering field, the extension LAMDA-RD is formalized in order to enhance the calculation of the internal parameters of LAMDA and the implementation of the automatic merge algorithm.

Chapter 4 presents the formalization of the different proposals of LAMDA controllers, the inference mechanism that allows obtaining the Rule-based LAMDA controller, the design of the LAMDA-SMC intelligent controller (LSMC) that bases its operation on the concepts of Sliding-mode control to guarantee robustness and stability through Lyapunov theory. Next, the combination of the LSMC controller with the

Z-number theory is detailed to consider reliability criteria (ZLSMC), and finally, the Adaptive LAMDA proposal for modeling and control of systems is presented, which bases its operation on online learning.

Chapter 5 presents the experiments and some of the most relevant results of the LAMDA-HAD and LAMDA-RD extensions. It also presents the detailed results of the application and design of the Rule-based LAMDA, LSMC, ZLSMC and Adaptive LAMDA controllers, which are tested in different case studies, validating the feasibility of their implementation, and the contributions that these present when are used in the control of systems.

Chapter 6 details the discussion of the results obtained in the experimental stage, carrying out an in-depth analysis of the advantages and disadvantages of the proposed methods, and addressing a comparative analysis of the LAMDA control with respect to other control methods.

Finally, Chapter 7 summarizes the most relevant conclusions obtained from this work, and provides suggestions for further research.

2. THEORETICAL FRAMEWORK

Classification for monitoring systems is a method where the learning process is carried out through a controlled training by an external agent (supervisor), which defines the response that an algorithm should generate from one or several determined inputs. In other words, it is a method to make a machine learn from the expertise, to take decisions based on the characteristics, categories or classes of interest of a given application [15]. In the literature, there are different approaches applied to classification, such as instance-based methods, decision tree-based methods, kernel-based methods, neural networks-based methods, among others [16,17]. These methods are used in several applications, such as business [18], medicine [19], fault detection [20], functional states detection in industrial systems [21], etc.

On the other hand, clustering, is useful in problems where unlabeled data is available [22]. The aim of clustering is to separate data into partitions with elements that have similar characteristics between them. Each cluster must be separable and compact, with respect to another cluster [13]. In the literature there exists different clustering approaches, some of them are distance-based [23], partitioning clustering, hierarchical clustering [24], density-based [25], fuzzy logic-based [26–29], or Gaussian methods [30],[31], among others. All these techniques depend on a previous stage of descriptor extraction, which are later used for the individual-cluster assignment performed by the algorithm. Historical data and streaming data [32] are application scenarios of the clustering techniques, but not all the methods can work in both contexts because the data is

obtained differently. In the first case, the complete database is available, while in the second case, new data arrives continuously. The importance of working in the context of data streaming is that the evolution patterns provide useful information, which can allow users to make immediate and correct decisions [33]. In classical clustering, data is assigned to one cluster; on the other hand, fuzzy clustering methods are based on the fuzzy membership degree, therefore, an individual can be a member of several clusters.

2.1 State of the Art of LAMDA

Classification and clustering methods based on fuzzy logic are widely used in the field of machine learning [34–40], [13,41–44]. One of the methods that can work in both contexts is LAMDA (Learning Algorithm for Multivariate Data Analysis), proposed in [11], which is based on the calculation of the Global Adequacy Degree (GAD) that corresponds to the membership degree of one individual to a class, through the contributions of all its descriptors. These contributions are calculated through fuzzy probability functions [45], obtaining as result the Marginal Adequacy Degree (MAD). By mixing MADs of an individual, GADs are calculated using fuzzy aggregation operators, which are the membership degrees of the individual to each class. With this information is possible determining the current state of a system (class).

This algorithm can work in supervised and unsupervised learning [46], and it is able to create new classes after the training. The decision to create a new class is based on a threshold known as the Non-Informative Class (NIC). The algorithm compares all the GADs, including the GAD of the NIC, and assigns the individual to the one that has the

highest value. If the GAD of the NIC has the highest value, a new class is created, otherwise the individual is assigned to some class. Due to the mentioned features, LAMDA stand out over other fuzzy classification algorithms.

Although the LAMDA feature of generating new classes can be very useful, there is a limitation on certain applications due to the handling of a single NIC for all classes because this parameter is not adaptable to each of them. That can cause individuals to be assigned to the NIC and therefore produce a new class; this occurs especially when the descriptors of the individuals have high levels of uncertainty (noise) intra-class [45]. Aditionally, the presence of descriptors that do not adequately characterize an individual can produce misclassification, because the algorithm ignores the membership degree with respect to all individuals of the classes; it only does with respect to the center of the classes.

There are several contributions of LAMDA in the field of machine learning, including: the detection of functional states of a pilot simulator [47]. In this work, the performance among LAMDA and neural networks is compared, and it is determined that the precision of LAMDA is better in 66% of the performed experiments. This algorithm has also been used in problems of fault detection in electrical distribution networks [48], in which an average performance of 83% has been obtained. In water plants [49–51] the algorithm is used to identify the different functional states that describe the behavior of the coagulation process. LAMDA has been identified 8 functional states of normal and abnormal functioning of the plant, allowing a constant monitoring of the process. In the previous cases, the algorithm has worked

online in unsupervised mode. In [21], it is applied for monitoring complex industrial processes, combined with the Markov's theory, which allows identifying the connections between functional states through a transition degrees matrix. In this case, the creation of new classes is not considered. In Fault Detection and Isolation (FDI) cases, this method has been tested in a two-tanks benchmark in the detection of states for avoidance of dangerous operating condition, in increasing availability and productivity, among other things [52,53]. In computer vision applications, this method has been used as a search algorithm for image recognition in supervised mode.

The versatility of the algorithm is demonstrated, and its performance is determined, based on tests with different operators, such as fuzzy probability distributions and aggregation operators [54,55]. One of the most important contributions of LAMDA is its implementation in a software for the supervision of complex systems. This software is SALSA (Situation Assessment using LAMDA classification Algorithm), and has been used in applications such as those presented in [53,56] for functional states detection.

Several modifications to the original LAMDA have been developed by different researchers to improve the results obtained by the algorithm in classification and clustering.

In the classification context, the following contributions have been proposed:

- T-LAMDA [57,58] is used in image processing, whose method is less sensitive to noise than the original algorithm, however it has not been tested in other fields.

- A hybrid method of LAMDA combined with genetic algorithms has been proposed by [59]. The results are analyzed to find the best parameters of compensation in an optical spectrum, improving the performance but a high machine time.
- LAMDA-FAR [60], improves the recognition of functional states of diesel engines. This method is based on the calculation of two thresholds used to make a verification in the assignment of an individual to a class. To establish the first threshold in each class, the difference between the maximum GAD value (greater in amplitude) and the minimum value of the second largest GAD is calculated. For the second threshold of each class, the difference between the minimum value of the largest GAD and the maximum value of the second largest GAD is calculated. With the two thresholds, it is verified if the GAD of an individual is within these ranges to be assigned to some class, otherwise it is sent to the NIC. This method can only be implemented when in each class the first and second GADs do not overlap, because this would result in negative thresholds, which is not logical, and it does not provide the information needed to improve the classification process.

In the clustering context, the most important recent contributions are:

- "LAMDA Triple Pi (π) operator (LAMDA-TP)" [45,61]. This operator is used in LAMDA as an aggregation function for the computation of the GADs, avoiding the creation of new clusters with few individuals.

- “LAMDA clustering method based on typicality degree and intuitionistic fuzzy sets” [13]. The authors propose the calculation of three functions: The Global Typicality Degree (GTD), the Intuitionistic Global Adequacy Degree (IGAD), and the Typicality and Intuitionistic Global Adequacy Degree (TIGAD). This proposal is applied in some study cases, presenting an adequate separation of clusters.

The algorithms described above have some drawbacks in the cluster formation. In the first case, LAMDA-TP does not depend on the exigency parameter α (formalized in the original LAMDA), which allows calibrating the permissiveness of the algorithm. In other words, it is a control parameter linked to the quality and number of created clusters. LAMDA-TP performs the clustering process based only on the similarity computed by the triple π operator, and the user cannot calibrate the algorithm partitions. LAMDA based on intuitionistic fuzzy sets improves the clustering stage; however, in [13], a comparison of the algorithm with respect to other similar methods is not presented, and based on the results, it is observed that a merge stage is required to group clusters of similar characteristics, to obtain better models. Aditionally, the formed partitions are not analyzed in terms of performance metrics, which allow evaluating their intra and inter-cluster qualities.

2.2 LAMDA fundamentals

LAMDA is a fuzzy method, based on the concept of the adequacy degree. Unlike conventional algorithms, it is not a distance-based method, which performs a similitude

analysis among descriptors of the individuals, in order to establish a relationship between each one and its respective class [62]. The membership degree of an object O to a class $C = \{C_1; C_2; \ldots; C_k; \ldots; C_m\}$ [11] is estimated in a non-iterative process. The individual O is formed by a vector of l descriptors [38]:

$$O = [o_1; \ldots; o_j; \ldots; o_l] \tag{2.1}$$

where o_j is the descriptor j of the individual O.

The descriptors are normalized in the range between [0,1]. This normalization is done with respect to the maximum and minimum values of each descriptor, as shown in (2.2).

$$\bar{o}_j = \frac{o_j - o_{jmin}}{o_{jmax} - o_{jmin}} \tag{2.2}$$

where o_{jmin} is the minimum value of o_j, o_{jmax} is the maximum value o_j and $\bar{o}_j$ is the normalized descriptor.

2.2.1 Marginal Adequacy Degree (MAD)

The MADs establish how similar a descriptor is with respect to the same descriptor in a given class. For a defined individual, the MAD vector is determined to characterize its situation with respect to each class. For MAD calculation, probability density functions are used. The most common are: Fuzzy Binomial Function, and Gaussian function [60].

2.2.1.1 Fuzzy Binomial Function

This is a fuzzy extension of the binomial function that allows calculating the membership degree, like a Bayesian probability [58]. The function depends on $\rho_{k,j}$ and $\bar{o}_j$ as follows:

$$MAD_{k,j} = \rho_{k,j}{}^{\bar{o}_j}\left(1-\rho_{k,j}\right)^{\left(1-\bar{o}_j\right)} \quad (2.3)$$

where $\rho_{k,j}$ is the average value of the descriptor j that belongs to the class k, and it is calculated for the case of supervised learning using (2.4).

$$\rho_{k,j} = \frac{1}{n_{k,j}} \sum_{t=1}^{t=n_k,j} \bar{o}_j(t) \quad (2.4)$$

where $n_{k,j}$ is the number of data of the descriptor j belonging to class k.

2.2.1.2 Gaussian Function

This function depends on three input variables $\sigma_{k,j}, \rho_{k,j}$ and $\bar{o}_j$ and assumes a normal distribution of the descriptor and it is computed as:

$$MAD_{k,j} = e^{-\frac{1}{2}\left(\frac{\bar{o}_j-\rho_{k,j}}{\sigma_{k,j}}\right)^2} \quad (2.5)$$

where $\sigma_{k,j}$ is the standard deviation of the descriptor j that belongs to the class k:

$$\sigma_{k,j}{}^2 = \frac{1}{n_{k,j}-1} \sum_{t=1}^{t=n_{k,j}} (\bar{o}_j(t)-\rho_{k,j})^2 \quad (2.6)$$

The possibility of finding MAD using (2.3), or (2.5), depends on the data distribution in each class. Applying (2.3) for the NIC, where the center is $\rho_{NIC} = 0.5$, thus the $MAD_{NIC} = 0.5$ for any value of the descriptor $\bar{o}_j$.

2.2.2 Global Adequacy Degree (GAD)

GADs establish the adequacy of the individual to each class, which is calculated mixing MADs with aggregation functions.

These functions are any of the tuples shown in the Table 2.1 and are computed recurrently. The exigency $\alpha \in [0, 1]$, allows a strict or permissible classification [60]. If α increases, the classification is stricter, therefore more individuals will be unrecognized (sent to the NIC), making the algorithm more selective [63]. The GAD of an individual $\bar{O}$ in the class k is calculated as:

$$GAD_{k,\bar{O}}\left(MAD_{k,1}, \dots MAD_{k,l}\right) = \alpha T\left(MAD_{k,1}, \dots, MAD_{k,l}\right) + (1-\alpha) S\left(MAD_{k,1}, \dots, MAD_{k,l}\right) \quad (2.7)$$

where $\bar{O}$ is the normalized individual, T is t-norm (intersection), and S is t-conorm (union).

Table 2.1. Aggregation functions for LAMDA [64]

Type	Aggregation functions
Product-Sum	$T(a,b) = ab$ $S(a,b) = a + b - ab$
Hammacher	$T(a,b) = \frac{ab}{p + (1-p)(a+b-ab)}$ $S(a,b) = \frac{a+b-ab-(1-p)ab}{1-(1-p)ab}$
Dombi	$T(a,b) = \frac{1}{1 + \sqrt[p]{\left(\frac{1-a}{a}\right)^p + \left(\frac{1-b}{b}\right)^p}}$ $S(a,b) = 1 - \frac{1}{1 + \sqrt[p]{\left(\frac{a}{1-a}\right)^p + \left(\frac{b}{1-b}\right)^p}}$

*p is the sensibility of the function

The object $\bar{O}$ is assigned to the class with the maximum GAD, where index corresponds to the number of the class:

$$index = \text{argmax}\left(GAD_{1,\bar{O}}, \ldots, GAD_{m,\bar{O}}, GAD_{NIC,\bar{O}}\right) \quad (2.8)$$

The stages of LAMDA operation are shown in Figure 2.1. The MAD calculations are made for each descriptor in each class, using the probability density functions presented in (2.3) and (2.5). With the MADs, the calculation of the GAD of each class is made using aggregation functions (block "$\mathcal{L}$") and the exigency level "α". Finally, the class corresponding to the higher GAD calculated for the individual $\bar{o}_j$ is where it belongs to. If the higher GAD is the GAD_{NIC}, the individual is not recognized to any class and is sent to the NIC to create a new class.

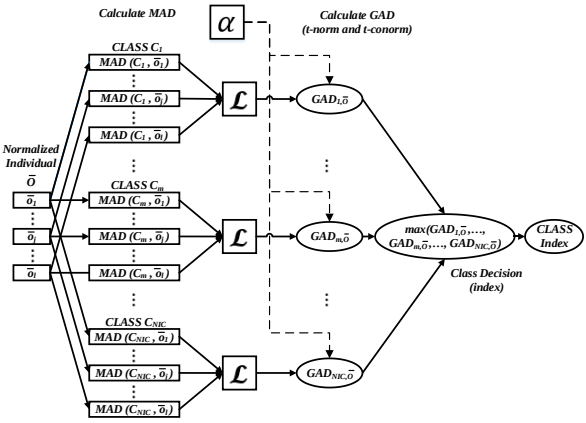


Figure 2.1. Structure of LAMDA algorithm

In the works like [50,60,65] where LAMDA is working in the classification context, in these papers we have been able to observe that LAMDA has problems making a correct classification, either because the NIC is constant for all classes, or because the GAD calculation is unreliable. On

the other hand, in the works like [13], [45], [58], where LAMDA is working in the clustering context, is observed that the number of created partitions does not correspond with the number of desired clusters, which can be excessive or impractical for the expert. Therefore, our contribution in the next section is the formalization of extensions that improve performance in both classification and clustering contexts.

2.3 Artificial Intelligence in the field of Control Systems

Nowadays, the artificial intelligence has allowed the development of very powerful techniques, useful for modeling nonlinear systems whose dynamics are complex and unknown [66]. The development of these techniques has increased considerably due to the computational power of the computers, allowing the implementation of learning algorithms with high accuracy and fast in processing terms, considering the inherent uncertainty and changing conditions of the systems [67]. Due to the versatility of these methods, it is possible to perform offline and online system modeling [68], which are very useful in control schemes, especially when the mathematical model of the system is unknown or variable. Specifically, in the case of online learning, the main advantage is the adaptation to changes in the dynamics of the system to be modeled/controlled, since it learns constantly the behavior of the process based on the input and output data. Many of the applications related to industrial processes, aeronautics, robotics and power systems require the incorporation of artificial intelligence into the control schemes, due to the adaptive feature that it provides when the mathematical model is

complex, unknown or inaccurate. The most used approaches for modeling and control of systems are Artificial Neural Networks (ANN) [69],[70],[9], Fuzzy Logic [71], and the hybrid models between the ANNs and the Fuzzy Inference Systems (ANFIS) [72], [73]. ANFIS is considered as a universal approximator [74], with the ability to represent any parameterized model. The adaptive neuro-fuzzy inference systems (ANFIS) are schemes that combine characteristics of both models (neuronal and fuzzy), with a fixed structure of nodes and layers, using the criteria of the neural networks for the learning process to perform the parametric adjustment, showing excellent results in different application fields related to model and control [75–79].

The aforementioned approaches are generally used in schemes of Adaptive Inverse Control (AIC). This control methodology has been studied for the last three decades, in which different researchers have made interesting proposals, applied to unknown plant dynamics [80].

Fuzzy logic is excellent for modeling systems. It is generally rule-based and has had a wide field of application in control systems, facilitating the design of applicable nonlinear controllers from simple systems to complex chaotic systems [81], [82]. Fuzzy logic has also been used in the design of AIC based on the creation of rules whose parameters can adapt automatically through learning criteria.

The neuro-fuzzy systems have also been developed in AIC as presented in [83–86]. In these works, the authors have been demonstrated that the proposed adaptive inverse dynamics control scheme is effective to improve the control performance of the system with uncertainties. In the

literature, several works have made efforts to improve the performance of neuro-fuzzy models through the use of hybrid learning [72], self-tuning ANFIS based in genetic programming [87], learning based on square-root cubature Kalman filter (SCKF) or recursive least squares (RLS) [88], or learning techniques that use input space partitioning through sub-clustering for higher dimensional regression problems (extreme learning [89]).

2.4 Z-Numbers

Zadeh [90] has proposed the Z-numbers, an extension of the fuzzy numbers composed of two elements: constraint and reliability, which are the ordered pair of fuzzy numbers. In a Z-number, the first element is used to characterize the uncertain information, and the second element is used to characterize the reliability (confidence) in the information. As it is analyzed in [91], the reliabilities of the fuzzy values of the variables in the set of rules are an issue in the modeling of the fuzzy systems, affecting the accuracy of the decision making process. Taking into consideration the uncertainties in the process to be controlled, the concept of Z-number can be more effectively used for the design of control algorithms in this case, the LAMDA controller.

Currently, Z-numbers are studied in different application fields, such as: decision making, economics, optimization, risk assessment, prediction and rule-based systems characterization with imprecise information. Thus, one of the applications is in the fuzzy reasoning to handle imperfect information characterized by the combination of fuzzy and probabilistic uncertainties in If-Then Rules systems [92]. The idea of converting a Z-number in a classic fuzzy number

without losing information is rather significant for many applications. Kang et al. [93] present a proposal to solve this issue based on the Fuzzy Expectation of a fuzzy set, through a simple procedure that can be applied to triangular and trapezoidal membership functions, but remaining open to research the application in Gaussian functions.

Due to the novelty of working with Z-numbers and being a recent development, Aliev [94] presents an initial proposal of basic operations that allow the treatment of uncertain information. Multi-criteria decision making (MCDM) under uncertain environments has been studied in [95–98], in order to take into consideration efficiently the reliability information. Aditionally, the conversion from a Z-number to a crisp number is useful in the fuzzy decision making and risk assessment. To do this, the work [93] proposes to compute the centroid of the interval-valued of the fuzzy set with the Karnik-Mendel algorithm. Z-numbers also have been used for the system state detection, especially failure modes in an aircraft turbine [99]. This paper demonstrates the viability of the proposed method using the reliability criterion. Finally, the Total Utility (TU) of a Z-number [100] is a new concept used to measure the total effects of a Z-number, and can be used to determine the ordering of Z-numbers with the aim to be applied in MCMDs under uncertain environments. The advantage of this method is to be able to work with triangular, trapezoidal and Gaussian membership functions, considerably expanding its field of application, which could be used by LAMDA.

In the field of control systems, there are few works that have focused their efforts on applying the concepts of Z-numbers in the design of controllers. Recently, Abyev [91,101]

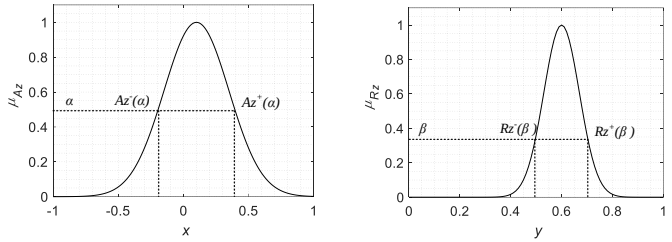
presented the development of a fuzzy inference system using Z-number for omnidirectional robot. In these works, the fuzzy inference system is designed for the control of the linear and angular speed of a robot soccer, independently. The proposed Z-rules are an extension of the classical fuzzy rules that consider the reliability of the constraint, but in all the rules is used the same reliability (Usually) to compute the control action. The inference method is based on distance measures of fuzzy sets, which takes the concepts of the α-cuts applied to the antecedent part, where the deviation of the input signals from the fuzzy values of the variables are determined. The proposed controller is validated and compared with other fuzzy methods, presenting interesting results. In [102], the same procedure is applied for a dynamic plant control where the transient response of the designed controller is compared with the transient response of a conventional fuzzy controller, demonstrating the suitability of the designed system in control of dynamic plants. Finally, the authors of [103] present the trajectory tracking of a wheeled mobile robot, combining the constraint and reliability in multi-input and multi-output rules. The antecedent considers the instantaneous distance measurements and the orientation gaps, and the consequent is computed by the interpolative reasoning and the graded mean integration approach. The authors highlight that this approach avoids the complexity of encoding error gradients, and it is able to cope with missing observations.

2.4.1 Z-numbers formalization

A Z-number is a pair of fuzzy numbers defined as:

$$Z = \{(Az, Rz) | \mu_{Az} \epsilon [0,1], \mu_{Rz} \epsilon [0,1] \} \quad (2.9)$$

where Az is the restriction (constraint) on the values of the observation x, and Rz is the reliability metric of the first fuzzy number in the space of y. For simplicity Az and Rz are considered Gaussian fuzzy numbers defined by a binary (ρ, σ), where ρ is the center of the function and σ is the width of the function. As an example, Figure 4.8 shows the membership functions of Az and Rz where it is appreciated how the parameters ρ, σ modify the position and shape of the curve respectively.



(a) (b)

Figure 2.2. Membership functions of the Z-number $Z = (Az, Rz)$ with $Az = Gauss(0.1,0.25)$, $Rz = Gauss(0.6,0.07)$

A conventional fuzzy if-then rule for two inputs and one output can be stated as:

$$IF\ o_1\ is\ F_1^p\ and\ o_2\ is\ F_2^q\ THEN\ y_1\ is\ G_1 \quad (2.10)$$

where F_1, F_2 are fuzzy sets and $G_1 = \gamma_1$ is a singleton function (constant).

In the case of Z-numbers, it is convenient to express a generalization of the basic if-then rule in terms of Z-valuations as follows:

$IF\ o_1\ is\ \left(Az_1^p, Rz_1^p\right)\ and\ o_2\ is\ \left(Az_2^q, Rz_2^q\right)$

$$THEN\ y_1\ is\ (\gamma_1, R_1) \quad (2.11)$$

where Az_1^p,Rz_1^p, Az_2^q,Rz_2^q,A_{G1},and $R_{G1}$are the fuzzy sets of the Z-number.

An example of the characterization of the Z-rules is: “If high price of oil, likely, and high price of the refining process, commonly, then high price of gasoline, very likely”.

As seen in the example, the application of Z-numbers is focused on modeling uncertain information from the real world. More examples of the real applications of Z-numbers are described in detail in [104]. In the case of control systems there is a potential application field of these concepts that can improve the performance of the controller.

2.4.2 Total Utility of Z-numbers (TU)

TU is potentially useful to simplify the Z-number applications in decision making. The function of the TU is derived from the format of Z-numbers without subjective judgment as stated in [100]. Such a simplification of the Z-numbers is of great importance when it is required to represent the restriction and the metric reliability in a single parameter, which can be useful in the design of intelligent controllers.

The procedure to compute is detailed as follows: let the mathematical expressions of the membership functions for Az and Rz defined as [100]:

$$\mu_{Az}(x) = e^{-\frac{1}{2}\left(\frac{x-\rho_1}{\sigma_1}\right)^2}\ ;\ \mu_{Rz}(y) = e^{-\frac{1}{2}\left(\frac{x-\rho_2}{\sigma_2}\right)^2} \quad (2.12)$$

where $-1 \leq \rho_1 \leq 1$ is the position of the center of the peak of the curve $\mu_{Az}(x)$ and $\sigma_1 > 0$ is its standard deviation used to control the width of the bell, $0 \leq \rho_2 \leq 1$ is the position of

the center of the peak of the curve $\mu_{Rz}(x)$ and $\sigma_2 > 0$ is its standard deviation.

If the α-cut, $\alpha = \mu_{Az}(x)$, then x is computed as:

$$x = \rho_1 \pm \sqrt{-2{\sigma_1}^2 ln\alpha};$$

$$Az^-(\alpha) = \rho_1 - \sqrt{-2{\sigma_1}^2 ln\alpha}; Az^+(\alpha) = \rho_1 + \sqrt{-2{\sigma_1}^2 ln\alpha} \quad (2.13)$$

If the β-cut, $\beta = \mu_{Rz}(y)$, then y is computed as:

$$y = \rho_2 \pm \sqrt{-2{\sigma_2}^2 ln\beta};$$

$$Rz^-(\beta) = \rho_2 - \sqrt{-2{\sigma_2}^2 ln\beta}; Rz^+(\beta) = \rho_2 + \sqrt{-2{\sigma_2}^2 ln\beta} \quad (2.14)$$

The TU of a Gaussian Z-number is computed as [100] :

$$TU(Z) = TU(Az, Rz) = \int_0^1 \int_0^1 \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \left\{ \begin{array}{l} \left[\frac{Az^-(\alpha) + Az^+(\alpha)}{2} + x(Az^+(\alpha) - Az^-(\alpha)) \right] e^{-[Az^+(\alpha) - Az^-(\alpha)]^2} \\ \times \left[\frac{Rz^-(\beta) + Rz^+(\beta)}{2} + y(Rz^+(\beta) - Rz^-(\beta)) \right] e^{-[Rz^+(\beta) - Rz^-(\beta)]^2} \end{array} \right\} dxdyd\alpha d\beta \quad (2.15)$$

Let:

$$A_1 = Az^-(\alpha) + Az^+(\alpha) \quad (2.16)$$

$$A_2 = Az^+(\alpha) - Az^-(\alpha) \quad (2.17)$$

$$R_1 = Rz^-(\beta) + Rz^+(\beta) \quad (2.18)$$

$$R_2 = Rz^+(\beta) - Rz^-(\beta) \quad (2.19)$$

Replacing (2.13), (2.14), (2.16)-(2.19) in (2.15), it is obtained:

$$TU(Z) = TU(Az, Rz)$$

$$= \int_0^1 \int_0^1 \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \left[\frac{A_1}{2} + xA_2 \right] e^{-{A_2}^2} \left[\frac{R_1}{2} + yR_2 \right] e^{-\tilde{B}_2{}^2} \, dxdyd\alpha d\beta$$

(2.20)

$$TU(Z) = \int_0^1 \int_0^1 e^{-{A_2}^2} e^{-{R_2}^2} \frac{A_1}{2} \frac{R_1}{2} d\alpha d\beta \quad (2.21)$$

$$= \rho_1 \rho_2 \int_0^1 \int_0^1 e^{-\left(2\sqrt{-2{\sigma_1}^2 ln\alpha}\right)^2} e^{-\left(2\sqrt{-2{\sigma_1}^2 ln\beta}\right)^2} d\alpha d\beta \quad (2.22)$$

$$= \frac{\rho_1 \rho_2}{(1 + 8{\sigma_1}^2)(1 + 8{\sigma_2}^2)} \quad (2.23)$$

3. PROPOSED LAMDA EXTENSIONS

Based on the literature described in the previous chapter, it has been possible to detect that LAMDA has certain drawbacks, as described above, in the learning stage for both the supervised and unsupervised cases, which cause its performance to decrease. In order to address these problems, this section presents extensions (improvements) to the algorithm that are described in detail in the papers [105–108] published based in this work, where large field of experimentation and validation of the proposed extensions is covered.

3.1 Extension of LAMDA in the classification context

In this subsection, the Iris dataset [109] (common benchmark for classification) is chosen as a practical example to expose the problems and weaknesses of the original algorithm, and the possible causes of misclassification when the algorithm is tested. The Iris dataset is a multivariate dataset with three classes corresponding to the species: setosa, virginica and versicolor, with four descriptors (sepal length and width in

cm, and petal length and width in cm), with 50 individuals in each class. The real distribution of its individuals in each class is shown in Figure 3.1, in which the different colors represent the three classes where are shown the class setosa (50 first elements in blue), virginica (50 elements in green) and versicolor (50 elements in black).

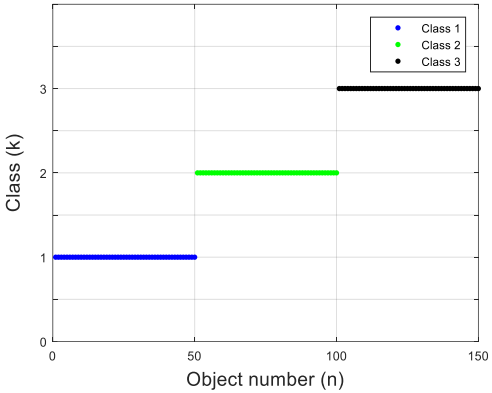


Figure 3.1. Iris Benchmark data and classes

The fuzzy binomial function has been used to calculate the MAD. For the calculation of the GAD, the Hammacher operator has been used recurrently, and a high exigency level has been parameterized ($\alpha = 0.9$) to reduce the misclassification. The GAD of each element for each class calculated by the algorithm, are presented in Figure 3.2.

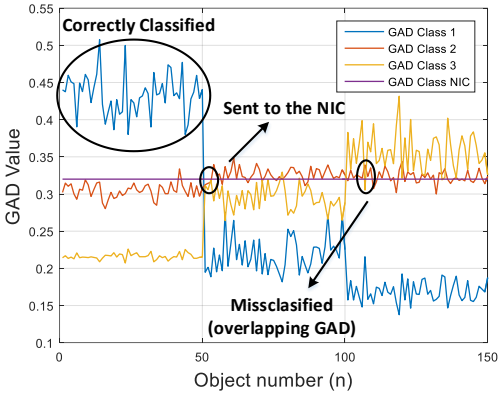


Figure 3.2. GADs in each class for Iris dataset

Choosing the maximum GAD for each individual of Figure 3.2. The classification shown in Figure 3.3 is obtained, where clearly it has been possible to observe the three cases marked in Figure 3.2:

1 Well classified: individuals whose descriptors are clearly differentiated in each class, so the GADs are well defined (for example in class 1, the blue GAD is clearly differentiable from the rest).

2 Sent to NIC class: individuals whose maximum GAD is less than the value of GAD_{NIC}, so that these individuals are sent to the NIC to create a new class (e.g, first individuals in class 2).

3 Misclassified: individuals that, due to the similarity of their descriptors, cause overlapped GAD, producing errors in the classification (as can be seen in class 3, the red and yellow GAD are overlapped).

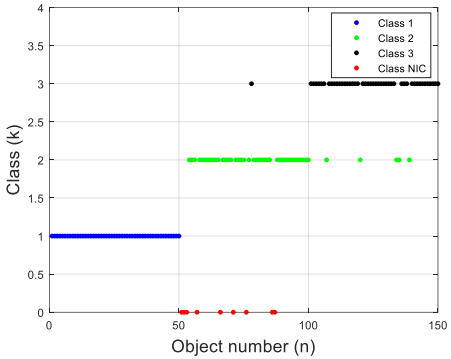


Figure 3.3. Classification results of the original LAMDA algorithm

Note that the assignments of elements in Figure 3.3 coincide with those shown in Figure 3.2 (same colors) since a correct classification is made, however in some cases elements are misassigned and others are sent to the NIC (red dots).

An alternative approach is proposed in [60], called LAMDA-FAR, which is based on measuring the maximum and minimum distances between the two highest GAD for each individual $\bar{O}$ and for each class k. However, this proposed methodology is valid only when the GADs in each class does not overlap each other. For this reason, LAMDA-FAR cannot be applied in problems as the Iris dataset. Based on this, it is important to attack the two problems outlined above. LAMDA-HAD addresses these drawbacks with two strategies:

- First, it is proposed to calculate as many NICs as the number of existing classes. In this case, each NIC (and its GAD) is calculated based on the intrinsic characteristics of each class, which prevents to send well-classified individuals to the NIC, as it occurs in the

original LAMDA algorithm, where only one NIC is calculated for all existing classes.

- The second strategy is based on calculating the Higher Adequacy Degree (HAD), which consists of measuring the degree of similarity of the GAD of an individual in relation to the average of the GADs of the existing classes through probabilistic operators, allowing to know with greater certainty the class to which an individual belongs. It allows improving the robustness of the algorithm and avoiding that individuals with similar characteristics between classes are misclassified.

3.1.1 Extension 1: Adaptable GAD_{NIC}

The proposal is based on reducing the drawbacks corresponding to individuals sent incorrectly to the NIC; this problem is clearly shown in Figure 3.3, where the red dots are unidentified individuals that really belong to class 2. To avoid this, the calculation of the GAD_{NIC} adaptable to each class has been considered.

For the calculation of Adaptable GAD_{NIC}, the following definitions are proposed, which consist of simple mathematical operations that do not consume excessive computational time.

Definition 1. $MGAD_{k,p}$ is the average value of the GADs of the class p in a class k. Considering that n_k is the number of individuals belonging to class k and $p = \{1, \dots m\}$ where m is the number of classes. This parameter is computed as:

$$MGAD_{k,p} = \frac{1}{n_k} \sum_{t=1}^{t=n_k} GAD_{p,t} \tag{3.1}$$

where $GAD_{p,t}$ is the GAD of the individual t for the class p, in the class k. For example, $MGAD_{1,1}$ is the average value of blue $GADs$ in the class 1, as shown the graphical representation of each $MGAD$ (green lines) in Figure 3.4.

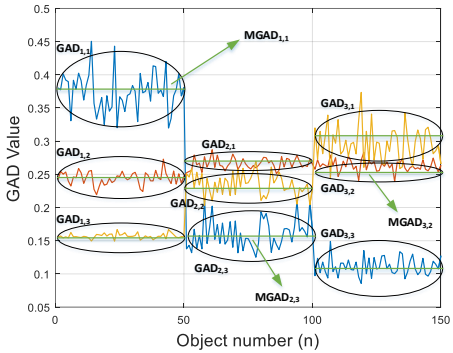


Figure 3.4. $MGAD$ obtained for each GAD in the different classes

In Figure 3.4. see that the MGAD are calculated based on the GAD of each class, thus $MGAD_{k,p}$. is the mean values of all the $GAD_{k,p}$.

Definition 2. Let GAD_{NIC_k} the GAD of the NIC for the class k. The GAD_{NIC_k} is computed as the average value of all the $MGAD_{k,p}$ in each class, e.g., if there are three classes as in Iris, it will be calculated GAD_{NIC_1}, GAD_{NIC_2}, and GAD_{NIC_3} (see purple lines of Figure 3.5). The calculation of GAD_{NIC_k} is done using (3.2).

$$GAD_{NIC_k} = \frac{1}{m}\sum_{p=1}^{p=m} MGAD_{k,p} \qquad (3.2)$$

As shown in (3.2), GAD_{NIC_k} depends on the contributions of all MGADs, therefore, it also depends on the distribution of all GADs in each class.

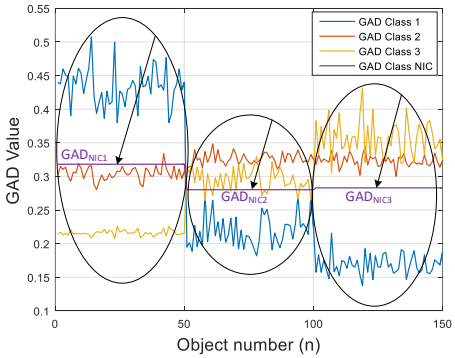


Figure 3.5. Mean value GAD_{NIC} for each class in the Iris dataset

3.1.2 Extension 2: Higher Adequacy Degree (HAD)

To decrease the probability of obtaining individual misclassification due to the similarity of its descriptors with features of different classes, the calculation of the Higher Adequacy Degree is proposed. This parameter allows establishing the similarity between the $GAD_{p,\bar{O}}$ of the normalized individual $\bar{O}$ and the GADs of each class k. Unlike the original LAMDA, the approach does not directly choose the maximum GAD. First, an estimation of the class index is made to which the GADs of the individual are most similar. For this, it is proposed performing additional calculations to the algorithm based on the following definitions.

Definition 3. $AD_{GAD_{k,p,\bar{O}}}$ is a parameter that allows to obtain a measure of similarity between the GAD of an individual $\bar{O}$ and each $MGAD_{k,p}$. It is calculated using the following probability density function:

$$AD_{GAD_{k,p,\bar{O}}} = MGAD_{k,p}{}^{GAD_{p,\bar{O}}}(1 - MGAD_{k,p})^{(1-GAD_{p,\bar{O}})} \quad (3.3)$$

For $\bar{O}$ evaluated in each class with (2.2)-(2.7). This expression is similar to the used (2.5) which is a binomial function used to measure similarity between objects.

Definition 4. Let $AD_{GAD_{k,p,\bar{O}}}$ the new adequacy degree of the GAD, the calculation of the Higher Adequacy Degree (HAD) of an individual to a class is done by adding all the $AD_{GAD_{k,p,\bar{O}}}$ in the class k as:

$$HAD_{k,\bar{O}} = \sum_{p=1}^{p=m} AD_{GAD_{k,p,\bar{O}}} \quad (3.4)$$

Based on this calculation, the highest value of the $HAD_{k,\bar{O}}$ is the one with the highest membership degree, obtaining the estimated index E_I of the class to which the individual has the highest probability to belong.

$$E_I = max(HAD_{1,\bar{O}}, \dots, HAD_{k,\bar{O}}, \dots, HAD_{m,\bar{O}}) \quad (3.5)$$

Definition 5. Let $index$, the index of the class with the highest adequacy degree, then it is necessary to verify if the maximum value of $GAD_{E_I,\bar{O}}$ is greater than the $GAD_{NIC_{E_I}}$. If this rule is met, then the individual $\bar{O}$ belongs to the class E_i; otherwise, it will be sent to the NIC (see (3.6)).

$$index = argmax(GAD_{E_I,\bar{O}}, GAD_{NIC_{E_I}}) \quad (3.6)$$

3.1.3 General procedure of LAMDA-HAD

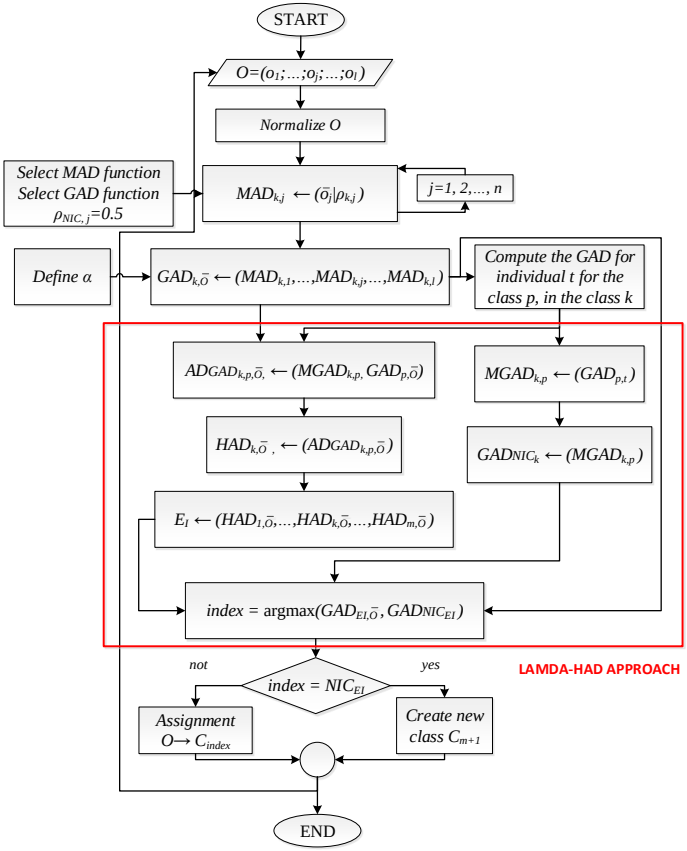


Figure 3.6. LAMDA-HAD structure

The operating scheme of the LAMD-HAD is presented in Figure 3.6, in which the additional steps of the approach (marked in the red box) can be seen in detail in order to improve the performance of LAMDA in classification tasks.

3.2 Extension of LAMDA in the clustering context

In some clustering applications, the number of created partitions performed by LAMDA does not correspond with the number of desired clusters, which can be excessive or impractical for the expert. Therefore, the contribution in this subsection is the formalization of an automatic merge technique to update the cluster partition performed by LAMDA to improve the quality of the clusters, and a new methodology to compute the Marginal Adequacy Degree that enhances the individual-cluster assignment.

In [13,45] has been shown that LAMDA creates clusters that do not correspond with the number of desired groups. Clusters with a high similarity degree should be merged in a single cluster, according to a similarity measure. Thus, the algorithm should automatically decide when a merge process between clusters is required. For that, it is proposed to hybridize the original algorithm with distance measurements. The proposed method is called LAMDA-RD and in where the split task is considered as an intrinsic LAMDA feature, because it can create new groups from the global adequacy concept.

Cauchy function [110] is used to compute a membership degree $\mu_c(o)$ that models the similarity of an individual to a cluster. This function requires the distance of the individual o to a prototype member p_0 (center of the class) represented as $dist(o, p_0)$. The Cauchy function is computed as:

$$\mu_c(o) = \frac{1}{1 + dist(o, p_0)} \tag{3.7}$$

3.2.1 Robust Distance

Definition 6. Cauchy Marginal Adequacy Degree ($CMAD$). This parameter corresponds to the MAD computed using the Cauchy function presented in (3.7). If we apply this expression in LAMDA context, it must be considered that $o = \bar{o}_j$ and $p_0 = \rho_{k,j}$ (descriptor j of the centroid of the cluster k computed with (2.4)), now redefining the MAD as $CMAD$ it is obtained (see [107] for more details):

$$CMAD_{k,j}(\rho_{k,j}) = \frac{1}{1 + dist(\bar{o}_j, \rho_{k,j})} \qquad (3.8)$$

To keep the MAD concept of probability function, we set as $CMAD_{NIC,j}(\rho_{k,j}) = 0.5$, this is the threshold for the NIC as computed in fuzzy binomial function. In clustering, $\rho_{k,j}(t)$ is calculated with (3.9), and it is the mean value of the descriptor j in the previously created cluster k. It is updated progressively each time that a new element is added. $n_k(t-1)$ is the number of objects previously assigned to the cluster k.

$$\rho_{k,j}(t) = \rho_{k,j}(t-1) + \frac{\bar{o}_j(t) - \rho_{k,j}(t-1)}{n_k(t-1) + 1} \qquad (3.9)$$

Definition 7. Robust Marginal Adequacy Degree ($RMAD$). This parameter corresponds to the product of the $CMAD$ and a penalty factor $K_{k,\bar{O}}$ computed for each cluster k. To obtain $K_{k,\bar{O}}$, two parameters are required: the first one is the distance of the individual $\bar{O}$ to the center of each cluster k ($d_{k,\bar{O}}$), which is calculated as:

$$d_{k,\bar{O}} = dist(\bar{o}_j, \rho_{k,j}) = \frac{1}{n}\sum_{j=1}^{n} \left|\bar{o}_j - \rho_{k,j}\right| \qquad (3.10)$$

And the second parameter is the threshold $d_{nb} \in [0,1]$, called "average distance between neighbors", which must be set by the user (in Appendix C is described a method to calibrate this parameter).

Proposition 1. The penalty factor $K_{k,\bar{O}}$ is computed with (3.11). If the average distance $d_{k,\bar{O}}$ is greater than d_{nb} $(d_{k,\bar{O}} > d_{nb})$, then $K_{k,\bar{O}}$ is computed as:

$$K_{k,\bar{O}} = \frac{d_{nb}}{d_{nb} + dist\left(d_{k,\bar{O}}, d_{nb}\right)} \tag{3.11}$$

As shown (3.11), if $dist\left(d_{k,\bar{O}}, d_{nb}\right)$ increases, then $K_{k,\bar{O}}$ decreases.

Proposition 2. If the average distance $d_{k,\bar{O}}$ is less than d_{nb}, $(d_{k,\bar{O}} \leq d_{nb})$, then $K_{k,\bar{O}}$ is set to 1, because it is not required to penalize the $CMAD$ of individuals that are within the threshold. Now, $RMAD_{k,j}$ is computed as:

$$RMAD_{k,j}\left(\rho_{k,j}\right) = K_{k,\bar{O}} \times CMAD_{k,j} \tag{3.12}$$

As shown (3.12), $RMAD_{k,j}$ is equal to $CMAD_{k,j}$ if the condition of Proposition 2 is met, in other words, the distance between the individual $\bar{O}$ and the cluster k is within the threshold d_{nb}. According to Proposition 1, if the distance between the individual $\bar{O}$ and the cluster k is greater than the threshold, then $CMAD$ is penalized; therefore, a decrease in the adequacy degree is established. $K_{k,\bar{O}}$ reinforces the measure of similarity degree based on distances. The two established conditions of $d_{k,\bar{O}}$ affect the computation of the $RMAD$. The following properties $P1$ and $P2$ clarify how the penalty factor behaves before the different inequalities between $d_{k,\bar{O}}$ and d_{nb}:

$$P1: If\ \left(d_{k,\bar{O}} > d_{nb}\right) | d_{nb} \in [0,1]$$

$$\Longrightarrow K_{k,\bar{O}} = \frac{d_{nb}}{d_{nb} + dist\left(d_{k,\bar{O}}, d_{nb}\right)} = \frac{d_{nb}}{d_{nb} + \delta}; | \delta \in [0,1]$$

$$\Longrightarrow K_{k,\bar{O}} < 1 \therefore RMAD_{k,j}\left(\rho_{k,j}\right) < CMAD_{k,j} \quad (3.13)$$

$$P2: If\ \left(d_{k,\bar{O}} \leq d_{nb}\right) | d_{nb} \in [0,1]$$

$$\Longrightarrow K_{k,\bar{O}} = 1 \therefore RMAD_{k,j}\left(\rho_{k,j}\right) = CMAD_{k,j} \quad (3.14)$$

The penalty factor for the NIC is set $K_{NIC,\bar{O}} = 1$, because it is not required to penalize the Non-Informative Class. As observed in (3.13) and (3.14), the distance $d_{k,\bar{O}}$ allows penalizing the dissimilarity between the samples and the clusters. This parameter is called Robust Distance, hence, this proposal takes the name of LAMDA-RD. Once calculated $RMAD$, the computation of the GAD is like the original LAMDA, using (2.7), but with $RMAD$ instead of MAD.

3.2.2 Automatic merge algorithm

To describe the automatic merge algorithm for LAMDA, the following definitions are formalized:

Definition 8. A cluster C_k with n_k elements is described by the tuple:

$$C_k = \left(\rho_{k,j}, \bar{O}_k, index - k\right) \quad (3.15)$$

where $\rho_{k,j}$ is the centroid of the descriptor j in the cluster k, which is updated every time that a new individual is assigned to C_k (see (3.9)), $\bar{O}_k$ is the set of individuals in C_k, and $index\ k$ is the identifier of C_k.

Definition 9. The neighbor cluster C_{nb} with n_{nb} elements is described by the tuple:

$$C_{nb} = \left(\rho_{nb,j}, \bar{O}_{nb}, index - n_b,\right) \quad (3.16)$$

where $\rho_{nb,j}$ is the centroid of the descriptor j in the cluster nb, $\bar{O}_{nb}$ is the set of individuals in C_{nb}, and $index - n_b$ is the identifier of C_{nb}.

LAMDA is non-iterative, therefore, in the clustering process, one individual is analyzed at a time. So, according to the LAMDA fundamentals, the maximum GAD is where the individual is assigned to, so, it is noted that the second GAD of greater value is the nearest neighbor cluster.

The main problem to solve is the drawback of the original LAMDA: the excessive creation of clusters which has been described based on the citations at the end of section 2. So, it is essential to perform an automatic merge. The proposal is characterized by similarity measures based on distances and densities. In the merge stage, two cases can occur:

If the individual was assigned to the NIC, and therefore, a new cluster was created, we have to do an analysis individual – cluster (see Figure 3.7a). In the other hand, if the individual was assigned to an existing cluster C_k, we have to do an analysis cluster – cluster (see Figure 3.7b).

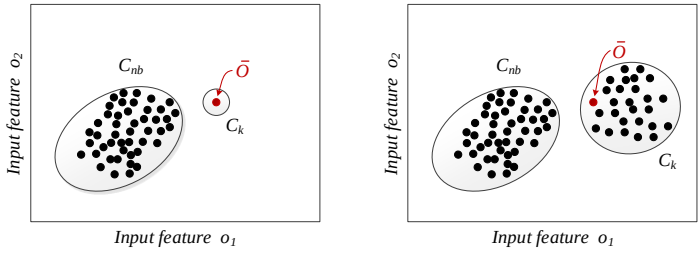


Figure 3.7. New sample assigned to (a) new cluster, (b) pre-existing cluster.

Definition 10. Measure of the compactness of the neighbor cluster $(t_{nb,j})$. It is the mean value of all the distances (in each descriptor) among the individuals belonging to the neighbor cluster C_{nb}, and it is computed as:

$$t_{nb,j} = \frac{\sum_{i=1}^{n_{nb}-1} \sum_{m=i+1}^{n_{nb}} \left| \bar{o}_{nb,j}^{i} - \bar{o}_{nb,j}^{m} \right|}{n_{nb} \times (n_{nb}-1) \times \ldots \times 1} \quad ; \forall\, j = 1, \ldots, n \tag{3.17}$$

where $\bar{o}_{nb,j}^{i}$ is the descriptor j of the individual i in the cluster C_{nb}.

Definition 11. Number of individuals in the overlapping area (N_I). The overlapping area is the region where individuals from more than two clusters are found. The number of individuals in this area is computed by counting the individuals in the overlapping area of the clusters C_k and C_{nb}, whose distance between its individuals is less than $t_{nb,j}$. For this, it is required to identify the individuals of each cluster C_k and C_{nb} that meet that condition, and then, the cardinality of the resulting subsets is calculated as:

$$N_k = \left\{ d\left(\bar{o}_{k,j}, \bar{o}_{nb,j}\right) < t_{nb,j} \; ; \forall\, j = 1, \ldots, n \right\}$$
$$\Rightarrow \; N_k = n(N_{kI}) \tag{3.18}$$

$$N_{nb} = \left\{ d\left(\bar{o}_{k,j}, \bar{o}_{nb,j}\right) < t_{nb,j} \; ; \forall\, j = 1, \ldots, n \right\}$$
$$\Rightarrow \; N_{nb} = n(N_{nbI}) \tag{3.19}$$

where N_k and N_{nb} are the number of individuals in the overlapping area for the cluster C_k and C_{nb}, respectively. The total number of individuals in the overlapping area N_I is:

$$N_I = N_k + N_{nb} \tag{3.20}$$

Definition 12. D_{k-nb} is the density in the overlapping area between two clusters C_k and C_{nb}, and it is computed as:

$$D_{k-nb} = \frac{N_I}{n_{nb} + n_k} \tag{3.21}$$

Proposition 3. Two clusters C_k and C_{nb} are merged, if $D_{k-nb} \geq D_t$. $D_t \in [0,1]$ is a density threshold set by the user. A high D_t value implies a greater density of individuals in the overlapping area.

Figure 3.8 shows the cases in which the condition of Proposition 3 is not satisfied.

$$\rho_{new,j}(t) = \bar{o}_j \tag{3.22}$$

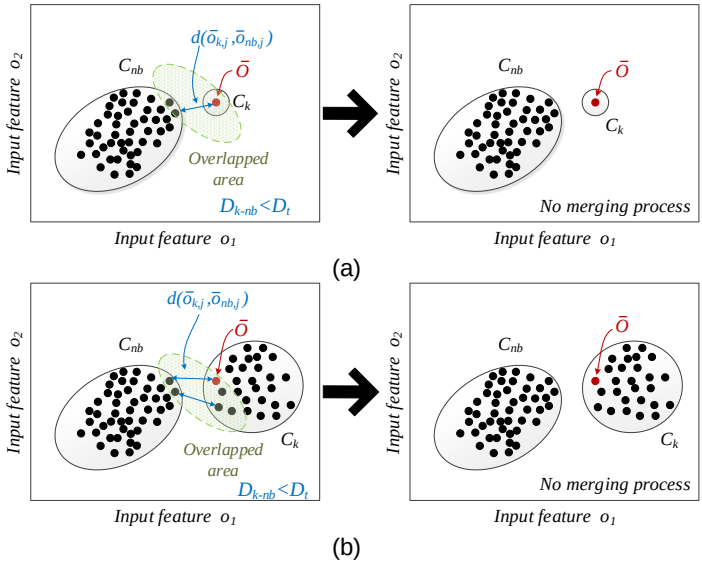


Figure 3.8. Graphical example to assign a new sample to a cluster, when Proposition 3 is not met, (a) the sample creates a new cluster, (b) the sample is assigned to a pre-existing cluster.

It is observed that the new individual $\bar{O}$ increases the density of the overlapped area between the clusters C_k and C_{nb}.

However, if $D_{k-nb} < D_t$, then the algorithm does not proceed to do the merge process, considering that there is not enough similarity between the two analyzed partitions.

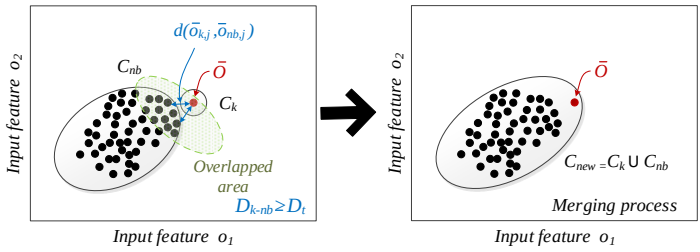
If the object is not merged, then it becomes the first element of a new cluster $C_k = C_{new}$. Figure 3.9 shows the cases in which the Proposition 3 is satisfied. It is observed that the new individual $\bar{O}$ increases the density of the overlapped area between the clusters C_k and C_{nb} computed with (3.21). If $D_{k-nb} \geq D_t$, then the algorithm proceeds to do the merge process, considering that there is enough similarity between the two analyzed groups.

Definition 13. Resulting New Cluster (C_{new}) with $n_k + n_{nb}$ elements. The resulting cluster after the merge process is given by the tuple:

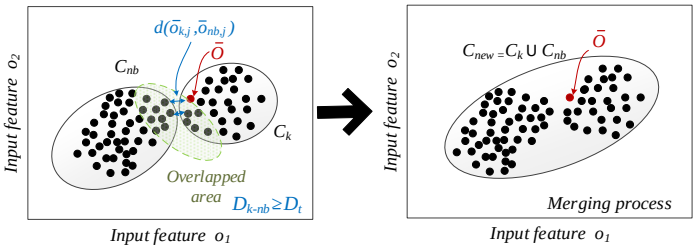
$$C_{new} = \{\rho_{new,j}, \bar{O}_k \cup \bar{O}_{nb}, index - new\} \quad (3.23)$$

$$\rho_{new,j} = \frac{1}{n_k + n_{nb}} \sum_{t=1}^{n_k+n_{nb}} \bar{o}_{new,j}^{t} \quad (3.24)$$

where $\bar{o}_{new,j}^{t}$ is the descriptor j of the individual t in the clusters C_k and C_{nb} that form the new cluster C_{new}.



(a)



(b)

Figure 3.9. Graphical example to assign a new sample to a cluster, if Proposition 3 is met, (a) the algorithm merges the individual to the neighbor cluster, (b) the algorithm merges the cluster where the individual was assigned with the neighbor cluster

As shown in Figure 3.9, each time that an individual is assigned to a cluster, it is evaluated if the density of the overlapping area has increased. The density is considered as a requirement to determine if the merge process should be executed according to the threshold D_t.

3.2.3 General procedure of LAMDA-RD

The scheme of Figure 3.10 details the LAMDA-RD based on distances and densities and the additional steps of our approach (marked in the red boxes). The first step is the normalization of the descriptors of the individual. Next, the $RMAD$ calculations are made for each descriptor in each cluster, using the Cauchy function, which considers the $K_{k,\bar{O}}$ parameter to penalize the dissimilarity between the individual and the clusters based on distances, as shown in (3.10)-(3.14). With $RMAD$, the GAD in each cluster is computed, setting a high value for the exigency level ($\alpha = 1$), with the aim to get a strict behavior.

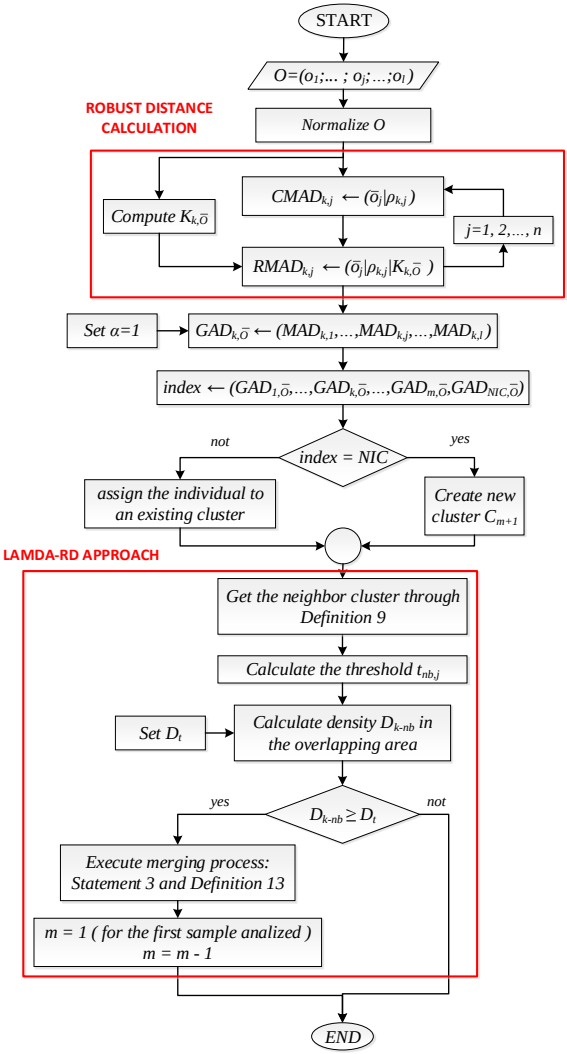


Figure 3.10. LAMDA-RD algorithm

The highest GAD defines the cluster in which the individual must be assigned. However, if the maximum GAD corresponds to the NIC, then a new cluster is created, being this individual the first sample of the new group. In the merge stage is evaluated if this process is required between the cluster in which the individual was assigned and the neighboring cluster, this because the individual can be located in the overlapping zone between both clusters, fulfilling the merge requirement of Proposition 3. In general, the algorithm starts with $m = 0$ when no element has been analyzed. When the first sample to be evaluated arrives, then the first cluster is created ($m = 1$) as shown in Figure 3.10. Next, when the second sample arrives, then it is evaluated, and if the conditions established by the algorithm are met, then this sample is assigned to cluster 1, otherwise, a new cluster is created ($m = 2$). This process is followed successively for all the samples, until evaluating the last sample N, assigning it to one of the current clusters or a new one. Thus, the algorithm does not require the definition of the number of clusters (m).

4. LAMDA IN THE FIELD OF CONTROL SYSTEMS

The application of artificial intelligence to the control theory aims to improve the overall performance of a complex system. In a control problem, artificial intelligence techniques can be used to meet different objectives, such as: plant modeling and/or control, system performance improvement, calibration and parameterization, among others [4]. A current trend is to incorporate artificial intelligence techniques into the control systems, in order to develop simple controllers that improve the performance without needing to know the model of the plant in detail, with enough robustness to achieve the control objectives and feasible implementations, considering, e.g., low computation time during the execution of the algorithms.

Over the last decades, researchers have devoted much effort to the study of SISO uncertain systems. These systems can have irregular and unpredictable behavior due to uncertainties in modeling [111]. Nonlinear controllers are a very useful tool to solve this problem, which is an open-field research and it is continuously developing new alternatives or combining different methodologies that have improved the response and behavior of the different systems to be controlled [7,112–115]. PID controller is still the most used standard tool for industrial applications [116]. The main drawback of PID is that its efficiency depends on a correct tuning of its scaling gains, a process that can be performed with different methodologies. However, in practice, more systems cannot be modeled exactly due to undesired features, such as non-linearity, and time-variability, which

make more difficult to tune the controller parameters. Also, some known tuning methods are model-based, so they are not suitable in applications of uncertain or variable dynamics.

Fuzzy Logic Control (FLC) is an interesting technique to develop intelligent controllers with excellent results in several applications. The main features of the FLC [117–120] are: a) excellent performance in systems in which the model of the system to be controlled is not exactly known [121], b) it allows to design robust controllers that are capable of delivering a satisfactory efficiency against uncertainty [111] and, c) a nonlinear controller developed empirically can be designed. Unlike the conventional model-based control techniques, fuzzy controllers require a set of heuristic rules to compute the control action to be applied to the plant. For the definition of the rules, the designer (expert) needs to have prior knowledge of the system operation [122] to cover a broader range of operating conditions. Because Fuzzy Logic (FL) based techniques do not require an accurate model of the plant to be controlled, they are considered as intelligent controllers. FL is one of the most used artificial intelligence methods for its ability to manage ambiguity and allowing reasoning processes under uncertainty. In control problems, the artificial intelligence methods are widely used to reach different objectives, such as plant and controller modeling and design, scaling gains calibration and parameterization, system performance improvement, among others [4].

The LAMDA technique, as described in Section 2, bases its operation on fuzzy logic. Therefore, it has been selected for

the development of a new controller due to the following features:

- It is a non-iterative identification algorithm.
- It works in supervised and unsupervised tasks.
- Low computational complexity (non-iterative algorithm).
- Its internal structure is known, i.e., it is a white box where all its parameters can be accessed.
- Easy implementation in programing terms; it does not involve complex routines or mathematical operations.

The fixed hidden layers of LAMDA is an additional advantage over other methods like Artificial Neural Networks (ANN) where the designer must specify this parameter, which is not trivial. Being a fuzzy classification/clustering algorithm, it does not have an inference method because original LAMDA only assigns elements to a class or cluster based on similarity. That is why in this work an inference method is proposed that allows taking a corrective action that takes the system to the desired class from the current class. In other words, the idea is that the controller can take the system from a current functional state to a desired one.

Different LAMDA approaches applied in the field of control systems are presented in detail in the following subsections. Initially, the Rule-based LAMDA proposal is formalized, in which an inference method applied to the algorithm is established, to convert it into a fuzzy controller. With these bases, the LAMDA Sliding-Mode Control (LSMC) controller is detailed, a proposal with robust characteristics and chattering free based on the Lyapunov theory to guarantee

the stability. Then, the Adaptive LAMDA proposal is presented, which can self-adjust its internal parameters to model and control systems based on online learning. Finally, the SMC based on Z-numbers (ZLSMC) approach is formalized, which makes use of the reliability concepts to improve the performance of the controller. Also, it uses the deviation between the reference and the current system output as a measurement criterion.

4.1 Rule-based LAMDA

Once the information of the object defined by its descriptors is available, LAMDA identifies the current state of the system and takes it to the desired state. For this purpose, it is necessary to define rules based on the system knowledge, which is also carried out in conventional fuzzy controllers. The analytic expression that summarizes the fuzzy logic system inference mechanism considering the classes in LAMDA is represented as follows:

$$Rule^{(k)}: IF\ o_1\ is\ F_1^p\ and \dots\ o_j\ is\ F_j^q \dots and$$
$$o_l\ is\ F_l^r\ THEN\ y_k\ is\ G_k \quad (4.1)$$

where $Rule^{(k)}$ is the rule applied for the class (functional state) k, o_j is the descriptor j of the object O, with U_j the universe of discourse that corresponds to the values that each descriptor j can take. The output linguistic variable y_k is defined on a universe of discourse V. $F_j = \{F_j^q : q = 1,2, \dots, Q\}$ is a fuzzy set on U_j with Q the number of linguistic values (fuzzy partitions), and G_k is a fuzzy set on V.

Because LAMDA does not have an inference mechanism, the presented proposal is based on using the GADs. We propose to use the first order TSK inference method [123],

where $G_k = \gamma_k$, with γ_k a constant value (weight) specified for each class which is described in detail in subsection 4.2. To compute the crisp output, it is proposed:

$$u = \Gamma \sum_{k=1}^{m} \gamma_k GAD_{k,\bar{O}} \tag{4.2}$$

where u is the controller output, γ_k is the weight applied in the $k-th$ rule, and Γ is an adjustment parameter to saturate the output. The parameter Γ is computed as:

$$\Gamma = \left| \frac{\text{argamax}(\gamma_k)}{\sum_{k=1}^{m} \gamma_k GAD_{k,\text{argmax}(\bar{O})}} \right| \tag{4.3}$$

where function $GAD_{k,\text{argmax}(\bar{O})}$ returns the value of the GAD calculated for the maximum value of each descriptor.

Finally, replacing (4.3) in (4.2), the crisp control action computed by LAMDA for an object $\bar{O}$ is:

$$u = \left| \frac{\text{argamax}(\gamma_k)}{\sum_{k=1}^{m} \gamma_k GAD_{k,\text{argmax}(\bar{O})}} \right| \sum_{k=1}^{m} \gamma^k GAD_{k,\bar{O}} \tag{4.4}$$

In (4.4), the controller output depends on the GADs and the centers of the classes $\rho_{k,j}$ (used for the calculation of the MADs) defined in the training (design) stage, which remains fixed during the operation of LAMDA as a controller. The scheme of the LAMDA controller is shown in Figure 4.1 [124], which has three layers. The number of nodes in each layer depends on the number of descriptors and their fuzzy sets. Based on the fact that all descriptors are considered to have the same number of classes "c," the total number of classes is $m = c^l$ (with l: the number of descriptors of one

object, see (2.1)) and the number of nodes in each layer is, for layer 1: lc nodes, for layers 2: m nodes, and for layer 3: 1 node. Note that using LAMDA as controller, it is not consider the effects of the NIC in the algorithm, since it will always assigns the object O (input data) to one of the previously defined classes.

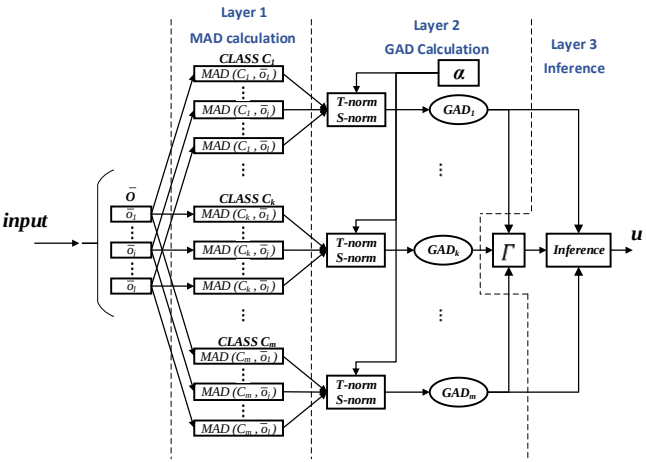


Figure 4.1. Scheme of the proposed LAMDA controller

4.2 LAMDA Sliding-Mode Control (LSMC)

4.2.1 System description and fundamentals of SMC

In this subsection, a class of SISO continuous nonlinear systems with external bounded disturbances is described. These n-th nonlinear systems are represented in state-space as:

$$\dot{x}_i(t) = x_{i+1}(t), \quad i = 1, \dots, n-1$$
$$\dot{x}_n(t) = A(X(t),t) + b(X(t),t)u(t) + d(t) \qquad (4.5)$$

where $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T = \left[x(t), \dot{x}(t), \dots, x^{(n-1)}(t)\right]^T \in \Re^n$ is the state vector of the system, which is measurable, $A(X(t), t)$ and $b(X(t), t)$ are not exactly known, continuous and nonlinear bounded functions, such that $|A(X(t), t)| \leq \beta_A$, with β_A the upper/lower bound of the function $A(X(t), t)$, $u(t) \in \Re$ is the control input, the upper/lower bound of the disturbance $d(t) \in \Re$ is β_d, such that $|d(t)| \leq \beta_d$. $b(X(t), t)$ is upper and lower bounded such as $0 < b < |b(X(t), t)| < \bar{b}$. [122].

The control objective is to design a control law for the system state $X(t)$ to track the desired state trajectory $X_d(t)$ in the presence of external disturbances and model uncertainties.

Considering the desired state trajectory $X_d(t)$ as:

$$X_d(t) = [x_{d1}(t), x_{d2}(t), \dots, x_{dn}(t)]^T$$
$$X_d(t) = \left[x_d(t), \dot{x}_d(t), \dots, x_d^{(n-1)}(t)\right]^T \quad (4.6)$$

The tracking error is defined as:

$$E(t) = X_d(t) - X(t)$$
$$= [x_{d1}(t), x_{d2}(t), \dots, x_{dn}(t)]^T - [x_1(t), x_2(t), \dots, x_n(t)]^T$$
$$= [e_1(t), e_2(t), \dots, e_n(t)]^T = \left[e(t), \dot{e}(t), \dots, e^{(n-1)}(t)\right]^T \quad (4.7)$$

Then, the controller is designed, such that for any desired state $X_d(t)$, the resulting tracking error vector satisfies:

$$\lim_{t \to \infty} \|E(t)\| = \lim_{t \to \infty} \|X_d(t) - X(t)\| = 0 \quad (4.8)$$

where $\|\cdot\|$ is the Euclidean norm of the vector

The idea behind SMC is to define a continuous surface in which the process can slide to its desired state trajectory.

The selected sliding surface is an integro-differential equation addressed in [125], defined as:

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^n \int e(t)\, dt \tag{4.9}$$

where n is the system order, and λ is a strictly positive constant that helps define the sliding hyperplane.

The control objective is to satisfy (4.8); that is, the system state is equal to the desired state as the time tends to infinite. When this happens, (4.9) reaches a constant value and the system is in the sliding mode satisfying $\dot{s}(t) = 0$. Therefore, the equivalent (continuous) control law, from (4.5) is computed as:

$$u_c = \frac{1}{b(X(t), t)}\left[-A(X(t), t) - d(t) - \dot{X}_d(t)\right] \tag{4.10}$$

Based on the Lyapunov stability theory, a Lyapunov function V can be defined as:

$$V(s(t)) = \frac{1}{2}s(t)^2 \tag{4.11}$$

Moreover, the derivative of V is:

$$\dot{V}(s(t)) = s(t)\dot{s}(t) = s(t)\left[\left(\frac{d}{dt} + \lambda\right)^n e(t)\right] \tag{4.12}$$

If (4.12) is negative for all $s(t) \neq 0$, then the reaching condition is obtained. Particularly, the control action u is designed to guarantee that the states are hitting on the sliding surface [126]. In the conventional SMC, the reaching (discontinuous) control law u_d is defined as [127]:

$$u_d = K_D sign(s(t)) \tag{4.13}$$

where K_D is the switching gain. According to Lyapunov theory, the state vector of the system approaches the hyperplane if $\dot{V} \leq -K_D|s|$. Thus, control action u is:

$$u = u_c + u_d \tag{4.14}$$

4.2.2 The proposed approach of LSMC

Considering the system presented in (4.5), a traditional SMC controller can be designed if the plant model is known. The LSMC method is focused on designing a control law that can be applied in that class of SISO continuous nonlinear systems whose parameters are variable, uncertain, or cannot be accurately defined. To achieve this goal, two points are raised in this document:

- Based on the fundamentals of the SMC, select a suitable sliding surface [117].
- Apply LAMDA to reach and maintain the system on the sliding surface in the presence of uncertainties in the model and external disturbances, eliminating the phenomenon known as chattering that occurs in the conventional SMC [112].

The following procedure is used to compute u_c and u_d with LAMDA for the system defined in (4.5):

The sliding surface shown in (4.9) is selected; developing that mathematical expression:

$$s(t) = \left(\frac{d^n}{dt} + r_{n-1}\lambda\frac{d^{n-1}}{dt} + r_{n-2}\lambda^2\frac{d^{n-2}}{dt} + \cdots + r_1\lambda^{n-1}\frac{d}{dt} + \lambda^n\right)\int e(t)\,dt \tag{4.15}$$

where $\{r_{n-1}, r_{n-2}, \ldots, r_1\}$ are the terms obtained by solving the polynomial of (4.9) with power n.

Solving (4.15), the following expression is obtained:

$$s(t) = \frac{d^{n-1}e(t)}{dt} + r_{n-1}\lambda\frac{d^{n-2}e(t)}{dt} + r_{n-2}\lambda^2\frac{d^{n-3}e(t)}{dt} + \cdots + r_1\lambda^{n-1}e(t) + \lambda^n\int e(t)\,dt \quad (4.16)$$

Then, the derivative of (4.16) becomes:

$$\dot{s}(t) = \frac{d^n e(t)}{dt} + r_{n-1}\lambda\frac{d^{n-1}e(t)}{dt} + r_{n-2}\lambda^2\frac{d^{n-2}e(t)}{dt} + \cdots + r_1\lambda^{n-1}\frac{de(t)}{dt} + \lambda^n e(t)$$

$$= e^{(n)}(t) + r_{n-1}\lambda e^{(n-1)} + r_{n-2}\lambda^2 e^{(n-2)} + \cdots + r_1\lambda^{n-1}\dot{e}(t) + \lambda^n e(t)$$

$$= e^{(n)}(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)} \quad (4.17)$$

with $r_0 = 1$ and $e^{(0)}(t) = e(t)$. From (4.7):

$$e^{(n)}(t) = \dot{x}_{dn}(t) - \dot{x}_n(t) \quad (4.18)$$

Replacing (4.5) in (4.18) :

$$e^{(n)}(t) = \dot{x}_{dn}(t) - A(X(t), t) - b(X(t), t)u - d(t) \quad (4.19)$$

and replacing (4.19) in (4.17) it is obtained:

$$\dot{s}(t) = \dot{x}_{dn}(t) - A(X(t), t) - b(X(t), t)u - d(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)} \quad (4.20)$$

Considering the continuous control law $u = u_c$ in (4.20), the following is obtained:

$$\dot{s}(t) = \dot{x}_{dn}(t) - A(X(t), t) - b(X(t), t)u_c - d(t)$$

$$+\sum_{i=1}^{n} r_{n-i}\lambda^{i}e^{(n-i)} \qquad (4.21)$$

The objective is to satisfy $\dot{s}(t) = 0$ with the control action u_c. For the design of the controller, it is necessary only to know the sign of $b(X(t), t)$, in order to establish the rules based on the classes of LAMDA.

In this work, we chose five classes that define the functional states of $\dot{s}(t)$. These are NB: Negative Big, NS: Negative Small, ZE: zero, PS: Positive Small and PB: Positive Big, values used to establish the rules that allow to compute u_c required to satisfy $\dot{s}(t) = 0$. In sub-Section 5.2.1.2.1, a brief sensitivity analysis of the number of classes is presented, to determine how this parameter affects the controller performance.

For simple handling of the classes and the control output, the classes are standardized between [-1,1] [121]: $NB = -1$, $NS = -0.5$, $ZE = 0$, $PS = 0.5$ and $PB = 1$. These values have been chosen initially to establish the rules that define the behavior of the controller. However, for proper calibration, the gain k_1 is used for the input $\dot{s}(t)$ (as shown in Figure 4.2), and for the control output, is proposed:

$$u_c = k_c u_{nc} \qquad (4.22)$$

$$u_c = k_c LSMC(\dot{s})\,; \quad k_c > 0 \qquad (4.23)$$

where $u_{nc} \in [-1,1]$ is the normalized control action of the continuous part, and k_c is its corresponding scaling gain.

Assuming that $b(X(t), t) > 0$, based on (4.21), it is noted that $\dot{s}(t)$ decreases as u_c increases, and $\dot{s}(t)$ increases as u_c decreases. This information is sufficient for the definition

of rules that allow obtaining $\dot{s}(t) = 0$. For instance, if $\dot{s}(t)$ is PB, then large positive control action u_c is needed in order to decrease quickly $\dot{s}(t)$. If $\dot{s}(t) = ZE$ (desired condition), then no control action is required, thus $u_c = ZE$.

Based on this analysis, it has been designed the rule table corresponding to the continuous part control actions, which is shown in Table 4.1. Unlike [111,112,122], LSMC does not require the complete model of the system to be controlled since, based on LAMDA, the controller is designed for the continuous part.

Table 4.1. Rule table of LSMC for $\dot{s}(t)$

	$\dot{s}(t)$				
	NB	***NS***	***Z***	***PS***	***PB***
$b(X(t),t) > 0$	$\gamma_1 = NB$	$\gamma_2 = NS$	$\gamma_3 = ZE$	$\gamma_4 = PS$	$\gamma_5 = PB$
$b(X(t),t) < 0$	$\gamma_1 = PB$	$\gamma_2 = PS$	$\gamma_3 = ZE$	$\gamma_4 = NS$	$\gamma_5 = NB$

Now, It is necessary to compute the control action u_d that attracts the states of the system towards the sliding surface. For this, it is selected the Lyapunov function in (4.11).

The derivative of (4.11) becomes:

$$\dot{V}(s(t)) = s(t)\dot{s}(t) \tag{4.24}$$

Here, based on the Lyapunov stability theory, it is necessary to satisfy the following condition:

$$s(t)\dot{s}(t) < 0 \tag{4.25}$$

Replacing (4.21) in (4.25) considering only the discontinuous part of the control $u = u_d$:

$$s(t)\dot{s}(t) = s(t)\dot{x}_{dn}(t) - s(t)A(X(t),t) - s(t)b(X(t),t)u_d -$$

$$s(t)d(t) + s(t)\sum_{i=1}^{n} r_{n-i}\lambda^{i}e^{(n-i)} < 0 \quad (4.26)$$

In (4.26), if $s(t)\dot{s}(t)$ is negative for all $s(t) \neq 0$, then the existence of the sliding mode is guaranteed [121], that is, the states of the system are attracted from any initial state to the sliding surface.

For the computation of u_d, two cases can be analyzed depending on the sign of $b(X,t)$. From (4.26), assuming $b(X(t),t) > 0$:

- If $s(t) > 0$ and u_d increases, then the product $s(t)\dot{s}(t)$ decreases and vice versa.
- If $s(t) < 0$ and u_d increases, then the product $s(t)\dot{s}(t)$ increases, and if u_d decreases, then $s(t)\dot{s}(t)$ decreases.

From this analysis, it is proposed to generate a control action u_d to satisfy $s(t)\dot{s}(t) < 0$.

Five classes are established for $s(t)$ and $\dot{s}(t)$, defined as NB, NS, ZE, PS, PB (the same for u_c). Due to the normalization of the classes, the gain k_2 is added (as shown in Figure 4.2), which is used for the input $s(t)$, and the control output is:

$$u_d = k_d u_{nd} \quad (4.27)$$

$$u_d = k_d LSMC(s,\dot{s}) \quad ; \quad k_d > 0 \quad (4.28)$$

where $u_{nd} \in [-1,1]$ is the normalized control action of the discontinuous part, and k_d is its scaling factor.

The rule table for the discontinuous control action is shown in Table 4.2 for the case $b(X(t),t) > 0$.

Table 4.2. Rule table of LSMC for $s(t)$ and $\dot{s}(t)$ with $b(X(t),t) > 0$

		$\dot{s}(t)$				
		NB	***NS***	***ZE***	***PS***	***PB***
$s(t)$	***PB***	$\gamma_5 = ZE$	$\gamma_{10} = ZE$	$\gamma_{15} = PS$	$\gamma_{20} = PB$	$\gamma_{25} = PB$
	PS	$\gamma_4 = ZE$	$\gamma_9 = ZE$	$\gamma_{14} = PS$	$\gamma_{19} = PB$	$\gamma_{24} = PB$
	ZE	$\gamma_3 = NB$	$\gamma_8 = NS$	$\gamma_{13} = ZE$	$\gamma_{18} = PS$	$\gamma_{23} = PB$
	NS	$\gamma_2 = NB$	$\gamma_7 = NB$	$\gamma_{12} = NS$	$\gamma_{17} = ZE$	$\gamma_{22} = ZE$
	NB	$\gamma_1 = NB$	$\gamma_6 = NB$	$\gamma_{11} = NS$	$\gamma_{16} = ZE$	$\gamma_{21} = ZE$

The proposed method to design the rule table is as follows:

- Positioning in the row of $s(t) = 0$, bring $\dot{s}(t)$ to zero using the same rules presented in Table 4.1, considering for this example $b(X(t),t) > 0$.
- In the class C_1, where $s(t) = NB$ and $\dot{s}(t) = NB$, the product $s(t)\dot{s}(t)$ is PB. Therefore, based on (4.26) considering $s(t) < 0$, a negative control input is required $(u_d = NB)$ to quickly decrease $s(t)\dot{s}(t)$. The same situation is presented in the classes C_2, C_6, C_7.
- In the class C_5, where $s(t) = PB$ and $\dot{s}(t) = NB$, the product $s(t)\dot{s}(t)$ is NB. Therefore, no change in the control action u_d is required, thus $u_d = ZE$. The other classes in which no change in the control action is required because the condition is met are: C_4, C_9, C_{10}, C_{16}, C_{17}, C_{21} and C_{22}.
- In the class C_{25}, where $s(t) = PB$ and $\dot{s}(t) = PB$, the product $s(t)\dot{s}(t)$ is PB. Therefore, based on (4.26) considering $s(t) > 0$, large positive control input is

required $(u_d = PB)$ to quickly decrease $s(t)\dot{s}(t)$. The same situation is presented in the classes C_{19}, C_{20}, C_{24}.

- In the class C_{11}, where $s(t) = NB$ and $\dot{s}(t) = ZE$, the product $s(t)\dot{s}(t)$ is ZE. Therefore, based on (4.25) with $s(t) < 0$, to ensure that the condition $s(t)\dot{s}(t) < 0$ is always met, a control action $u_d = NS$ is applied. The same situation is presented in the class C_{12}.
- In the class C_{15}, where $s(t) = PB$ and $\dot{s}(t) = ZE$, the product $s(t)\dot{s}(t)$ is ZE. Therefore, based on (4.25) with $s(t) > 0$, to ensure that the condition $s(t)\dot{s}(t) < 0$ is always met, a control action $u_d = PS$ is applied.

LSMC removes the chattering present in the conventional SMC, replacing the discontinuous sign function using the fuzzy logic rules and the classes defined in LAMDA. For the case when $b(X(t), t) < 0$, Table 4.3 is obtained with a similar analysis as the one presented before. Finally, the overall control action is computed as:

$$u = k_c LSMC(\dot{s}) + k_d LSMC(s, \dot{s}) \qquad (4.29)$$

Table 4.3. Rule table of LSMC for $s(t)$ and $\dot{s}(t)$ with $b(X(t), t) < 0$

		$\dot{s}(t)$				
		NB	***NS***	***ZE***	***PS***	***PB***
$s(t)$	***PB***	$\gamma_5 = ZE$	$\gamma_{10} = ZE$	$\gamma_{15} = NS$	$\gamma_{20} = NB$	$\gamma_{25} = NB$
	PS	$\gamma_4 = ZE$	$\gamma_9 = ZE$	$\gamma_{14} = NS$	$\gamma_{19} = NB$	$\gamma_{24} = NB$
	ZE	$\gamma_3 = PB$	$\gamma_8 = PS$	$\gamma_{13} = ZE$	$\gamma_{18} = NS$	$\gamma_{23} = NB$
	NS	$\gamma_2 = PB$	$\gamma_7 = PB$	$\gamma_{12} = PS$	$\gamma_{17} = ZE$	$\gamma_{22} = ZE$
	NB	$\gamma_1 = PB$	$\gamma_6 = PB$	$\gamma_{11} = PS$	$\gamma_{16} = ZE$	$\gamma_{21} = ZE$

The LSMC scheme is shown in Figure. 4.2, where the blocks of the controller applied in the continuous and discontinuous part are shown, the descriptors used in each LAMDA block, the scaling gains in the inputs and the outputs, and the block corresponding to obtain the sliding surface.

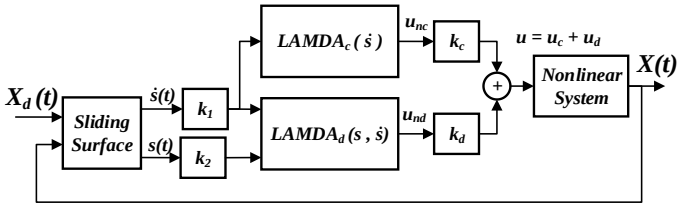


Figure 4.2. Block diagram of the LSMC.

4.2.3 Stability Analysis

The proposed Lyapunov function V is defined in (4.11) and its derivative $\dot{V}$ is presented in (4.24).

To guarantee the stability of the system, the derivative $\dot{V}$ must satisfy the condition $\dot{V} < 0$. For our case, this condition is presented in (4.25) [76]. Therefore, replacing in (4.25) the system presented in (4.5), controlled by $u(t)$ defined in (4.29), with u_c defined in (4.23) and u_d defined in (4.28), it is obtained:

$$\dot{V} = s(t)\left(\dot{x}_{dn}(t) - A(X(t),t) - b(X(t),t)\left(k_c LSMC(\dot{s}) + k_d LSMC(s,\dot{s})\right) - d(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}\right) < 0 \qquad (4.30)$$

In order to prove the stability of the proposed method, from (4.30), it is considered that $\dot{x}_{dn}(t)$ and $\sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}$ are continuous and bounded [111]:

$$|\dot{x}_{dn}(t)| \leq \beta_{dn} \tag{4.31}$$

$$\left|\sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}\right| \leq \beta_e \tag{4.32}$$

where β_{dn} and β_e are unknown positive constants.

Theorem 1. Consider the system presented in (4.5), controlled by $u(t)$ defined in (4.29), where u_c is defined in (4.23), u_d is defined in (4.28) and $(k_c + k_d) > \beta_{dn} + \beta_e - (\beta_A + \beta_d)$. Then, the error state trajectory converges to the sliding surface $s(t) = 0$.

Proof. The stability demonstration can be addressed in two cases, based on the sign of $b(X,t)$, as follows:

- Case 1: Assuming for simplicity $b(X(t),t) = -1$, without loss of generality for systems in which $b(X(t),t) < 1$:

$$\dot{V} = s\left(\dot{x}_{dn}(t) - A(X(t),t) + k_c LSMC(\dot{s}) + k_d LSMC(s,\dot{s}) - d(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}\right) \tag{4.33}$$

From [72] and Table 4.3, for $b(X(t),t) < 0$, it is demonstrated that $k_d u_d = -k_d|s|$, and $k_c u_c = -k_c|s|$, e.g., u_d has the opposite sign of $\dot{s}$. Thus, replacing (4.31) and (4.32) in (4.33):

$$\dot{V} = s\left(\dot{x}_{dn}(t) - A(X(t),t) + k_c LSMC(\dot{s}) + k_d LSMC(s,\dot{s}) - d(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}\right)$$

$$\leq \beta_{dn}|s| - \beta_A|s| - k_c|s| - k_d|s| - \beta_d|s| + \beta_e|s| \quad (4.34)$$

$$\dot{V} \leq \beta_{dn}|s| - \beta_A|s| - k_c|s| - k_d|s| - \beta_d|s| + \beta_e|s|$$

$$= [-(k_c + k_d) + (\beta_{dn} + \beta_e - \beta_A - \beta_d)]|s| \quad (4.35)$$

Therefore, to fulfill that $s(t)\dot{s}(t) < 0$, it is required to satisfy:

$$(k_c + k_d) > \beta_{dn} + \beta_e - (\beta_A + \beta_d) \quad (4.36)$$

- Case 2: Assuming for simplicity $b(X(t),t) = 1$, without loss of generality for systems in which $b(X(t),t) > 0$:

$$\dot{V} = s\left(\dot{x}_{dn}(t) - A(X(t),t) - k_c LSMC(\dot{s}) - k_d LSMC(s,\dot{s}) - d(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}\right) \quad (4.37)$$

From [72] and Table 4.2, for $b(X(t),t) > 1$, it is demonstrated that $k_d u_d = k_d|s|$ and $k_c u_c = k_c|s|$. Thus, replacing (4.31) and (4.32) in (4.37):

$$\dot{V} = s\left(\dot{x}_{dn}(t) - A(X(t),t) - k_c LSMC(\dot{s}) - k_d LSMC(s,\dot{s}) - d(t) + \sum_{i=1}^{n} r_{n-i}\lambda^i e^{(n-i)}\right)$$

$$\leq \beta_{dn}|s| - \beta_A|s| - k_c|s| - k_d|s| - \beta_d|s| + \beta_e|s| \quad (4.38)$$

$$\dot{V} \leq \beta_{dn}|s| - \beta_A|s| - k_c|s| - k_d|s| - \beta_d|s| + \beta_e|s|$$

$$= [-(k_c + k_d) + (\beta_{dn} + \beta_e - \beta_A - \beta_d)]|s| \qquad (4.39)$$

It can be seen that (4.35) is equal to (4.39). Therefore, if selected $(k_c + k_d) > \beta_{dn} + \beta_e - (\beta_A + \beta_d)$, then it is concluded that the reaching condition $s(t)\dot{s}(t) < 0$ is always satisfied. Therefore, the proof is achieved completely. The two cases analyzed are sufficient for the proof of the theorem, since as previously analyzed, the only parameter to be known is the sign of the function $b(X(t), t)$, which, as shown in (4.33) and (4.37) does not affect the other variables and can be analyzed in a general manner for cases 1 and 2.

4.2.4 LSMC scaling gains offline calibration

The LSMC scaling gains (k_1, k_2, k_c, k_d) have not a formalized equations for proper calibration and we will consider its mathematical formalization in a future work, however, the parameter λ can be calibrated with the method presented in [127]. Based on this consideration, this work proposes two calibration methods for the LSMC:

- Heuristic (empirical) calibration
- Offline calibration using Particle Swarm Optimization (PSO)

These two methods have been used to make a comparison and analyze if there is a considerable improvement in the performance of the controller by performing simulation and experiments.

4.2.4.1 Heuristic calibration

The heuristic calibration used in this work is based on the trial and error method, which, in contrast to the quantitative tuning procedures, where the numerical values for the

configuration of the parameters of the controllers are obtained through the collection of data and analysis, a "heuristic" or "trial and error" tuning procedure is one where general rules are followed to obtain approximate or qualitative results according to the system requirements [128]. Most control loops, especially in the case of PID or control schemes that do not have formulas for their calibration, have been tuned with this method.

A true heuristic tuning method can be executed if one is aware of the characteristic of the process to be controlled, knowing the applicability of the control actions based on their respective scaling gains. Simply experimenting with random values for the scaling gains, it is very tedious at best and dangerous at worst if we do not understand what each type of control action causes in the process, and its limitations. However, the heuristic method can be implemented and tested in simulations before going to the real plant, which is an advantage in order to validate this methodology for the tuning of controllers.

In this work, the LSMC controller has been calibrated based on the measurement and reduction of the Integral Square Error “ISE” index (4.43). This index is used because it integrates the square of the error over time, penalizing large errors more than smaller ones. Therefore, the controller that obtains the minimum index performs the best.

$$ISE = \int_0^t e^2(t)dt \tag{4.43}$$

The method also involves observing the quality of the control action, seeking that it is within the energy limits allowed by the actuator and observing the output of the system in order to obtain an adequate response from qualitative point of view.

The previously detailed heuristic calibration scheme is shown in Figure 4.3.

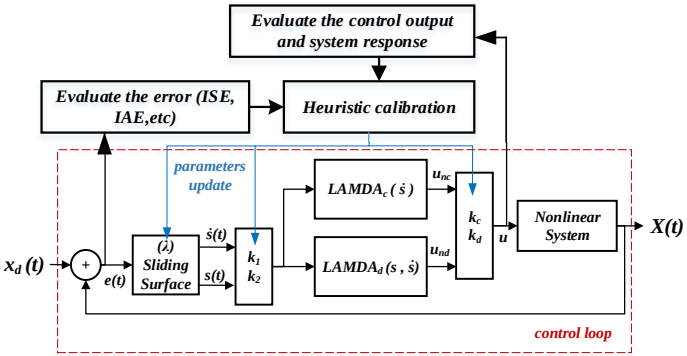


Figure 4.3. Offline optimization of the scaling gains scheme

4.2.4.2 Particle Swarm Optimization (PSO)

The evolutionary algorithm PSO is an intelligence algorithm based on swarms [5] related to the social behavior of animals, such as a group of birds or a swarm of fishes. It uses the animal behavior from three point of views: its habits, its memory capacity and its cooperation capability.

In PSO, the population is the number of particles in the space, which are initialized in a random way. The performance of a particle is measured by an adjustment value, which is specific to the problem. Thus, each particle will have an adjustment value, which will be evaluated by an adjustment function that will be optimized in each generation.

The movement of the different particles is coordinated by a speed that has magnitude and direction. Each particle

position in any instance of time is influenced by its best position and the position of the best particle. Thus for each particle, is known its best position, also called "best local", and the best position of all the group of particles, called "best global". In each generation, the velocity and position of the particles are updated as follows [5]:

$$\mathcal{V}_i^{new} = Dec(t)w_{i0}\mathcal{V}_i + Random_1 w_{i1}(b_i - \mathcal{X}_i) + Random_2 w_{i2}\left(b_g - \mathcal{X}_i\right) \quad (4.40)$$

$$\mathcal{X}_i^{new} = \mathcal{X}_i + V_i^{new} \quad (4.41)$$

Where $\mathcal{X}_i$ is the current position and b_i is the best position of the particle. b_g is the best global position, $\mathcal{V}_i$ is the velocity of each particle, $Dec(t)$ is a decreasing function, and w_{i0}, w_{i1} ,w_{i2} are the weights of each component. The PSO is summarized in Figure 4.4.

The optimization criterion for the calibration (fitness function in the PSO) for this work is the minimization of the ISE.

Figure 4.5 shows the scheme for the offline optimization of the scaling gains of the LSMC, in which the control loop is simulated n iterations until the ISE is minimized, meeting the stop criterion. The restriction considered is the controller output (values minimum and maximum), which will avoid its saturation.

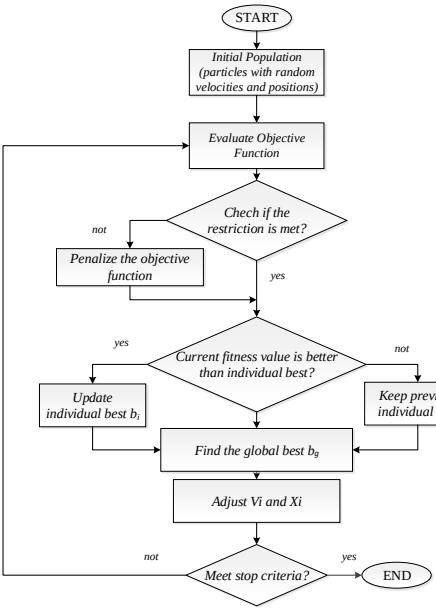


Figure 4.4. Offline optimization of the scaling gains scheme

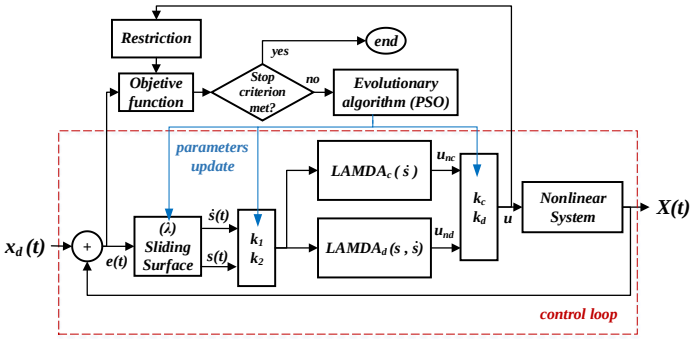


Figure 4.5. Offline optimization of the scaling gains scheme

4.3 Adaptive LAMDA

In general, the design of the Rule-based LAMDA and LSMC can be a time-consuming process, depending on the system to be controlled, especially in the stages of scaling gains tuning and in the definition of the classes and rules. Also, these controllers do not have the ability to automatically learn or be adaptive, because their internal values (class centers, exigency degree, and consequent parameters) are set at the design stage, and they do not change during the operation of the control in the process. In order to solve these problems, we propose a new method called Adaptive LAMDA.

The background presented in section 2.3, have motivated us in this subsection to propose a new approach based on LAMDA. The research contribution consists in proposing an adaptive learning method for the LAMDA parameters update, which allows controlling a system through the detection of functional states, the theory on which this algorithm is based, without requiring the process model in detail.

Regarding the previous LAMDA controllers the advantages of this proposal are the following:

- The implementation of LAMDA as an identifier is proposed for the first time, handling the concept of self-adjustment of the exigency (α) and the antecedent parameters used for the GADs calculation.
- A stability analysis of the learning algorithm is proposed to guarantee a rapid convergence of the estimated output towards the desired output.

- The proposed scheme, with respect to approaches such as those presented in [83,129,130], does not requires to compute the output/input gradient of the system to be controlled, which reduces the computational cost.
- This approach has a known number of hidden layers, which is an advantage with respect to algorithms such as those presented in [9,131], avoiding the heuristic definition of the number of internal layers.
- The proposed learning for LAMDA is based on a hybrid learning, which allows a quick convergence to the desired output, improving the learning time and preventing that solutions be trapped in local minima. This is a great advantage over learning methods that only work with gradient descent, which is generally slow [14].
- The modeling and control of nonlinear systems are based on the concept of classes or functional states established by the LAMDA theory.

4.3.1 Adaptive LAMDA Model

The original LAMDA presents interesting results in classification and clustering applications, however, for modeling and control, the algorithm needs to work as a regressor with the feature of online self-adjustment of parameters, for which the addition of layers and a different learning method is required. In this work, the addition of a first-order T-S fuzzy inference system to LAMDA is proposed, due to the excellent results that this method presents for modeling and control [132]. This methodology establishes that the output of each class is represented as a linear combination of input descriptors, plus a constant

parameter. Finally, the last output is the weighted average of each class output.

The implementation of the T-S fuzzy inference system applied to LAMDA requires the addition of layers 3, 4 and 5 to the original model presented in Figure 2.1. Figure 4.4 shows the scheme for the adaptive model, which takes an individual $\bar{O}$ for the computation of the outputs of each layer.

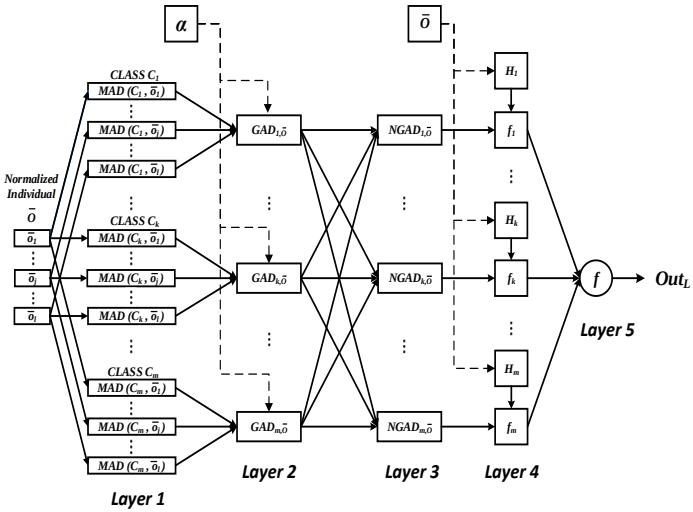


Figure 4.6. Adaptive scheme for LAMDA

The scheme of Figure 4.6 corresponds to a MISO (Multiple-Input Single-Output) system, with 5 layers, each one with a specific function:

Layer 1: each node in this layer computes the $MAD_{k,j}(\bar{o}_j, \rho_{k,j}, \sigma_{k,j})$ of each descriptor j in each class k, as described in (2.5). The set of parameters $\{\rho_{k,j}, \sigma_{k,j}\}$ must be optimized, changing the bell shape by adjusting the classes

of the model. These parameters are known as the premise parameters of the LAMDA structure.

Layer 2: Each node in this layer computes the $GAD_{k,\bar{O}}$ of each class k through the aggregation functions and the exigency α (see (2.7)). This parameter must be optimized, changing the exigency degree for the classes of the model, and therefore, the linear interpolation between the t-norm and t-conorm, which affects the behavior of the GADs.

Layer 3: In this node, the normalization of each GAD is computed, with respect to the sum of all the GAD for each class. The normalization is performed by:

$$NGAD_{k,\bar{O}}\left(GAD_{1,\bar{O}},\dots,GAD_{k,\bar{O}},\dots,GAD_{m,\bar{O}}\right)=\frac{GAD_{k,\bar{O}}}{\sum_{k=1}^{m}GAD_{k,\bar{O}}} \quad (4.43)$$

Layer 4: Each node of this layer corresponds to the result of multiplying the $NGAD_{k,\bar{O}}$ with a first-order T-S function $H_k(\cdot)$ for the class k that uses the descriptors of the analyzed individual, and it is defined by (4.44). This function has $n+1$ parameters, that is, it depends on the number of descriptors of $\bar{O}$. These values are known as consequent parameters.

$$H_k\left(\bar{O},h_{k1},\dots,h_{kj},\dots,h_{kn},h_k\right)= \bar{o}_1 h_{k1}+\cdots+\bar{o}_j h_{kj}+\cdots+\bar{o}_l h_{kl}+h_k \quad (4.44)$$

and the output of layer 4 is computed by:

$$f_k\left(NGAD_{k,\bar{O}},H_k\right)=NGAD_{k,\bar{O}}H_k \quad (4.45)$$

Layer 5: This layer has only one node, which computes the sum of all the inputs, returning the value Out_L:

$$Out_L(f_1,\dots,f_k,\dots f_m)=\sum_{k=1}^{m}f_k \quad (4.46)$$

Using the previous expressions, the construction of an Adaptive LAMDA algorithm based on the T-S inference is proposed. This model must adjust the premise parameters that correspond to the calculation of the $MAD_{k,j}$, such as: $\rho_{k,j}, \sigma_{k,j}$, the exigency parameter α, and the consequent parameters in the functions: $H_1, \dots, H_k, \dots, H_m$.

The number of nodes in each layer depends on the number of descriptors and the selected fuzzy sets. Based on the fact that all descriptors are considered to have the same number of classes "c", the total number of classes is $m = c^l$, and the number of nodes in each layer is, for layer 1: (lc) nodes, for layers 2, 3, 4: m nodes, and for layer 5: 1 node.

4.3.1.1 Hybrid Learning Algorithm

In the adaptive LAMDA, each node fulfills a function in a unidirectional manner. Some of these nodes have parameters that are adapted as a result of the learning process based on the input and output data. In this process, the hybrid learning has been considered. It consists of a step forward and a step backward that considerably improves the learning time, preventing that solutions be trapped in local minima [133], [134].

The proposed learning has been studied in different works where adaptive networks are designed [77,78]. In the first stage, a forward pass is carried out with the least-squares estimate (LSE) method to adjust the consequent parameters, then a backward pass is performed using the gradient descent (GD) algorithm to adjust the antecedent parameters. The scheme of the hybrid learning is presented in Figure 4.7, detailing the two steps for parameters update.

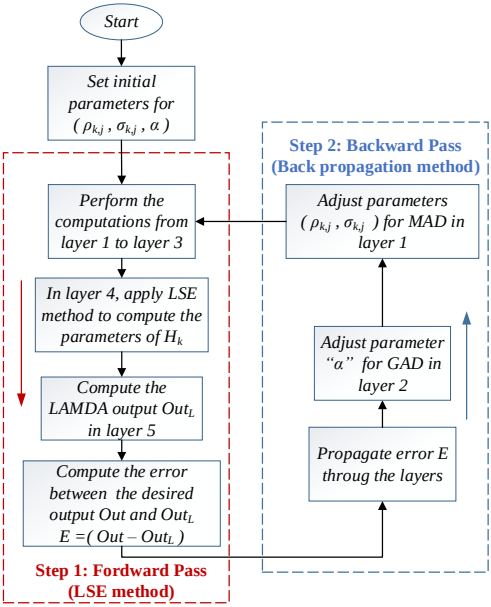


Figure 4.7. Hybrid Learning Scheme

- Forward pass

In the forward pass, the learning algorithm keeps fixed the antecedent parameters $\theta = \{\rho_{k,j}, \sigma_{k,j}, \alpha\}$ required for the calculation of $MAD_{k,j}$ and $GAD_{k,\bar{O}}$ in the layers 1 and 2, respectively, and the process goes forward until the calculation of the nodes $NGAD_{k,\bar{O}}$ in layer 3.

Layer 4 requires the consequent parameters for all the classes $C = \{C_1, \ldots, C_k, \ldots, C_m\}$. Thus, the LSE is used for the adjustment, considering that the function $H_k(\cdot)$ is linear in the consequent parameters. To demonstrate this, (4.44) is developed, considering the d-th individual $\bar{O}^d = [\bar{o}_1^d, \ldots, \bar{o}_j^d, \ldots, \bar{o}_l^d]$ that produces the output Out_L^d:

$$Out_L^d = f_1 + \cdots + f_k + \cdots + f_m \quad (4.47)$$

Expressing (4.47) on the terms of $H(\cdot)$:

$$Out_L^d = NGAD_{1,\bar{O}^d} H_1 + \cdots + NGAD_{k,\bar{O}^d} H_k + \cdots + NGAD_{m,\bar{O}^d} H_m \quad (4.48)$$

With:

$$H_k = \{h_{k1}, .., h_{kj}, ... h_k\} \quad (4.49)$$

where $H_k \in R$ is linear in the consequent, for all the desired values at the output $Out = [Out^1 \ldots Out^d \ldots Out^D]^T$. Thus (4.49) can be rewritten as:

$$Out = Ah \quad (4.50)$$

If A is non-invertible, the pseudoinverse must be computed with (4.51), which minimizes the difference ($\|Ah - O\|^2$):

$$h = (A^T A)^{-1} A^T Out \quad (4.51)$$

The fact that an inverse matrix must be calculated makes (4.51) computationally expensive (in our case it depends on the class number, descriptors and output data used in the training). For this reason, a sequential method is used to calculate h.The recursive method applied to time-varying systems uses the d-th row vector of matrix A, defined by a^T, and the d-th element of Out, defined by Out^d. Then, h is iteratively computed using the covariance matrix $P(t+1)$ as follows:

$$P(k+1) = \frac{1}{\lambda}\left[P(k) - \frac{P(k)a(k+1)a^T(k+1)P(k)}{\lambda + a^T(k+1)P(k)a(k+1)}\right] \quad (4.52)$$

$0 < \lambda \leq 1$ is the forgetting factor and is chosen close to 1 to achieve stability [135].

Finally, $h(k+1)$ is computed by:

$h(k+1) =$

$h(k) + P(k+1)a(k+1)[Out(k+1) - a^T(k+1)h(k)]$ (4.53)

where $d = \{1, \dots, D-1\}$.

According to LAMDA model, if a data output set $Out = [Out^1 \dots Out^d \dots Out^D]^T$ is available, then a supervised learning process can be carried out, propagating backward the error from layer 5 to layer 1 by the chain rule, after computing the consequent parameters $h(t+1)$. Considering that Out^d is the d-th data of the desired outputs Out, and Out_L^d the output calculated by LAMDA corresponding to the individual $\bar{O}^d$, the error in layer 5 is:

$$E_d(k) = \frac{1}{2}[Out^d(k) - Out_L^d(k)]^2 \tag{4.54}$$

For online learning, the aim is to propagate backward the error E_d , through each layer and each node, until obtaining the derivative of the error E_d with respect to the adjustment terms $\theta = \{\rho_{k,j}, \sigma_{k,j}, \alpha\}$ required in (2.5) and (2.7).

In this way, the adjustment of θ in an instant of time $(k+1)$ by the GD is done with (4.55), and the updated with (4.56):

$$\Delta\theta(k) = -\eta\frac{\partial E_d(k)}{\partial\theta(k)} \tag{4.55}$$

$$\begin{aligned}\theta(k+1) &= \theta(k) + \Delta\theta(k) + \beta\big(\theta(k) - \theta(k-1)\big) \\ &= \theta(k) - \eta\frac{\partial E_d}{\partial\theta} + \beta\big(\theta(k) - \theta(k-1)\big)\end{aligned} \tag{4.56}$$

where $\eta \in [0,1]$ corresponds to the learning rate, and $\beta \in [0,1]$ is the momentum term.

The learning process using the GD method through the backpropagation of the error E_d from layer 5 to layer 1 of the scheme presented in Figure 4.6 is:

Layer 5:

$$\epsilon^{(5)} = \frac{\partial E_d}{\partial Out_L^d} = \frac{\partial}{\partial Out_L^d}\left[\frac{1}{2}(Out^d - Out_L^d)^2\right] = -(Out^d - Out_L^d) \quad (4.57)$$

Layer 4: From (4.46), the derivative of Out_L^d with respect to ∂f_k is:

$$\frac{\partial Out_L^d}{\partial f_k} = \frac{\partial[f_1 + \cdots f_k + \cdots f_m]}{\partial f_k} = 1 \ ; \forall k = 1, \ldots, m \quad (4.58)$$

$$\epsilon_k^{(4)} = \frac{\partial E_d}{\partial Out_L^d}\frac{\partial Out_L^d}{\partial f_k} = \epsilon^{(5)} \quad (4.59)$$

Layer 3: From (4.45), the derivative is:

$$\frac{\partial f_k}{\partial NGAD_{k,\bar{o}}} = \frac{\partial\left[\partial NGAD_{k,\bar{o}} \times H_k\right]}{\partial NGAD_{k,\bar{o}}} = H_k \ ; \forall k = 1, \ldots, m \quad (4.60)$$

$$\epsilon_k^{(3)} = \frac{\partial E_d}{\partial Out_L^d}\frac{\partial Out_L^d}{\partial f_k}\frac{\partial f_k}{\partial NGAD_{k,\bar{o}}} = \epsilon^{(5)} H_k; \forall k = 1, \ldots, m \quad (4.61)$$

Layer 2: The partial derivatives of layer 3 are calculated with respect to the outputs of layer 2. Because each node k of layer 3 depends on all the outputs of layer 2, as shown in (4.43), the term k_2 is used to refer to the nodes of layer 2.

$$\frac{\partial NGAD_{k,\bar{o}}}{\partial GAD_{k_2,\bar{o}}} = \begin{cases} \dfrac{\left(\sum_{k_2=1}^{m} GAD_{k_2,\bar{o}}\right) - GAD_{k_2,\bar{o}}}{\left(\sum_{k_2=1}^{m} GAD_{k_2,\bar{o}}\right)^2} & if: k_2 = k \\ -\dfrac{GAD_{k,\bar{o}}}{\left[\sum_{k=1}^{m} GAD_{k_2,\bar{o}}\right]^2} & if: k_2 \neq k \end{cases} \quad (4.62)$$

$$\epsilon_k^{(2)} = \frac{\partial E_d}{\partial o_L^d}\frac{\partial o_L^d}{\partial f_k}\frac{\partial f_k}{\partial NGAD_{k,\bar{O}}}\sum_{k=1}^{m}\frac{\partial NGAD_{k,\bar{O}}}{\partial GAD_{k_2,\bar{O}}} \quad (4.63)$$

$$\epsilon_k^{(2)} = \epsilon^{(5)} H_k \sum_{k=1}^{m}\frac{\partial NGAD_{k,\bar{O}}}{\partial GAD_{k_2,\bar{O}}} \; ; \forall k = 1,\dots,m \quad (4.64)$$

Layer 1: The partial derivatives of layer 2 are computed with respect to the outputs of layer 1. Because the $GADs$ are calculated recursively by (2.7), it is used the term j_1 to refer to each of the nodes of the layer 1, in order to facilitate the mathematical expression of the derivative.

$$\frac{\partial GAD_{k,\bar{O}}}{\partial MAD_{k,j}} = \alpha T\left(MAD_{k,1},\dots,MAD_{k,j_1},\dots,MAD_{k,n}\right) + (1-\alpha)\left(1 - S\left(MAD_{k,1},\dots,MAD_{k,j_1},\dots,MAD_{k,n}\right)\right) \; ; \forall j_1 \neq j \quad (4.65)$$

In (4.65), the derivative of $GAD_{k,\underline{O}}$ respect to $MAD_{k,j}$ is equal to the calculation of the GAD without considering this term. Now, the propagated error in layer 1 is:

$$\epsilon_k^{(1)} = \epsilon_k^{(2)}\frac{\partial GAD_{k,\bar{O}}}{\partial MAD_{k,j}} \quad (4.66)$$

The parameters $\rho_{k,j}, \sigma_{k,j}$ are adjusted in each class k and each descriptor j with equations (4.67) and (4.68), respectively, and α is adjusted for all the model with (4.69).

$$\frac{\partial MAD_{k,j}}{\partial \rho_{k,j}} = \frac{\left(\bar{o}_j - \rho_{k,j}\right)}{\left(\sigma_{k,j}\right)^2}\partial MAD_{k,j} \Longrightarrow \frac{\partial E_d}{\partial \rho_{k,j}} = \epsilon_k^{(1)}\frac{\partial MAD_{k,j}}{\partial \rho_{k,j}} \quad (4.67)$$

$$\frac{\partial MAD_{k,j}}{\partial \sigma_{k,j}} = \frac{\left(\bar{o}_j - \rho_{k,j}\right)^2}{\left(\sigma_{k,j}\right)^3}\partial MAD_{k,j} \Longrightarrow \frac{\partial E_d}{\partial \sigma_{k,j}} = \epsilon_k^{(1)}\frac{\partial MAD_{k,j}}{\partial \sigma_{k,j}} \quad (4.68)$$

$$\frac{\partial GAD_{k,\bar{o}}}{\partial \alpha} = \left[T\left(MAD_{k,1}, \dots, MAD_{k,j}, \dots, MAD_{k,n}\right) - S\left(MAD_{k,1}, \dots, MAD_{k,j}, \dots, MAD_{k,n}\right)\right] \Rightarrow$$

$$\frac{\partial E_d}{\partial \alpha} = \sum_{k=1}^{m} \epsilon_k^{(2)} \frac{\partial GAD_{k,\bar{o}}}{\partial \alpha} \quad (4.69)$$

Finally, the terms are updated as:

$$\rho_{k,j}(k+1) = \rho_{k,j}(k) + \eta\left(-\frac{\partial E_d}{\partial \rho_{k,j}}\right) + \beta\left(\rho_{k,j}(k) - \rho_{k,j}(k-1)\right) \quad (4.70)$$

$$\sigma_{k,j}(k+1) = \sigma_{k,j}(k) + \eta\left(-\frac{\partial E_d}{\partial \sigma_{k,j}}\right) + \beta\left(\sigma_{k,j}(k) - \sigma_{k,j}(k-1)\right) \quad (4.71)$$

$$\alpha(k+1) = \alpha(k) + \eta\left(-\frac{\partial E_d}{\partial \alpha}\right) + \beta(\alpha(k) - \alpha(k-1)) \quad (4.72)$$

The proposed procedure for online learning is performed at every sample time.

4.3.1.2 Adaptive LAMDA Control

As background, it has been used the AIC strategy [86]. This method requires an offline learning by using random values as training output, but also, the plant response to these values as training input, as shown in Figure 4.8a. Here, it is proposed to use the Adaptive LAMDA as an identifier, applying a random input $u(k)$ to the plant and taking the output $x(k+1)$, its previous values $[x\,(k);\,\dots;\,x(k-q)]$, and the delayed values $[u(k-1);\dots;\,u(k-p)]$ as descriptors. The delayed network uses as inputs in the application stage: the desired plant output and the current plant. With training,

the internal parameters of the LAMDA model are updated to minimize the error $e_u(k)$ through the process detailed in subsection 4.3.1.1. The application stage is implemented with the trained LAMDA model, as shown in Figure 4.8b.

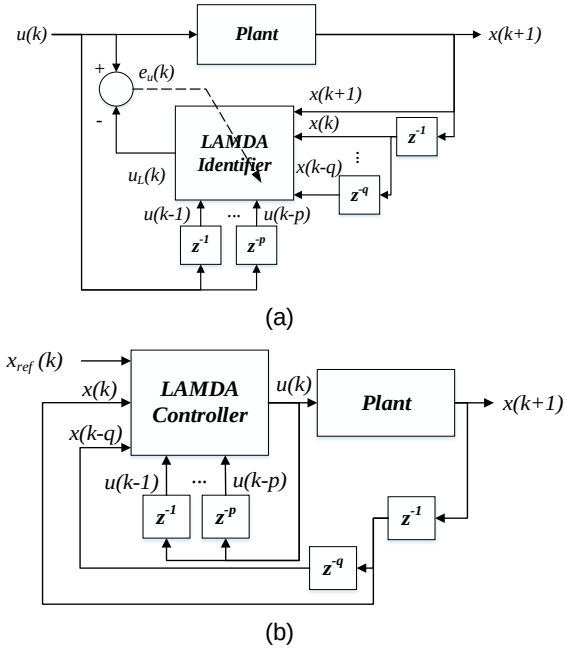


Figure 4.8. (a) Block diagram of the training phase of the inverse control method, (b) Block diagram of the application phase of the inverse control method

This model takes as inputs the desired reference $x_{ref}(k+1)$, the states of the plant $[x(k);\ldots;x(k-q)]$, and the delayed values $[u(k-1),\ldots,u\,(k-p)]$. The main idea of this method is to estimate the inverse plant model based on past and current plant outputs and inputs, to obtain the feedback control. The selection of p and q depends on an

estimation of the order of the plant. That is, the expert must define how many states passed for the input and for the output must be considered, for example, for a first-order system would be required: $p = 1, q = 1$, for a second-order system: $p = 2, q = 2$. This considering the number of poles and zeros that the system could have.

4.3.1.3 Feedback Control with Adaptive LAMDA

The feedback control scheme presented in Figure 4.9 is proposed to control systems with online learning.

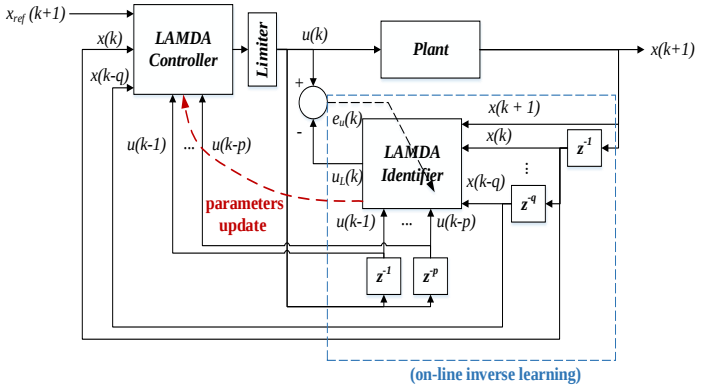


Figure 4.9. Block diagram of the online inverse learning control with Adaptive LAMDA

Once the model has been trained, as shown in Figure 4.6a, initial parameters are set in the identifier. In the application, the identifier is trained online in a supervised manner with the hybrid learning of Figure 4.5. A duplicate LAMDA is used as controller, considering now the desired reference

$x_{ref}(k+1)$, updating its internal parameters in each sample time based on the learning performed by the identifier.

Due to the online learning, the proposed scheme is able to bring the system output to the reference even in the presence of disturbances, or when the dynamic of the plant is variable.

4.3.1.4 Convergence of the learning algorithm

For online learning, it is used the hybrid algorithm presented in subsection 4.3.1.1. In each iteration of the closed-loop of Figure 4.9, the antecedents and consequents are adjusted with the aim that the LAMDA output converges to a desired value.

As a fundamental contribution of our work we have shown that the stability of the system is guaranteed by complying with the restrictions analyzed in detail and proven in Theorem 2 presented in Appendix E which is part of our paper [136], in which the Adaptive-LAMDA method is discussed in greater depth. The conditions are:

$$0 < \eta < \frac{2}{3(N_\Upsilon + N_\Psi + N_\alpha)} \tag{4.73}$$

For $e_r(k) > 0$:

$$0 < \beta < \frac{2e(k)}{3\left(\left(\frac{\partial Out_l^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right)} \tag{4.74}$$

$$0 < tr\big(P(k+1)\big) < \frac{2e(k)}{3e_r(k)(\|a(k+1)\|_F)^2} \tag{4.75}$$

For $e_r(k) < 0$:

$$0 < \beta < \frac{-2e(k)}{3\left(\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right)} \quad (4.76)$$

$$0 < tr\big(P(k+1)\big) < \frac{-2e(k)}{3e_r(k)(\|a(k+1)\|_F)^2} \quad ; \quad e_r < 0 \quad (4.77)$$

Where:

$$e_r(k) = Out(k+1) - a^T(k+1)h(k) \quad (4.78)$$

$$N_\Upsilon = \left(\|\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\|_2\right)^2, N_\Psi = \left(\|\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\|_2\right)^2,$$

$$N_\alpha = \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)^2 \quad (4.79)$$

And the terms of the antecedent in vector form corresponding to the centers are grouped in $\Upsilon(k)$, the standard deviation of the classes are grouped in $\Psi(k)$ and the consequent parameters are grouped in the matrix $h(k)$ as follows:

$$\Upsilon(k) = [\rho_{1,1}(k), \dots, \rho_{1,j}(k), \dots, \rho_{2,1}(k), \dots, \rho_{2,j}(k), \dots, \rho_{k,1}(k), \dots, \rho_{k,j}(k), \dots, \rho_{m,n}(k)]^T \quad (4.80)$$

$$\Psi(k) = [\sigma_{1,1}(k), \dots, \sigma_{1,j}(k), \dots, \sigma_{2,1}(k), \dots, \sigma_{2,j}(k), \dots, \sigma_{k,1}(k), \dots, \sigma_{k,j}(k), \dots, \sigma_{m,n}(k)]^T \quad (4.81)$$

$$h(k) = \begin{bmatrix} h_{11} & \dots & h_{1j} & \dots & h_{1n} & h_1 \\ h_{k1} & \dots & h_{kj} & \dots & h_{kn} & h_k \\ h_{m1} & \dots & h_{mj} & \dots & h_{mn} & h_m \end{bmatrix} \quad (4.82)$$

Equations (4.73)-(4.77) guarantee the convergence of the error $e(k) \longrightarrow 0$ in the training stage for a controllable system independent of the application, system order, number of inputs and classes.

Some considerations must be taken into account to determine the stability of the entire system:

- As mentioned in [9], analyzing controller stability based on online learning is a complex task that is still an open field in adaptive inverse learning schemes that will be addressed in a future work. However, the following aspects can be considered for local stability:
- Bounded Input–Bounded Output (BIBO) Stability: BIBO stability is guaranteed. The normalization of the $GADs$, through the computation of the normalized $NGAD_{k,\bar{0}} \leq 1$ and the introduced limiter shown in Figure 4.9, ensure that the adaptive LAMDA model is bounded for all inputs.
- It is assumed that the learning algorithm in the LAMDA Identifier has converged because a constant change in the parameters would make it hard to analyze stability. Under this assumption, it is only necessary to take the LAMDA controller into account. If the error of the learning converges $e(k) \longrightarrow 0$, then the LAMDA model is an identical copy of the real process, so it is guaranteed that there is a solution to the inverse model, allowing to calculate a control action $u(k)$, which satisfies $x(k+1) \cong x_{ref}(x+1)$.

4.4 LSMC based on Z-numbers (ZLSMC)

In subsections 4.2 and 4.3, two new controllers have been formalized: LSMC and Adaptive LAMDA, respectively.

Based on experiments where the controllers have been tested in different nonlinear systems, only one of these proposals has been selected to apply the theory of Z-numbers as it has been proposed in the scope of this work. Considering the controllers described in subsections 4.2 and 4.3, the decision has been made to apply the Z-numbers theory to the LSMC controller based on the following considerations:

- The computational complexity detailed in subsection 5.3 shows that LSMC is less computationally expensive than Adaptive LAMDA.
- The characteristics of SMC applied to LAMDA make this controller a robust and chattering-free proposal.
- The Lyapunov theory guarantees a stable controller in LSMC and stable learning in Adaptive LAMDA; however, LSMC does not require a learning phase.

The Z-number theory is recent and is expanding into the field of control systems due to the great potential it can offer. In this work, it has been considered that the management of the reliability related to the error (deviation between the reference and the current output of the process) can help to reach the reference in a faster way to large errors, and it is less aggressive for small errors. The proposal is called LSMC based on Z-Numbers (ZLSMC).

In some applications, the Z-numbers can be represented with the restriction and reliability of two singleton functions. In order to compute the TU of singleton functions of Z-numbers we propose the Lemma 1.

Lemma 1. The TU of a fuzzy number with two singleton functions is equal to the product of the functions $TU(Z) \approx \rho_1\rho_2$.

Proof. According to (2.23), the TU is calculated based on the centers and variances of the two fuzzy numbers. From (2.12), if $\sigma_1 \approx 0$ and $\sigma_2 \approx 0$, then the Gaussian functions behave similarly to singleton functions (impulsive response in the centers ρ_1 and ρ_2), therefore:

$$TU(Z) = TU(Az, Rz) = \frac{\rho_1\rho_2}{(1+8{\sigma_1}^2)(1+8{\sigma_2}^2)}$$

$$\approx \frac{\rho_1\rho_2}{(1+8\times 0^2)(1+8\times 0^2)} = \rho_1\rho_2 \qquad (4.83)$$

Being the proof demonstrated completely. The final result obtained in this operation (4.83) coincides with the presented in [137]. Lemma 1 is useful for the formalization of the ZLSMC controller, specifically in the calculation of the new centers of the classes and for the calculation of the controller output based on the GAD and reliability as detailed below.

4.4.1 Formalization of ZLSMC

As a starting point, it is considered to use the Gaussian functions to compute the MAD as presented in (2.5). This expression is similar to the one presented in (2.12), therefore the restriction in the case of LAMDA corresponds to MAD, thus $\mu_{Az} = MAD_{k,j}$. Also, it is necessary to measure the reliability parameter for μ_R, procedure that will be detailed later in the document for continuous and discontinuous control actions. With the two parameters of a Z-number, then

it is feasible to compute the TU of each class of the control system. LAMDA identifies the current state of the system and takes it to the desired state. For this purpose, it is necessary to define rules based on the system knowledge, which is also carried out in conventional fuzzy controllers. The analytic expression that generalizes (2.11) for the k-rule of the Z-fuzzy logic inference system considering the classes in LAMDA is represented as follows:

$$Rule^{(k)}: IF\ o_1\ is\ Z_1^p\ and \dots\ o_j\ is\ Z_j^q \dots and\ \ o_l\ is\ Z_l^r$$
$$THEN\ y_k\ is\ Z^k \quad (4.84)$$

where $Rule^{(k)}$ is the rule applied for the class k, o_j is the descriptor j of the object O that takes values from the universe of discourse U_j. The output linguistic variable y_k is defined on a universe of discourse V. $Z_j^q = (A_j^q, R_j^q)$ denotes de Z-number for the descriptor j and the fuzzy set q, and $Z^k = (\gamma_k, R_k)$ is the consequent Z-number.

In the LAMDA context it is proposed to use the Z-number concepts to improve the controller response, thus rewriting (4.84) for two inputs (descriptors) it is obtained:

$$Rule^{(k)}: IF\ o_1\ is\ \left(MAD_{k,1}, \mu_{Rz1}\right)\ and\ o_2\ is\ \left(MAD_{k,2}, \mu_{Rz2}\right)$$
$$THEN\ y_k\ is\ (\gamma_k, R_k) \quad (4.85)$$

The TU is used to compute the new centers of the MADs and it is used at the output to recalculate the weights applied to the GADs, making them adaptable as a function of the sliding surface s and its first derivative $\dot{s}$ which are the descriptors of the proposed approach as detailed as follows:

The same procedure for the design of LSMC is used for the design of ZLSMC (see from (4.5) to (4.21)).

- Continuous control action:

Based on (4.21), it is necessary to compute u_c to obtain $\dot{s}(t) = 0$, for which it is required to know only the sign of the function $b(X(t), t)$ associated with u_c and to set the rules for the classes of LAMDA. In (4.21), it can be seen that if $b(X(t), t) > 0$, then $\dot{s}(t)$ decreases as u_c increases, and vice-versa. With this information, a set of classes and rules to obtain $\dot{s}(t) = 0$ is established.

Five classes $C \in [-1,1]$ for $\dot{s}(t)$ are used, this, supported by the sensitivity analysis performed in subsection 5.2.1.2.1 and detailed in [138] where has been shown that selecting three or seven classes does not significantly affect the performance of controllers based on LAMDA and the only required consideration is an adequate calibration of the scaling gains. The fuzzy sets for the classes of the variable $\dot{s}(t)$ are Negative Big ($NB = -1$), Negative Small ($NS = -0.5$), zero ($ZE = 0$), Positive Small ($PS = 0.5$), and Positive Big ($PB = 1$). These classes are used to define the rules to compute the normalized continuous control action u_{nc}. For proper calibration, the scaling gain k_1 is used for the input $\dot{s}(t)$, and the scaling gain k_c at the continuous control output as:

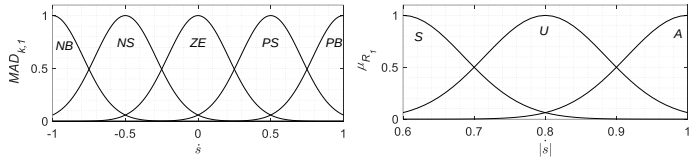
$$u_c = k_c u_{nc} \implies u_c = k_c ZLSMC(\dot{s}); \quad k_c > 0 \qquad (4.86)$$

The rule table corresponding to u_c where the measure of $\dot{s}(t)$ is required as shown (4.21) is presented in Table 4.4 considering $C_k = (\gamma_k, R_c)$; $\forall k = \{1,2,\dots,5\}$.

Table 4.4. Rule table of ZLSMC for $\dot{s}(t)$

	$\dot{s}(t)$				
	NB,S	***NS,U***	***Z,A***	***PS,U***	***PB,S***
$b(X,t) > 0$	$C_1 = NB, R_c$	$C_2 = NS, R_c$	$C_3 = ZE, R_c$	$C_4 = PS, R_c$	$C_5 = PB, R_c$
$b(X,t) < 0$	$C_1 = PB, R_c$	$C_2 = PS, R_c$	$C_3 = ZE, R_c$	$C_4 = NS, R_c$	$C_5 = NB, R_c$

For the computation of the u_c, the classes per descriptor and the membership functions of the reliability are presented in Figure 4.10. Note that for the reliability part the absolute value of $\dot{s}$ is used. In [91] is proposed the establishment of three classes to represent reliability in a simple and complete way for control systems, these are: "S" is "sometimes," "U" is "usually," and "A" is "always".



(a) (b)

Figure 4.10. Membership functions for: a) MADs of $\dot{s}(t)$, b) reliability of $|\dot{s}(t)|$

To define the rules in Table 4.4, when $b(X(t),t) > 0$ and based on (4.21), it is noted that $\dot{s}(t)$ decreases as u_c increases, and vice-versa. This information is sufficient for the rules definition to satisfy $\dot{s}(t) = 0$. For instance, if $\dot{s}(t)$ is PB, then large positive control action u_c is needed in order to decrease quickly $\dot{s}(t)$, and the reliability $|\dot{s}(t)|$ is assigned with a value sometimes "S" whose center is in 0.6. This causes the center calculated with the Total Utility to move to the left, which in fuzzy control means a more abrupt control

action that causes a rapid decrease of $\dot{s}(t)$. If $\dot{s}(t)$ is NB, then the error and its derivatives are close to zero, then a small positive control action u_c is needed in order to decrease slowly $\dot{s}(t)$, also, in order to obtain a smoother control action, the reliability is assigned a value usually "U" whose center is in 0.8. Finally, if $\dot{s}(t) = ZE$ (desired condition), then no control action is required, thus $u_c = ZE$ and the reliability is assigned a value always "A" whose center is in 1. Therefore as $\dot{s}(t)$ is close to zero, the control action is smoother. As shown in Table 4.4, by using the value$|\dot{s}(t)|$, the same previous analysis is valid for the NB and NS classes.

On the other hand, when $b(X(t), t) < 0$ and based on (4.21), it is noted that $\dot{s}(t)$ decreases as u_c decreases, and $\dot{s}(t)$ increases as u_c increases. Therefore, as observed in Table 4.4, only the sign of the restriction changes and the criterion to define reliability is the same for the case $b(X(t), t) > 0$.

In the ZLSMC, two descriptors at the input of LAMDA are used, therefore (2.5) is computed for $s(t)$ and its derivative, that is $j = 1$ for $\dot{s}(t)$ and $j = 2$ for $s(t)$. The calculation of the new class centers is made based on the $TU_{k,j}(Z)$ applied to the descriptor $\dot{s}(t)$, replacing the LAMDA restriction and reliability in (4.84):

$$TU_{k,1}(Z) = TU\left(MAD_{k,1}, R_1\right) = \frac{\rho_{k,1} c_{R_1}}{\left(1 + 8{\sigma_{k,1}}^2\right)\left(1 + 8{\sigma_{R_1}}^2\right)} \quad (4.87)$$

where $R_1 = Gauss(c_{R_1}, \sigma_{R_1})$ is the reliability of the $MAD_{k,1}$. Unlike works that address the control with Z-numbers [91,102,103], the reliability at the output is not defined, in this work is proposed to compute its weight value as [137]:

$$R_c = \frac{\int_0^1 |\dot{s}| \mu_{R1}(|\dot{s}|)}{\int_0^1 \mu_{R1}(|\dot{s}|)} \quad (4.88)$$

From (4.84), it is used the R_c and the singleton values γ_k. The Total Utility of the Z-number at the output can be denoted as:

$$\gamma_{R_c}^k = \gamma_k \times R_c \quad (4.89)$$

Then, the output of the control action u_{nc}, based on (4.4) and (4.134), is computed as:

$$u_{nc} = \left| \frac{arg\ max(\gamma_k)}{\sum_{k=1}^{m} \gamma_k GAD_{k,max(\bar{o})}} \right| \sum_{k=1}^{m} \gamma_{R_c}^k GAD_{k,\bar{o}} \quad (4.90)$$

- Discontinuous control action

Based on (4.26), it is necessary to compute u_d to satisfy $s(t)\dot{s}(t) < 0$. As in the case of continuous control action, five classes are set for each input $\dot{s}(t)$ and $s(t)$ based in the scalability analysis presented in [138] and three classes for the reliability as presented in [91]. Due to the normalization of the classes is computed u_{nd}, therefore, the scaling gain k_2 is added at the input $s(t)$, and the scaling gain k_d at the discontinuous control output as:

$$u_d = k_d u_{nd} \implies u_d = k_d ZLSMC(s, \dot{s}) \quad ; \quad k_d > 0 \quad (4.91)$$

For the computation of the discontinuous control action based on Z-numbers, it is addressed the case $b(X(t), t) > 0$ since in the opposite case ($b(X(t), t) < 0$), only the sign of the classes changes in the restriction part as detailed in the definition of rules of Table 4.4. The centers of the Z-classes are presented in Table 4.5 considering $C_k = (\gamma_k, R_d)$; $\forall k = \{1,2, \dots,25\}$.

Table 4.5. Rule table of ZLSMC for $s(t)$ and $\dot{s}(t)$, $b(X(t),t) > 0$

$s(t)$	$\dot{s}(t)$				
	NB,S	***NS,U***	***ZE,A***	***PS,U***	***PB,S***
PB,S	$C_5 = ZE, R_d$	$C_{10} = ZE, R_d$	$C_{15} = PS, R_d$	$C_{20} = PB, R_d$	$C_{25} = PB, R_d$
PS,U	$C_4 = ZE, R_d$	$C_9 = ZE, R_d$	$C_{14} = PS, R_d$	$C_{19} = PB, R_d$	$C_{24} = PB, R_d$
ZE,A	$C_3 = NB, R_d$	$C_8 = NS, R_d$	$C_{13} = ZE, R_d$	$C_{18} = PS, R_d$	$C_{23} = PB, R_d$
NS,U	$C_2 = NB, R_d$	$C_7 = NB, R_d$	$C_{12} = NS, R_d$	$C_{17} = ZE, R_d$	$C_{22} = ZE, R_d$
NB,S	$C_1 = NB, R_d$	$C_6 = NB, R_d$	$C_{11} = NS, R_d$	$C_{16} = ZE, R_d$	$C_{21} = ZE, R_d$

For the computation of u_d, the classes per descriptor and the membership functions of the reliability are presented in Figure 4.11. Note that for the reliability part, the absolute values of $\dot{s}$ and s are measured, since it is considered to give more weight when$|\dot{s}|$ and $|s|$ are far from zero.

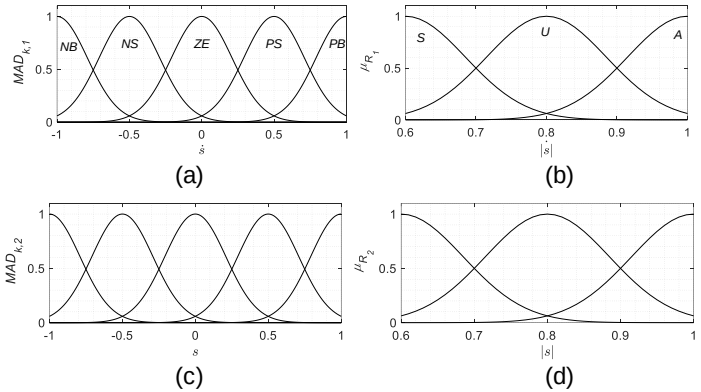


Figure 4.11. a) Membership functions for: a) MADs of $\dot{s}(t)$, b) reliability of $|\dot{s}(t)|$, c) MADs of $s(t)$, d) reliability of $|s(t)|$

In the Z-LSMC, for u_d are used the two descriptors at the input of LAMDA, therefore (2.5) is computed for s and its derivative, that is $j = 1$ for $\dot{s}(t)$ and $j = 2$ for $s(t)$. Then, the calculation of the new class centers is made based on the $TU_{k,j}(Z)$ as:

$$TU_{k,1}(Z) = TU\left(MAD_{k,1}, R_1\right) = \frac{\rho_{k,1} c_{R_1}}{\left(1 + 8{\sigma_{k,1}}^2\right)\left(1 + 8{\sigma_{R_1}}^2\right)} \quad (4.92)$$

$$TU_{k,2}(Z) = TU\left(MAD_{k,2}, R_2\right) = \frac{\rho_{k,2} c_{R_2}}{\left(1 + 8{\sigma_{k,2}}^2\right)\left(1 + 8{\sigma_{R_2}}^2\right)} \quad (4.93)$$

The reliability at the output is computed as the weight value as proposed in [137]:

$$R_d = arg\ max\left(\frac{\int_0^1 |\dot{s}| \mu_{R1(|\dot{s}|)}}{\int_0^1 \mu_{R1(|\dot{s}|)}}, \frac{\int_0^1 |s| \mu_{R2(|s|)}}{\int_0^1 \mu_{R2(|s|)}}\right) \quad (4.94)$$

It is proposed to choose the maximum value of the two reliabilities to obtain a more aggressive control action when the surface or its derivative are far from zero (the error is big) to take the system faster towards the reference.

To define the rules of Table 4.5, the following analysis has been considered:

- If $s(t) > 0$ and u_d increases, then the product $s(t)\dot{s}(t)$ decreases and vice-versa.
- If $s(t) < 0$ and u_d increases, then the product $s(t)\dot{s}(t)$ increases, and if u_d decreases, then $s(t)\dot{s}(t)$ decreases.

From this analysis, it is proposed to generate a control action u_d to satisfy $s(t)\dot{s}(t) < 0$. As has been described for the continuous control action, the case of the discontinuous control action is similar, that is, S is associated as reliability to classes PB and NB, U to classes PS and NS and A to class

ZE in order to generate abrupt control actions when the surface and its derivative are far from the desired value and smooth control actions when they are close to zero.

From (4.84), it is used the weight of the reliability R_d and the values γ_k. The TU at the output can be denoted as:

$$\gamma_{R_d}^k = \gamma_k \times R_d \tag{4.95}$$

Then, the output of the control action u_{nd}, based on (4.4), and (4.95), is calculated as:

$$u_{nd} = \left| \frac{arg\ max(\gamma_k)}{\sum_{k=1}^{m} \gamma_k GAD_{k,max(\bar{O})}} \right| \sum_{k=1}^{m} \gamma_{R_d}^k GAD_{k,\bar{O}} \tag{4.96}$$

This approach removes the chattering of the SMC replacing the discontinuous sign function with the rules and classes of LAMDA. The overall control action is computed as:

$$u = k_c ZLSMC(\dot{s}) + k_d ZLSMC(s, \dot{s}) \tag{4.97}$$

The ZLSMC scheme is shown in Figure 4.12, detailing the blocks of the controller applied in the continuous ($ZLSMC_c(\dot{s})$) and discontinuous ($ZLSMC_d(s,\dot{s})$) parts, the descriptors used in each LAMDA block, and the scaling gains in the inputs and the outputs.

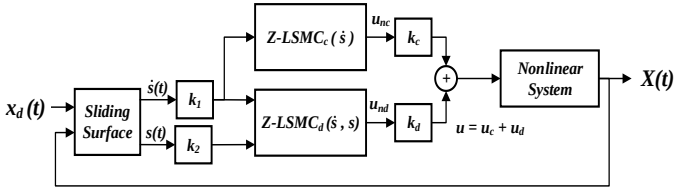


Figure 4.12. Block diagram of the ZLMC

Finally, the stability analysis is the same as LSMC shown in the subsection 4.2.3.

5. EXPERIMENTATION AND RESULTS

5.1 LAMDA in classification and clustering tests

5.1.1 LAMDA-HAD in classification process

To analyze the performance of the LAMDA-HAD algorithm and validate the proposal, two types of tests are performed:

- A comparative analysis between LAMDA-HAD and other well-known classification methods, applied to several balanced and unbalanced classification benchmarks with different characteristics.
- A comparative analysis between LAMDA, LAMDA-HAD and the classifiers with the best performance, to test the skills of the methods for the identification of new classes.

Additionally, in Appendix B and [105], a detailed comparative analysis between LAMDA and LAMDA-HAD is presented, applied to the aforementioned benchmarks where the improvements presented by each of the proposed extensions are exhaustively analyzed.

5.1.1.1 LAMDA-HAD validation

For LAMDA-HAD validation, MAD uses the Fuzzy binomial function. GAD uses a Hammacher operator, and the exigency is set to $\alpha = 0.9$. The algorithm is tested in some classification problems from the UCI Machine Learning Repository [109], and benchmark datasets from [139,140], with different number of data, attributes, and classes C_i. Table 5.1 reports the characteristics of the datasets, and in

Appendix A, two-dimensional graphs of the datasets are shown using the t-Distributed Stochastic Neighbor Embedding, method used for High-Dimensional Data reduction (tsne function in Matlab).

Table 5.1. Datasets used to validate LAMDA-HAD

Dataset	**Size**	**Features**	**Characteristics number of instances in each class**
Iris	150	4	3 classes corresponding to types of Iris plants (setosa, versicolor, virginica). $C_1 = 50;\ C_2 = 50;\ C_3 = 50.$
Brest Cancer	699	9	2 classes corresponding to types of cancer (benign, malignant). $C_1 = 458;\ C_2 = 241.$
R15	600	2	15 classes corresponding to types to 2D Gaussian groups that are positioned in rings. $C_1 = C_2 = \cdots = C_{15} = 40.$
Wine type	178	13	3 classes corresponding to three types of wines. $C_1 = 59;\ C_2 = 71;\ C_3 = 58.$
Glass	214	10	7 classes corresponding to types of lens. $C_1 = 70;\ C_2 = 76;\ C_3 = 17;$ $C_4 = 0; C_5 = 13; C_6 = 9;$ $C_7 = 29.$
Seeds	210	7	3 classes corresponding to types of wheat (kama, rosa and Canadian). $C_1 = 70;\ C_2 = 70;\ C_3 = 70.$
Wholesale costumers	440	3	3 classes corresponding to types of clients of a wholesale distributor. $C_1 = 77;\ C_2 = 47;\ C_3 = 316.$

Wine quality	1599	11	6 classes corresponding to types of red variant of the Portuguese "Vinho Verde". $C_1 = 10;\ C_2 = 53;\ C_3 = 681; C_4 = 638;\ C_5 = 199; C_6 = 18.$
s2	5000	2	15 Gaussian classes of synthetic data with 22% of overlapping between them. $C_1 \approx C_2 \approx \cdots \approx C_{15} \approx 330.$
Wireless Indoor Localization	2000	7	7 classes corresponding to the indoor location based on wifi signal strengths observed on smartphone. $C_1 = 500;\ C_2 = 500;\ C_3 = 500;\ C_4 = 500$.

To test the classifiers, the k-fold cross validation with $k = 10$ is implemented, where 90% random samples of the datasets are used in the training process, while the remaining 10% of the samples are used for the test. $k = 10$ is a value that has been found through experimentation, which gives results with low bias and modest standard deviation. The advantage of k-fold cross validation is that all the samples in the dataset are eventually used for both, training and testing. Other validation methods like those presented in [141] allow making a more efficient cross-validation of classifiers. However, for the comparison of the presented algorithms in this work, the standard cross validation technique is enough because it does not introduce any perturbation to the results.

The algorithms used for the comparison are: LAMDA, LAMDA-HAD (LM-HAD), LDA (Linear Discriminant Analysis [142]), NN (Feedforward Neural Networks [143]), SVM

(Support Vector Machines [144]), NBC (Naive Bayes Classifier [145]), DT (Decision Trees [146]), and RF (Random Forest [147]). The consigned results of the aforementioned methods correspond to own experiments, using the Matlab software toolboxes, applying cross validation to all of them. Most algorithms do not require an exhaustive parameterization, but in the cases in which it has been required, a heuristic calibration has been carried out to obtain the best possible results, to perform a fair comparison between the methods.

The results of the mean value of the cross validation $(\bar{v})$ of F-measure and Accuracy are used to evaluate the performance of the algorithm (see Tables 5.2 and 5.3) respectively, where the algorithm that got the highest classification metric is standing out in bold text, the second best one is standing out in blue, while the third best one is standing out in red. Also the standard deviation (σ) is presented to evaluate the dispersion of the data. To more clearly observe the performance of the algorithms, in Appendix B, the analysis of the classifiers is presented based on the ROC (Receiver Operating characteristic) and AUC (Area Under the Curve) curves, parameters used in the diagnostic field. of classifiers.

Table 5.2 shows that, based on the average values LAMDA-HAD is the third best behind RF and LDA. The results of F-measure in most cases are very close to the values obtained with the best algorithm, which allows to conclude that the proposal presents a very good performance considering that this metric is a combination of Precision and Recall.

Table 5.2. Average F-measure (%) of LAMDA-HAD and other classification algorithms

		LDA	NN	SVM	NBC	DT	RF	LMD	LM-HAD
Iris	$\overline{v}$	**0.98**	0.95	**0.97**	**0.96**	0.95	0.94	0.78	**0.97**
	σ	0.010	0.001	0.005	0.003	0.016	0.008	0.019	0.005
Breast	$\overline{v}$	0.95	**0.96**	0.35	**0.96**	0.93	**0.97**	0.42	0.94
	σ	0.011	0.009	0.001	0.013	0.010	0.019	0.033	0.008
R15	$\overline{v}$	**0.99**	**0.77**	**0.99**	**0.99**	**0.99**	**0.99**	**0.93**	**0.99**
	σ	0.001	0.010	0.001	0.001	0.001	0.001	0.005	0.001
Wine-Type	$\overline{v}$	**0.98**	0.93	0.42	**0.97**	0.92	**0.98**	0.82	0.95
	σ	0.006	0.012	0.055	0.008	0.017	0.007	0.024	0.009
Glass	$\overline{v}$	0.51	0.36	**0.52**	0.50	**0.64**	**0.71**	0.37	0.51
	σ	0.052	0.041	0.026	0.030	0.066	0.012	0.041	0.052
Seeds	$\overline{v}$	**0.96**	0.87	0.90	0.91	0.92	**0.93**	0.52	**0.92**
	σ	0.008	0.004	0.008	0.006	0.007	0.004	0.007	0.009
Whol.	$\overline{v}$	0.30	0.29	0.28	**0.33**	**0.34**	0.29	0.21	**0.34**
	σ	0.017	0.019	0.019	0.025	0.058	0.042	0.047	0.025
Wine	$\overline{v}$	**0.32**	0.30	0.12	0.26	**0.32**	**0.36**	0.23	0.30
	σ	0.052	0.012	0.036	0.042	0.029	0.010	0.031	0.07
s2	$\overline{v}$	**0.96**	0.44	0.01	**0.97**	0.96	0.97	0.92	**0.97**
	σ	0.005	0.001	0.006	0.002	0.006	0.003	0.007	0.004
Wire-less	$\overline{v}$	0.95	0.94	0.78	**0.96**	0.95	**0.95**	0.86	**0.95**
	σ	0.006	0.005	0.002	0.003	0.004	0.001	0.004	0.002

*The detailed values are presented in [105].

Accuracy is another important metric to evaluate the performance of the algorithm. These results are shown in Table 5.3. The accuracy is computed to know in which cases the algorithms are able to detect imbalance situations.

Table 5.3. Average accuracy (%) of LAMDA-HAD and other classification algorithms

		LDA	NN	SVM	NBC	DT	RF	LMD	LM-HAD
Iris	$\bar{v}$	**0,99**	0,98	0,98	**0,97**	0,96	0,96	0,81	0,98
	σ	0.001	0.005	0.001	0.003	0.002	0.005	0.01	0.001
Breast	$\bar{v}$	0,96	0,96	0,41	**0,96**	0,94	**0,97**	0,51	0,94
	σ	0.007	0.005	0.007	0.006	0.004	0.005	0.035	0.006
R15	$\bar{v}$	1,00	0,97	**1,00**	1,00	1,00	**1,00**	0,95	1,00
	σ	0.000	0.002	0.000	0.000	0.000	0.000	0.005	0.000
Wine-Type	$\bar{v}$	**0,99**	0,97	0,62	**0,98**	0,93	0,98	0,84	0,95
	σ	0.001	0.003	0.025	0.004	0.06	0.002	0.012	0.004
Glass	$\bar{v}$	0,88	0,82	**0,89**	0,82	0,90	**0,93**	0,72	0,82
	σ	0.042	0.041	0.002	0.004	0.007	0.007	0.032	0.011
Seeds	$\bar{v}$	**0,98**	0,90	0,94	0,93	**0,95**	0,96	0,73	0,93
	σ	0.001	0.005	0.006	0.004	0.007	0.005	0.017	0.007
Whol.	$\bar{v}$	0,81	0,65	**0,81**	0,65	0,70	**0,79**	0,43	0,49
	σ	0.021	0.038	0.021	0.074	0.065	0.033	0.041	0.023
Wine	$\bar{v}$	**0,87**	0,86	0,73	0,73	0,87	**0,90**	0,66	0,77
	σ	0.036	0.039	0.028	0.069	0.058	0.031	0.051	0.043
s2	$\bar{v}$	**0,97**	0,52	0,07	**0,97**	0,96	0,97	0,96	0,97
	σ	0.017	0.019	0.021	0.012	0.018	0.002	0.002	0.015
Wire-less	$\bar{v}$	**0,95**	0,95	0,77	0,95	0,95	**0,98**	0,89	0,96
	σ	0.015	0.011	0.032	0.012	0.012	0.001	0.022	0.021

*The detailed values are presented in [105].

Based on the results of the Table 5.3 and taking into account that this metric is valid for the cases in which the classes are balanced, it is observed that the algorithm is the third best behind LDA, in first place, and RF and SVM, in second place. This analysis has been based on the number of times that the algorithms rank first, second and third in relation to the average value. The results where LAMDA-HAD decreases its performance are the datasets: Glass, Wholesale Costumers and Wine Quality. The proposed algorithm allows identifying the imbalance in the classes,

what other approaches cannot because they have a high value for the accuracy. As discussed above, the accuracy is not recommended when the classes are unbalanced. Therefore, in these cases, F-measure shows adequately the performance of the classifier. F-measure is used for unbalanced datasets and accuracy for balanced datasets. These results are presented in Table 5.4, showing the relative error between the result of the best classifier based on the average value and LAMDA-HAD.

Table 5.4. Relative error between the highest value metrics and LAMDA-HAD metrics

Dataset	Metric	Best Value	LAMDA-HAD	Relative Error (%)
Iris	Acc.	0.9867	0.9822	0.4561
R15	Acc.	0.9996	0.9993	0.0300
Wine Type	Acc.	0.9885	0.9545	3.4396
Seeds	Acc.	0.9778	0.9311	4.7760
s2	Acc.	0.9726	0.9700	0.2673
Wireless	Acc.	0.9849	0.9555	2.9851
Breast	F-mea.	0.9692	0.9403	2.9818
Glass	F-mea.	0.7138	0.5139	28.005
Whol.	F-mea.	0.3372	0.3372	0.0000
Wine Quality	F-mea.	0.3644	0.2987	18.029

Table 5.4 allows to observe that the performance of LAMDA-HAD is comparable to the performance of the classifiers that perform better. Low relative errors show that the new algorithm has a high performance in tasks of supervised learning, and greatly improve the performance in relation to the original LAMDA in all cases (see Tables 5.2 and 5.3).

There are two specific benchmarks: Glass and Wine Quality, in which LAMDA-HAD has the largest difference, which are due to the distribution of data in descriptors that do not

provide relevant information for an adequate classification. In the other cases, the difference does not exceed 0.0467 in relation to the best classifier.

5.1.1.2 Testing algorithms for the identification of new classes

In this test, a Wireless dataset is used [109,148], which consists of data collected to perform experimentation on how wifi signal strengths can be used to determine locations. The dataset has 2000 data instances with 7 descriptors each one, corresponding to wifi signal strength observed on smartphones, used to identify 4 locations (classes). To perform the classification, the following procedure is considered: Data belonging to three locations (classes C_1, C_2 and C_4) were selected for the training stage, while the data of the other location (class C_3), combined with the other classes, was used for the validation stage (see Figure 5.1). The training stage was carried out with 90% of the database of the three classes, and the remaining 10% of the data and the location not taken into account during the training stage, were selected for the validation stage. This test is done to validate the ability of the algorithms for the identification of new classes.

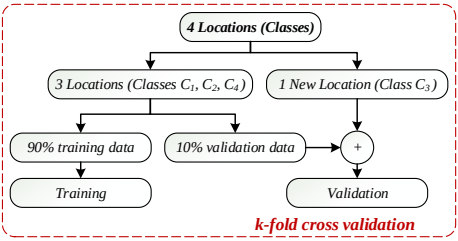


Figure 5.1. Database partitioning for new classes identification

During the training stage, the data of class 3 have not been considered as shown Figure 5.1. The obtained results in the validation phase, in which those individuals have been considered are shown in Figure 5.2. With this experiment, the behavior of LDA, RF, LAMDA and LAMDA-HAD is observed when they are tested with data that did not belong to neither class in the training stage. Additionally, Figure 5.2 shows the process of assignment of individuals. It shows the different results for each algorithm:

- the individuals who have not been marked are the correctly assigned;
- the misassigned data are marked in green;
- the individuals sent incorrectly to the NIC are marked in red;
- and the data marked in black are considered part of a new class automatically identified by the algorithms.

In the case of LDA, the elements of class 3 were mostly assigned to class 1 and 4; on the other hand, in the case of RF (which in the previous experiment presented a perfect classification), now the majority individuals of class 3 have been assigned to class 2, that is, the algorithm found greater similarity of that data with respect to that class. LAMDA has been able to identify a new class, however, there are errors in the assignment of individuals, especially in classes 2 and 4, which is not appropriate. Finally, LAMDA-HAD is the algorithm that performs the best identification, allowing to observe several interesting features, which are:

- The algorithm has correctly identified all the elements of classes 1, 2 and 4.
- It has identified the most individuals of a new class (assigning them to class 0 that represents the NIC).
- There is an incorrect assignment of certain individuals, but with a lower percentage with respect to the other algorithms. In particular, LAMDA-HAD is able to learn and identify new classes in the testing stage.

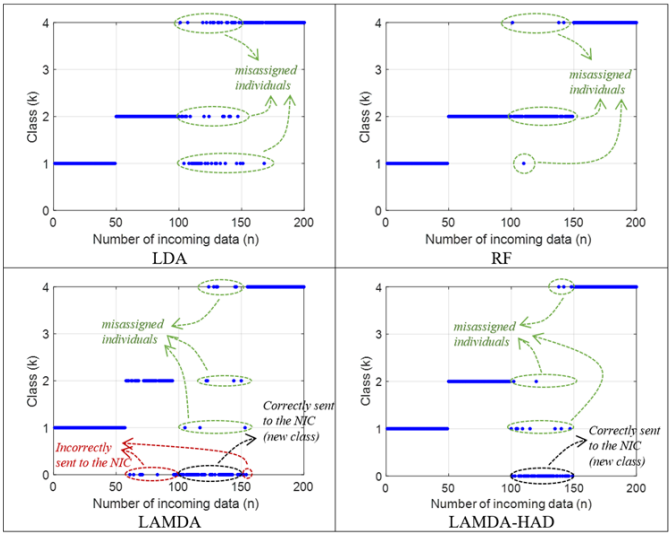


Figure 5.2. Classification results of new classes identification

Note that the comparison is not fair because LDA and RF cannot detect new classes, however the comparison is useful to identify one of the main advantages of the algorithm which is to identify classes not considered in the training

5.1.2 LAMDA-RD in clustering process

In this subsection, the experimental tests in different clustering tasks are presented. The goal of the experiments is to validate LAMDA-RD, analyzing the cluster quality and its performance. The following tests are carried out:

i. A general validation among LAMDA-RD, original LAMDA, and other well-known clustering methods tested in different benchmarks, making a comparative analysis of the quality of the results. In a first test, the statistics for clustering validation that handles the criterion that the labels of the clusters are unknown is computed. Then, the evaluation is performed considering the intrinsic and extrinsic characteristics of the obtained model. These metrics are: Silhouette Coefficient (SC), WB-index (WB), and Performance Coefficient (P_c) based on SC. Metrics as: Modification of the Silhouette coefficient $(SILA)$, Sum-of-squares within clusters (SSW), Sum-of-squares between clusters (SSB) are presented in [107]. In a second test, the datasets with labeled clusters to compare the formed partitions performed by the algorithms against the real classes is presented; this is a standard evaluation procedure for clustering used to compute the Rand Index (RI).

ii. A comparative analysis in a streaming data scenario among LAMDA-RD, LAMDA-TP and the algorithm called "Autonomous Data-driven Clustering for Live DataStream (ADDclustering) [149] is presented. This algorithm has been selected since it allows online clustering, a characteristic to be considered in order to make a fair comparison with LAMDA-RD. In this

experiment, the individuals are acquired from streaming data, to test the algorithms in online operation.

The tests described in i) and ii) are validated using datasets from [140,150,151]. The datasets have different characteristics as: number of individuals and features, level of intra-cluster overlap to observe how the allocation of individuals is made in those cases, balanced and unbalanced classes, and finally, the number of clusters (see detail in Table 5.5). Datasets with a large number of data are: Dim 1024, Unbalance and Postures (high-dimensional) and their analysis is required to observe the cluster quality and to measure the machine time to perform the partitions. 2-dimensional datasets are used for visualization purposes, to easily observe the behavior of the different algorithms. In all benchmarks, the original dimensionality has been maintained to make a fair comparison between the algorithms. Also, in Appendix A, a two-dimensional graphs of the datasets are shown using the t-Distributed Stochastic Neighbor Embedding.

Table 5.5. Datasets used to test the clustering algorithms [107]

Dataset	# Individuals	# Features	Overlapping	# Clusters
Dim 1024	1024	1024	0%	16
Segment	2310	19	unknown	7
Hepta	212	3	0%	7
R15	600	2	0%	15
Aggregation	788	2	0%	7
Unbalance	6500	2	0%	8
s1	5000	2	9%	15
s2	5000	2	20%	15
s3	5000	2	41%	15
a1	3000	2	22%	20
Postures	74975	15	Unknown	5

5.1.2.1 Comparison of LAMDA-RD with other clustering algorithms

In the following experiments, the parameters of the compared algorithms are tuned with the same care and separately for each dataset, to make a fair comparison. The tuning procedure of LAMDA-RD parameters is addressed in [107], where a sensitivity analysis of the results is detailed.

This test is done to compare the quality of the formed clusters with respect to the results of original LAMDA and other methods which are generally iterative, do not work online, and require the number of clusters as input parameter which has been considered in this work. These algorithms will serve to make an in-depth comparison in terms of performance. The results of some conventional algorithms such as: (K-means (KM), K-medoids (KMD), Fuzzy c-means (FCM), DBSCAN (DBS)), and Agglomerative hierarchical tree (AHT) [4] are presented in this work. We have presented a more detailed analysis in [107], showing the results of other algorithms as Spectral clustering (SPC) [53], Hierarchical density-based clustering (HDBSCAN “HDB”) [54] and Link-based cluster ensemble framework with consensus function (CON) [55].

Figure 5.3 shows the methodology used for this experiment. It should be noted that LAMDA works with streaming data while the other algorithms require the complete dataset. This test allows to evaluate the quality of the created clusters considering benchmarks with historical data.

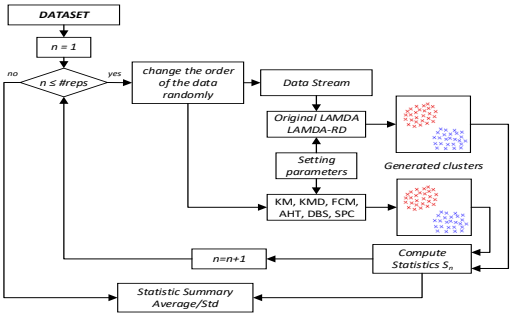


Figure 5.3. Methodology used for the comparison of LAMDA-RD with other approaches

As shown Figure 5.3, to obtain more reliable results, the experiment is repeated 20 times ($\#reps = 20$), each time performance metrics are computed, and from the obtained results, the mean value ($\bar{v}$) and standard deviation (σ) of the metrics are computed, to observe the repeatability and the confidence interval of the experiment.

Table 5.6. presents the SC for each clustering algorithm, where the best average (highest value) in each benchmark has been marked in bold text. The standard deviation shows the variability of the results in the different tests.

The best algorithms have SC values the closest to 1, identifying dense and well-separated clusters. In all benchmarks, LAMDA-RD is better than original LAMDA, in most cases significantly improving the quality of the created partitions, for instance, see the results in Segment, or in the cases of Unbalance, s1, s2, s3 and a1, where SC goes from negative values (bad clustering) to positive values, in some cases better than conventional algorithms (SC close to 1).

Table 5.6. SC for different clustering algorithms

		LMD	LMD-RD	KM	KMD	FCM	AHT	DBS
Dim1024	$\overline{v}$	**1.00**	**1.00**	**1.00**	**1.00**	0.58	**1.00**	**1.00**
	σ	0.00	0.00	0.00	0.00	0.03	0.00	0.00
Segment	$\overline{v}$	0.25	**0.57**	0.49	0.50	0.51	0.42	0.38
	σ	0.00	0.05	0.03	0.02	0.02	0.00	0.00
Hepta	$\overline{v}$	0.85	**0.88**	0.77	**0.88**	**0.88**	**0.88**	**0.88**
	σ	0.03	0.00	0.07	0.00	0.00	0.00	0.00
R15	$\overline{v}$	0.08	0.90	0.80	**0.90**	0.90	0.90	0.88
	σ	0.06	0.13	0.06	0.00	0.00	0.00	0.00
Aggreg.	$\overline{v}$	0.34	0.57	0.64	**0.64**	0.62	0.62	0.61
	σ	0.05	0.02	0.02	0.01	0.03	0.00	0.00
Unbalan.	$\overline{v}$	-0.1	**0.94**	0.90	0.89	0.78	0.81	0.94
	σ	0.08	0.03	0.07	0.07	0.07	0.00	0.00
s1	$\overline{v}$	-0.2	**0.85**	0.81	0.84	0.85	0.85	0.83
	σ	0.03	0.05	0.03	0.06	0.04	0.00	0.00
s2	$\overline{v}$	-0.2	0.72	0.74	0.75	0.78	0.74	0.59
	σ	0.03	0.06	0.04	0.05	0.03	0.00	0.00
s3	$\overline{v}$	-0.2	0.46	0.63	0.64	**0.65**	0.43	-0.3
	σ	0.02	0.04	0.02	0.02	0.02	0.00	0.00
a1	$\overline{v}$	--	0.65	0.72	0.72	0.72	0.68	0.54
	σ	--	0.05	0.02	0.03	0.03	0.00	0.00

LAMDA-RD obtains a performance comparable to the best clustering algorithms in datasets as Dim1024, and Hepta. Also is the best algorithm for Segment, Unbalance and s1, which are datasets of balanced and unbalanced distribution, with a maximum intra-cluster overlap of 9%. In the benchmarks R15, Aggregation and s2, LAMDA-RD presents results very close to the best value (KMD). In s3 and a1, the algorithm decreases its performance due to the dispersion of the individuals (the overlap increases). Nevertheless, based on SC, it is observed that LAMDA-RD, in s3 and a1 datasets, presents better results with respect to DBS.

The weakness of LAMDA-RD in datasets with high overlap occurs since the number of clusters to be built is unknown. It has the same problems as density techniques as DBS, which decrease their performance since they are not based on distance optimization criteria, like KM, FCM or AHT.

The σ in all cases allows to notice that similar results are obtained in each iteration. The worst case is given in R15, where $\sigma = 0.13$ reaches 14% of the average value, giving an idea of a correct behavior of the algorithm. The results of WB_{index} for each clustering algorithm are presented in Table 5.7, where the best average (lowest value) in each benchmark has been marked in bold text.

Table 5.7. WB_{index} for different clustering algorithms

		LMD	LMD-RD	KM	KMD	FCM	AHT	DBS
Dim1024	$\overline{v}$	**0.14**	**0.14**	**0.14**	**0.14**	2.24	**0.14**	**0.14**
	σ	0.00	0.00	0.00	0.00	0.07	0.00	0.00
Segment	$\overline{v}$	32.1	4.56	3.64	3.49	**3.45**	7.09	5.05
	σ	1.29	0.23	0.14	0.08	0.07	0.00	0.00
Hepta	$\overline{v}$	2.72	**1.76**	2.41	**1.76**	**1.76**	**1.76**	**1.76**
	σ	0.49	0.12	0.39	0.00	0.00	0.00	0.00
R15	$\overline{v}$	22.75	1.43	1.76	**1.42**	1.43	1.43	1.47
	σ	2.76	0.14	0.17	0.00	0.00	0.00	0.00
Aggreg.	$\overline{v}$	3.86	2.29	2.15	**2.11**	2.15	2.20	2.45
	σ	0.31	0.31	0.06	0.01	0.05	0.00	0.00
Unbalan.	$\overline{v}$	11.54	**1.03**	1.04	1.04	1.10	2.95	1.08
	σ	0.83	0.07	0.04	0.04	0.06	0.00	0.00
s1	$\overline{v}$	30.50	**1.77**	2.04	1.83	1.79	1.78	1.78
	σ	1.59	0.17	0.22	0.31	0.22	0.00	0.00
s2	$\overline{v}$	57.52	2.44	2.42	2.37	2.19	2.39	2.68
	σ	1.45	0.23	0.25	0.24	0.18	0.00	0.00
s3	$\overline{v}$	91.8	5.58	3.08	3.05	**2.92**	4.15	7.76
	σ	2.48	0.26	0.14	0.14	0.11	0.00	0.00
a1	$\overline{v}$	55.2	2.90	2.78	2.75	2.72	2.89	2.43
	σ	2.08	0.14	0.11	0.15	0.13	0.00	0.00

As in the previous metric, LAMDA-RD is the best for the WB_{index} in the datasets: dim 1024, Hepta and s1, where the individuals have a percentage of overlap under (9%), and in the case where the clusters are unbalanced (Unbalance).

In the datasets Segment, R15 and Aggregation, LAMDA-RD is very close to the best values, as explained before, in cases where there is no overlap between groups. For s2, s3 and a1, the performance of the method decreases, due to the presence of individuals in overlapping areas. The other methods can build better models because they know the number of clusters to build; this is evidenced by the results obtained with the methods KM, KMD, FCM, and AHT whose results are similar in the last three benchmarks. Small values of σ, again show that the repeatability in the experiments performed at each iteration is adequate.

Finally, we propose one way to determine the best algorithms with only one metric, we propose to compute the Performance Coefficient "P_C" defined in Appendix C and detailed in [107]. These values are shown in Table 5.8. The best result has been marked in bold text for each benchmark.

The results presented in Table 5.8 show that LAMDA-RD is the best algorithm for the following datasets: Dim 1024, Hepta, Unbalance and s1, which implies a correct clustering based on P_C. In Segment, and R15, LAMDA-RD has values very close to the best algorithm (KMD). The performance for s2, s3 and a1 is reduced in LAMDA-RD and DBSCAN, which is reasonable because they are based on densities, in which, if there are scattered individuals, then the algorithms cannot make a correct assignment in the clusters. Also, it can be

seen that our proposal makes an adequate clustering when the groups have not overlapping between them.

Table 5.8. P_C metric for different clustering algorithms

		LMD	LMD-RD	KM	KMD	FCM	AHT	DBS
Dim1024	$\overline{\boldsymbol{v}}$	**0.14**	**0.14**	**0.14**	**0.14**	3.92	**0.14**	**0.14**
	$\boldsymbol{\sigma}$	0.00	0.00	0.00	0.00	0.35	0.00	0.00
Segment	$\overline{\boldsymbol{v}}$	127	8.01	7.43	7.04	**6.78**	17.02	13.48
	$\boldsymbol{\sigma}$	15.5	0.16	0.59	0.42	0.44	0.00	0.00
Hepta	$\overline{\boldsymbol{v}}$	3.23	**1.99**	3.18	**1.99**	**1.99**	**1.99**	**1.99**
	$\boldsymbol{\sigma}$	0.69	0.29	0.71	0.00	0.00	0.00	0.00
R15	$\overline{\boldsymbol{v}}$	110	1.59	2.24	**1.58**	1.59	1.60	1.68
	$\boldsymbol{\sigma}$	32.0	0.17	0.40	0.00	0.00	0.00	0.00
Aggreg.	$\overline{\boldsymbol{v}}$	11.7	4.72	3.37	**3.28**	3.47	3.56	4.04
	$\boldsymbol{\sigma}$	2.91	2.44	0.19	0.04	0.25	0.00	0.01
Unbalan.	$\overline{\boldsymbol{v}}$	-34.3	**1.10**	1.16	1.18	1.43	3.66	1.17
	$\boldsymbol{\sigma}$	3.26	0.06	0.12	0.13	0.19	0.00	0.00
s1	$\overline{\boldsymbol{v}}$	-109	**2.04**	2.55	2.22	2.12	2.08	2.15
	$\boldsymbol{\sigma}$	10.4	0.34	0.37	0.56	0.35	0.00	0.00
s2	$\overline{\boldsymbol{v}}$	-290	3.40	3.29	3.20	**2.82**	3.22	4.53
	$\boldsymbol{\sigma}$	37.3	0.73	0.52	0.57	0.38	0.00	0.00
s3	$\overline{\boldsymbol{v}}$	-329	12.30	4.86	4.79	**4.49**	8.64	-23.6
	$\boldsymbol{\sigma}$	14.74	1.53	0.40	0.37	0.32	0.00	0.01
a1	$\overline{\boldsymbol{v}}$	--	4.47	3.89	3.84	3.78	4.26	4.53
	$\boldsymbol{\sigma}$	--	0.59	0.25	0.35	0.32	0.00	0.01

In the benchmark s1 (9% of overlapping), LAMDA-RD is the best algorithm, concluding that the performance of the algorithm is not affected by individuals slightly overlapped between clusters. Also, based on the metrics, we can note that LAMDA-RD perform a proper clustering process for the unbalanced datasets (unbalance). When the overlapping percentage increases, e.g., in s2 (20% overlap), the algorithm still makes a correct clustering; however, in the case of a1 and s3 (22% and 40% of overlapping,

respectively), based on the experiments, it is concluded that density-based methods have problems assigning individuals located in the overlap zone. KMD, FCM, and AHT have the advantage of knowing the number of clusters a priori, which makes it easier to assign those samples to the nearest cluster, e.g., in s3 ($P_C \approx 4.48$), while LAMDA-RD decreases its performance ($P_C \approx 12.3$), a value that shows that when there is an overlap greater than 20% between clusters, the proposal builds clusters with poor quality, incorrectly assigning individuals to the most similar clusters. Particularly, the proposal is better than DBS, the methods with which a fairer comparison can be made without setting the desired number of partitions.

Based on the P_C, the results are consistent with SC, and WB_{index}. LAMDA-RD presents works adequately if the overlapping between them is less than 20%. If the overlapping increases, then the iterative methods are better, which is logical due to their individual assignment methodology that allows minimizing distance functions at the intra-cluster, and maximizing inter-cluster distances; however, these iterative methods increase the computation time depending on the dimensions of the objects, and the dataset.

Finally, the quality of the clusters related to the real classes of each benchmark is evaluated with RI. The results are computed with the best partitions obtained with each algorithm. These values are shown in Table 5.9. The best (highest values) has been marked in bold text.

Table 5.9. RI metric for different clustering algorithms

	LMD	LMD-RD	KM	KMD	FCM	AHT	DBS
Dim1024	**1.00**	**1.00**	**1.00**	**1.00**	0.58	**1.00**	**1.00**
Segment	0.53	0.43	0.62	0.74	**0.84**	0.34	0.01
Hepta	0.84	**0.99**	0.94	0.91	0.89	**0.99**	**0.99**
R15	0.13	**0.99**	0.55	0.99	0.95	0.98	0.42
Aggreg.	0.12	0.81	0.73	0.70	0.57	**0.99**	0.91
Unbalan.	0.08	**1.00**	0.88	0.88	0.83	0.61	**1.00**
s1	0.10	**0.99**	0.83	0.80	0.97	0.98	0.86
s2	0.09	0.89	0.83	**0.96**	**0.96**	0.87	0.73
s3	0.08	0.52	0.80	**0.84**	0.83	0.70	0.39
a1	0.07	0.83	0.86	0.93	0.90	**0.95**	0.78

The results of Table 5.9 show that clusters constructed by LAMDA-RD have a high value of coinciding with the real classes, taking into consideration that RI is an extrinsic clustering validation measure that compares the output of the clustering method and the real results (groups). LAMDA-RD is better than LAMDA in all benchmarks, and in some datasets like Dim 1024, Hepta, R15, Unbalance and s1, the results are as good as the best algorithms, and in some cases better than them (see R15, and s1). In the rest of datasets. The has problems in Segment dataset, in which a high number of descriptors is affecting the performance of the density-based methods (see the values of LAMDA-RD, DBS and HDB), so, an evaluation of the relevant descriptors should be made, discarding those that do not adequately characterize each group. Due to the distribution and different densities of the clusters of Unbalance (see the distribution of data in [150]), LAMDA-RD, DBS and HDB algorithms are the best since they can clearly distinguish each group due to the

separation that exists among them, without the existence of overlap.

Table 5.10, shows the relative error between the result of the best clustering algorithm based on the average value and LAMDA-RD.

Table 5.10. Relative error between the highest RI and LAMDA-RD metrics

Dataset	Best Value	LAMDA-RD	Relative Error (%)
Dim1024	1.00	1.00	0
Segment	0.84	0.43	48.81
Hepta	0.99	0.99	0
R15	0.99	0.99	0
Aggreg.	0.99	0.81	18.18
Unbalan.	1.00	1.00	0
s1	0.99	0.99	0
s2	0.96	0.89	7.29
s3	0.84	0.52	38.10
a1	0.95	0.83	12.63

The results of Table 5.10 show that LAMDA-RD presents the largest errors in segment (48.81%) and s3 (38.10%), these errors due to the overlap presented by their respective samples. Relative errors of less than 20% are evidenced for all other datasets, and in some cases the relative error is 0%, which allows us to validate our algorithm in the clustering context.

5.1.2.2 Performance comparison of LAMDA-RD and other online clustering algorithms

To analyze and determine how LAMDA-RD improves the behavior of LAMDA and other online clustering algorithms

that work with data stream, the following test is performed. Aditionally, it is presented the time consuming of each proposal in a streaming data scenario. In this context, a successful algorithm must consider the following restrictions [152]:

- Individuals continually arrive;
- There is no control in the order in which the individuals are generated;
- The size of a stream is (potentially) unbounded;
- Data objects are discarded after they have been processed.

All these restrictions are considered in this experiment, in which are tested the algorithms LAMDA-RD, LAMDA-TP, and ADDclustering, for online data streams [149] (the results of original LAMDA are presented in [107]). A maximum exigency parameter is set ($\alpha = 1$), because it is desired a strict behavior for the algorithms in the assignment process. The control parameters of LAMDA-RD (d_{nb} and D_t) have been heuristically set to obtain a number of clusters closer to the real classes in each dataset. The methodology used for this experiment is presented in Figure 5.4.

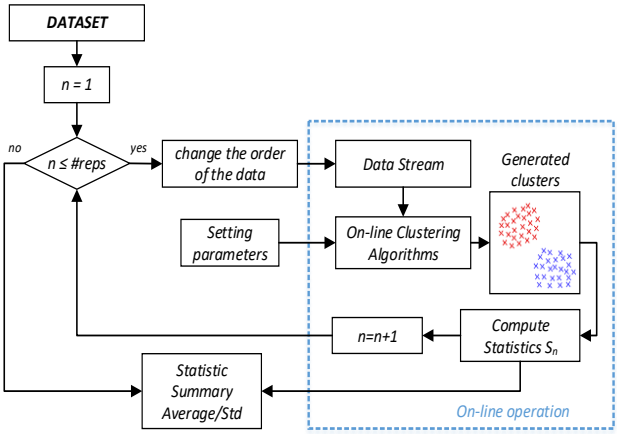


Figure 5.4. Methodology used for the comparison of the different online clustering algorithms

The experiment is repeated 20 times ($\#reps = 20$), each time performance metrics are computed. Finally, from the obtained results, the average and standard deviation are computed, to observe the repeatability in the creation of clusters of each online algorithm.

The results of the metrics are shown in Table 5.11, and the algorithm with the best average metric is marked in bold text. According to P_C, LAMDA-RD is the best in all the cases, even with high-dimensional datasets (see Postures), which shows an acceptable scalability of LAMDA-RD at the cost of increasing the computational time, which is common in data stream scenarios. It can be observed that this metric increases directly proportional when the percentage of overlap between clusters increases (see s1, s2, s3 and a1), which is expected because the clustering is more complex

since different individuals can belong to two or more clusters.

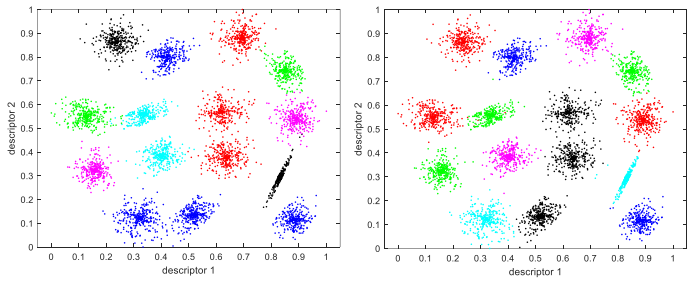
LAMDA-RD has the highest computational cost, due to the additional operations that are executed for the merging stage. This time depends especially on the number of individuals and the number of dimensions, e.g. Segment (65.23s, 2310 individuals and 19 features), and Postures (600.8s, 74975 individuals and 15 features). Additionally, it is observed that LAMDA-RD, with respect to ADDClustering, presents better results in all the benchmarks for online clustering. Evaluating P_C, LAMDA-RD is always the best, in Segment (LAMDA-RD: 65.23s and ADDClustering: 1.780s) ADDclustering is faster, which shows that this algorithm works better with several descriptors, decreasing its performance when the number of individuals increases (see Postures).

Table 5.11. Performance metrics of online clustering algorithms

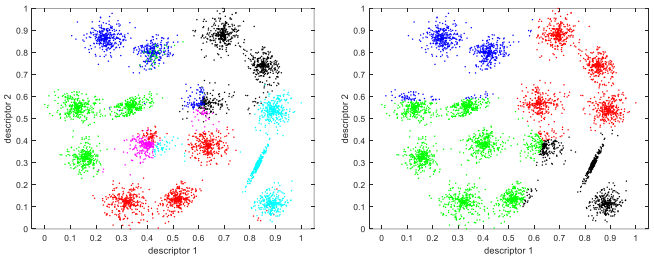
			SC	#Clus.	WB_{ind}	T (s)	PC
Segment	LMD-RD	$\overline{\boldsymbol{v}}$	**0,57**	**7,00**	**4,56**	65,2	**8,01**
		$\boldsymbol{\sigma}$	0,05	1,00	0,23	5,19	0,16
	LMD-TP	$\overline{\boldsymbol{v}}$	0,19	3,00	125	2,50	662
		$\boldsymbol{\sigma}$	0,03	1,00	6,24	0,33	37,48
	ADDC	$\overline{\boldsymbol{v}}$	0,55	3,00	4,57	**1,78**	8,31
		$\boldsymbol{\sigma}$	0,03	0,00	0,68	0,33	1,55
s1	LMD-RD	$\overline{\boldsymbol{v}}$	**0,85**	**15,0**	**1,77**	26,8	**2,04**
		$\boldsymbol{\sigma}$	0,05	1,00	0,17	4,23	0,34
	LMD-TP	$\overline{\boldsymbol{v}}$	0,47	**15,0**	4,17	**3,84**	9,41
		$\boldsymbol{\sigma}$	0,08	1,00	0,44	0,47	3,09
	ADDC	$\overline{\boldsymbol{v}}$	0,51	4,00	2,32	8,59	4,53
		$\boldsymbol{\sigma}$	0,07	0,00	0,02	0,00	0,01
s2	LMD-RD	$\overline{\boldsymbol{v}}$	**0,72**	**15,0**	**2,44**	31,84	**3,40**
		$\boldsymbol{\sigma}$	0,06	1,00	0,23	6,15	0,73
	LMD-TP	$\overline{\boldsymbol{v}}$	0,39	22,0	5,93	**3,92**	16,5
		$\boldsymbol{\sigma}$	0,09	2,00	0,69	0,33	6,65

	ADDC	$\overline{v}$	--	1,00	Inf	1,22	--
		σ	--	0,00	--	--	--
s3	LMD-RD	$\overline{v}$	**0,46**	30,00	**5,58**	30,25	**12,30**
		σ	0,04	1,00	0,26	1,48	1,53
	LMD-TP	$\overline{v}$	0,35	**24,0**	6,22	**3,88**	18,46
		σ	0,05	2,00	0,48	0,09	3,99
	ADDC	$\overline{v}$	--	1,00	Inf	1,25	--
		σ	--	0,00	--	--	--
a1	LMD-RD	$\overline{v}$	**0,65**	**20,0**	**2,90**	5,98	**4,47**
		σ	0,05	1,00	0,14	0,78	0,59
	LMD-TP	$\overline{v}$	0,32	17,00	5,59	**1,55**	17,81
		σ	0,05	1,00	0,44	0,08	3,74
	ADDC	$\overline{v}$	0,53	11,00	3,79	26,98	7,16
		σ	0,00	0,00	0,00	0,00	0,00
Postures	LMD-RD	$\overline{v}$	0,03	**6,00**	**16,0**	600	**497**
		σ	0,00	1,00	1,90	5,24	6,87
	LMD-TP	$\overline{v}$	0,05	25,0	28,8	**110,**	644
		σ	0,00	0,00	2,69	3,65	5,24
	ADDC	$\overline{v}$	**0,68**	2,00	6547	766	9672
		σ	0,03	0,00	10,25	10,98	35,24

An illustration of the obtained clusters with the different algorithms in s1, is presented in Figure 5.5.



(a) (b)



(c) (d)

Figure 5.5. Tests performed with s1 dataset; (a) Original partition, clusters generated by (b) LAMDA-RD (15 clusters), (c) LAMDA-TP (15 clusters), and (d) ADDclustering (4 clusters), see the detailed statistics in Table 5.9.

The parameters of LAMDA-RD (d_{nb} and D_t) hIave been calibrated to obtain the desired number of clusters. On the other hand, LAMDA-TP creates 15 clusters of poor quality because it incorrectly assigns individuals in different clusters (bad quality clusters). Finally, ADDClustering builds 4 clusters, and according to the results of Table 5.10 (for s1), it can be noted that the quality of the clusters is not as good as that obtained by LAMDA-RD, where all quality metrics are the best, e.g. $P_C = 2.04$. The method that follows is ADDClustering, with $P_C = 4.532$ (almost double), this is, the groups formed have better inter-cluster (the individuals in the same group are very similar to each other) and intra-cluster characteristics.

5.2 Tests of LAMDA as controller

In order to validate the LAMDA controller in the different proposals, in this subsection different case studies are addressed, based on the papers developed throughout this research, such as [124,136,138,153,154], in which its

behavior has been observed in various non-linear systems whose characteristics may be of interest in different applications, which shows the versatility of LAMDA-based controllers.

The results of Rule-based LAMDA are not shown in detail in this section because this work focuses on the comparison of LSMC, ZLSMC and Adaptive LAMDA proposals. These methods use the Rule-based LAMDA presented in subsection 4.1 as a basis for the design and implementation. For a more in-depth analysis of its application in the field of control, the reader is recommended to review our papers[124,154].

5.2.1 LSMC experiments

The proposal is validated in two SISO continuous nonlinear systems: 1) Temperature control of a continuous stirred tank reactor (CSTR) under bounded disturbances and reference changes, and 2) Regulation of a mixing tank with variable parameters (variable dynamics). The tests are compared with PID controllers (or their variants), Rule-based LAMDA [124,154] and SMC presented in [127]. Additionally, the scaling gains calibration of the LSMC is performed using two methods: Heuristic calibration and Offline calibration using PSO (as mentioned in the fifth objective of this work).

The purpose of making the comparison of both scaling gains calibration methods is to analyze if there is a considerable improvement in the performance of the controller in the tested processes.

5.2.1.1 Case Study 1

The system studied is a continuous stirred tank reactor (CSTR) presented in Figure 5.6, where the exothermic reaction $A \rightarrow B$ is carried out. To remove the heat of reaction, the reactor is surrounded by a jacket through which a cooling liquid flows. The temperature controller has been calibrated to operate in a range of 80 to 100$[°C]$.

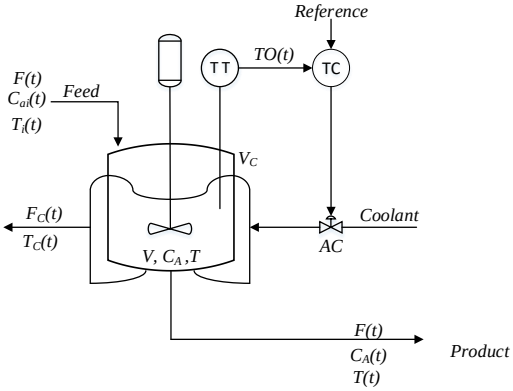


Figure 5.6. Studied process (CSTR)

For the system analysis, the following considerations are accepted [127]:

- Densities and heat capacities of the reactants and products are equal and constant.
- The heat losses from the jacket to the surroundings are negligible.
- The heat of reaction is constant.
- The liquid volume in the tank is constant.
- The jacket and the reactor are well mixed.

The mathematical equations that describe the dynamic behavior of the system are:

- Mole balance on reactant A

$$\frac{dC_A(t)}{dt} = \frac{F(t)}{V}\big(C_{ai}(t) - C_A(t)\big) - k{C_A}^2(t) \qquad (5.1)$$

- Energy balance on reactor contents

$$\frac{dT(t)}{dt} = \frac{F(t)}{V}\big(T_i(t) - T(t)\big) - k{C_A}^2(t)\frac{\Delta H_R}{\rho C_p} - \frac{UA}{V\rho C_p}\big(T(t) - T_C(t)\big) \qquad (5.2)$$

- Energy balance on the jacket

$$\frac{dT_C(t)}{dt} = \frac{UA}{V_C\rho_C C_{pc}}\big(T(t) - T_C(t)\big) - \frac{F_C(t)}{V_C}\big(T_C(t) - T_{ci}(t)\big) \qquad (5.3)$$

- Reaction rate coefficient

$$k = k_0 e^{-\frac{E}{R(T+273)}} \qquad (5.4)$$

- Temperature transmitter

$$\frac{dTO(t)}{dt} = \frac{1}{\tau_T}\left[\frac{T(t) - 80}{20} - TO(t)\right] \qquad (5.5)$$

- Temperature transmitter

$$F_C(t) = F_{C_{max}}\alpha^{-m(t)} \qquad (5.6)$$

where $C_A(t)$: concentration of the reactant in the reactor $[kgmol/m^3]$, $C_{ai}(t)$: concentration of the reactant in the feed $[kgmol/m^3]$, $T(t)$: temperature in the reactor $[°C]$, $T_i(t)$: temperature of the feed $[°C]$, $T_c(t)$: temperature of the jacket $[°C]$, $T_{ci}(t)$: coolant inlet temperature $[°C]$, $TO(t)$: transmitter output signal normalized from 0 a 1, $[fraction\ TO]$, $F(t)$:

process feed rate $[m^3/s]$, V: reactor volume $[m^3]$, k: reaction rate coefficient $[m^3/kgmol-s]$, ΔH_R: heat of reaction, assumed constant $[J/kgmol]$, ρ: density of the reactor contents $[kgmol/m^3]$, C_p: heat capacity of the reactants and products $[J/kgmol-°C]$, U: overall heat-transfer coefficient $[J/s-m^2-°C]$, A: heat transfer area $[m^2]$, V_C: the jacket volume $[m^3]$, ρ_C: density of the coolant $[kg/m^3]$, C_{pc}: specific heat of the coolant $[J/kg-°C]$, $F_C(t)$: coolant rate $[m^3/min]$, τ_T: time constant of the temperature sensor $[s]$, $F_{C_{max}}$: maximum flow through the control $[m^3/min]$, α: valve rangeability parameter, k_0: Arrhenius frequency parameter $[m^3/s-kgmol]$, E: activation energy of the reaction $[J/kgmol]$, R: ideal gas law constant $[J/kgmol-K]$, $m(t)$: the fraction of controller output, from 0 to 1 $[p.u]$. Table 5.11 shows the parameters in steady-state at the desired operating point of the CSTR.

Table 5.12. Steady-state values of the CSTR

Var.	Value	Var.	Value
$C_A(t)$	$1.1\ kgmol/m^3$	V_C	$1.82\ m^3$
$C_{ai}(t)$	$2.8\ kgmol/m^3$	$F(t)$	$0.45\ m^3/min$
T	$88°C$	$F_{C_{max}}$	$1.2\ m^3/min$
T_i	$66°C$	C_{pc}	$4184\ J/kg-°C$
T_{ci}	$27°C$	α	50
Ref.	$88°C$	τ_T	$0.33\ min$
ΔH_R	$-9.6e^7\ J/kgmol$	k_0	$0.07\ m^3/kgmol-s$
C_p	$1.81e^5\ J/kgmol-°C$	E	$1.182e^7\ J/kgmol$
U	$3550\ J/s-m^2-°C$	T_c	$50.5\ °C$
ρ_C	$1000\ kg/m^3$	$\overline{m}$	$0.254\ fraction\ CO$
A	$5.4\ m^2$	V	$7.08\ m^3$
ρ	$19.2\ kgmol/m^3$	R	$8314\ J/kgmol-K$

The model can be approximated to a First-Order Plus Dead Time (FOPDT) as presented in [127] with the form of :

$$\frac{X(s)}{U(s)} = \frac{Ke^{-t_o s}}{(\tau s + 1)} \tag{5.7}$$

where $X(s)$ is the Laplace transform of the controlled variable (the transmitter output), and $U(s)$ is the Laplace transform of the manipulated variable (the controller output), K is the process gain, τ is the process time constant, and t_o is the process dead time.

In [127], the parameter identification gives the following values: $K = 1.6, \tau = 13\ min, t_0 = 3.0\ min$, and for the design of the controller, the dead time t_0 is modeled using a first-order Taylor series approximation, as:

$$e^{-t_o s} \cong \frac{1}{t_0 s + 1} \tag{5.8}$$

Substituting (5.8) in (5.7), it is obtained:

$$\frac{X(s)}{U(s)} \cong \frac{K}{(\tau s + 1)(t_0 s + 1)} = \frac{K}{\tau t_0 s^2 + (\tau + t_0)s + 1} \tag{5.9}$$

Solving (5.9) in the time domain:

$$\tau t_0 \ddot{x} + (\tau + t_0)\dot{x} + x - Ku = 0 \tag{5.10}$$

The system represented in state-space, where $x_1 = x$, is:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{(\tau + t_0)}{\tau t_0} x_2 - \frac{1}{\tau t_0} x_1 + \frac{K}{\tau t_0} u \end{aligned} \tag{5.11}$$

Since this is a second-order differential equation, $n = 2$, from (4.9), $s(t)$ becomes:

$$s(t) = \left(\frac{d}{dt} + \lambda\right)^2 \int e(t)\, dt$$

$$= \left(\frac{d^2}{dt} + 2\lambda\frac{d}{dt} + \lambda^2\right) \int e(t)\, dt$$

$$= \dot{e}(t) + 2\lambda e(t) + \lambda^2 \int e(t)\, dt \qquad (5.12)$$

The derivative of (5.12) becomes:

$$\dot{s}(t) = \ddot{e}(t) + 2\lambda\dot{e}(t) + \lambda^2 e(t) = 0 \qquad (5.13)$$

For $n = 2$ in (4.18):

$$\ddot{e}(t) = \dot{x}_{d2}(t) - \dot{x}_2(t) \qquad (5.14)$$

Replacing (5.11) and (5.14) in (5.13):

$$\dot{s}(t) = \dot{x}_{d2}(t) + \frac{(\tau + t_0)}{\tau t_0} x_2 + \frac{1}{\tau t_0} x_1 - \frac{K}{\tau t_0} u + 2\lambda\dot{e}(t) + \lambda^2 e(t) = 0 \qquad (5.15)$$

Because $K > 0$, then $\frac{K}{\tau t_0} > 0$, so, based on (4.21), it is concluded that $b(X(t), t) > 0$. Thus, the rule tables to be used in this case study are presented in Tables 4.1 and 4.2. Figures 5.7 and 5.8 show the implemented rules to obtain continuous and discontinuous control actions.

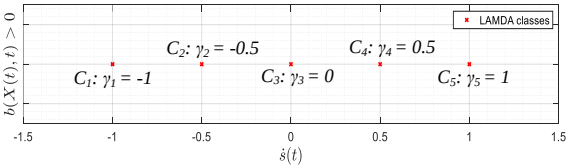


Figure 5.7. Classes and rules for u_c based on $\dot{s}(t)$ for the CSTR

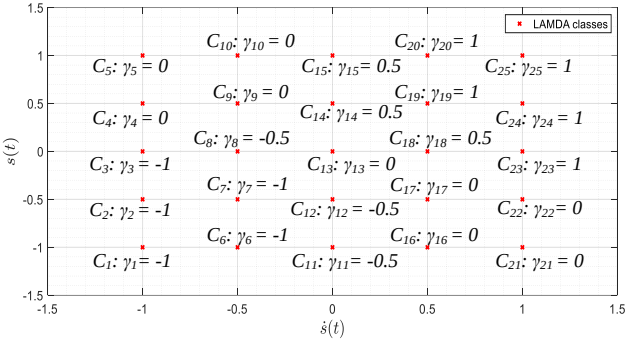


Figure 5.8. Classes and rules for u_d based on $\dot{s}(t)$ and $s(t)$for the CSTR

Four different approaches are tested to control the process, a conventional PID, a SMC controller proposed in [127], the LAMDA-PID presented in [154], and the LSMC controller. From [155], a PI is recommended when $t_0 < \tau/4$; for this reason it is implemented for the test instead of a PID. The controller parameters have been tuned considering the method of Dahlin synthesis, obtaining $K_C = 1.35$ and $\tau_I = 13\,min$. The SMC parameters have been tuned considering the method proposed in [127], these are: $\lambda_0 = 0.0421$, $\lambda_1 = 0.410$, $K_D = 0.96$, $\delta = 0.76$. The LAMDA-PID controller has the parameters $k_p = 0.028$, $k_i = 3.5$, $k_d = 0.25$, and for the LSMC, experimentally: $\lambda = 0.144$, $k_1 = 2.5 \times 10^{-4}$, $k_2 = 1.2$, $k_c = 1$ and $k_d = 3$. These values have been obtained empirically to decrease the ISE of the system.

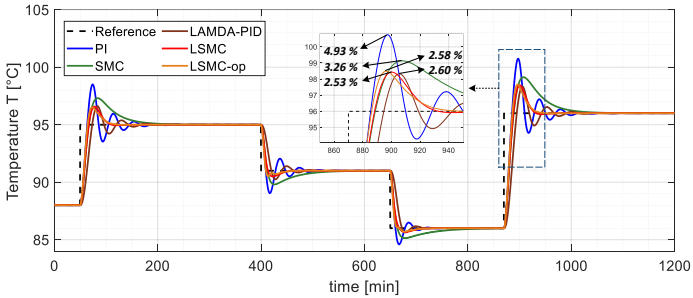
The PSO optimization has the following parametrization: $\#particles = 500$, $stalltime\ limit = 100$, $\#variables = 5$, $constraint\ 0 \leq m(t) \leq 1$. As result, the optimized parameters are: $\lambda = 0.121$, $k_1 = 2.389 \times 10^{-4}$, $k_2 = 1.284$, $k_c = 1$ and $k_d = 4.985$. The results of the controller with

heuristic calibration are labeled (LSMC), and the results with the optimized calibration are labeled (LSMC-op).

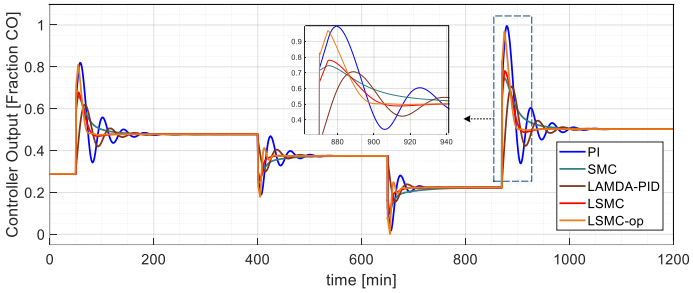
- Reference Change Test

Figure 5.9 shows the temperature output $T(t)$. At $50, 400, 650,$ and $870\ min$ step changes are applied to the reference, to analyze the tracking carried out by the controllers.

The results show that LSMC presents a smoother response than the other three methods in terms of control action and system output. Additionally, it is observed that when the greater magnitude reference change occurs, LSMC is the one that presents the minimum overshoot. The control action is similar to SMC and LAMDA and less abrupt than PI. An important point to note is how LSMC works much better than LAMDA-PID, with a less oscillatory response that quickly reaches the reference. Figure 5.10 summarizes the values of overshoot and settling time obtained in the reference change at time $870\ min$, and the ISE of the entire simulation to compare the values obtained by each controller.



(a)



(b)

Figure 5.9. (a) Comparative outlet temperature of the CSTR, (b) applied control actions

The results presented in the bars of Figure 5.10 show that the controller with the best performance is LSMC ($ISE_{LSMC} = 5.709$), with respect to the PI, SMC and LAMDA-PID controllers. This index is consistent with the values of settling time ($0.596h$) and overshoot (2.53%), which shows that the LSMC response is smoother and reaches the reference in a shorter time than the other control schemes.

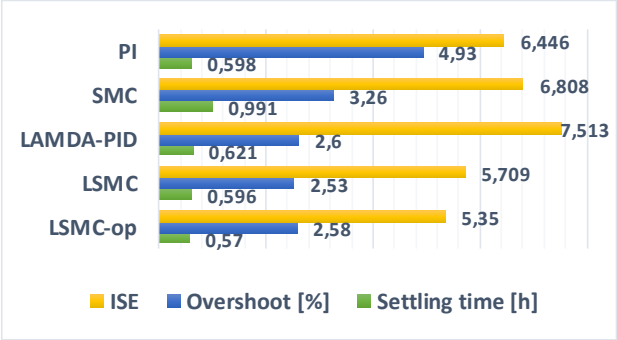
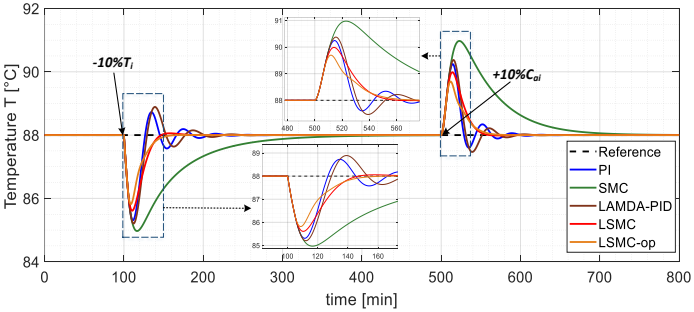


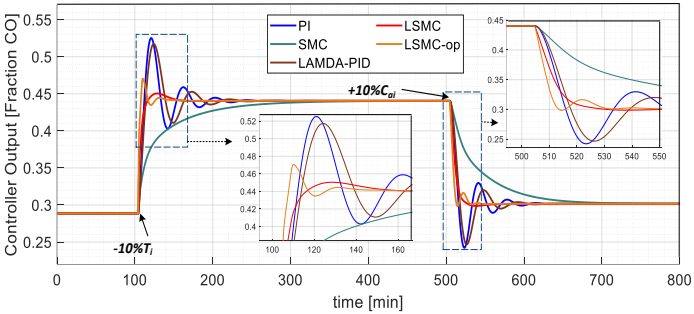
Figure 5.10. Comparative values of performance indexes of the controllers applied to the mixing tank

- Robustness Test

This test presents the response of the controllers when two disturbances are applied to the system. The first disturbance is applied at $100\ min$, reducing by 10% the temperature of the feed $T_i(t)$ as shown in Figure 5.11. At this point, it is seen that if T_i(t) decreases, then the controller output must be increased to close the AC valve, such that less refrigerant enters the jacket, and the temperature increases to the desired reference. The second disturbance is applied at $500\ min$ increasing at 10% the concentration of the reactant in the feed. At this point, it is seen that if C_{ai} increases, then the temperature in the reactor increases; therefore, it is required that the control action decreases to open the valve to allow more refrigerant in the jacket to reach the desired temperature. The PI, SMC, LAMDA-PID, and LSMC present the following performance indexes $ISE_{PI} = 0.402$, $ISE_{SMC} = 1.8$, $ISE_{LAMDA} = 0.448$, and $ISE_{LSMC} = 0.38$, respectively. Analyzing these results, it can be seen that the index of LSMC is the best and little lower than PI and LAMDA-PID, and much better than SMC. The control action of the LSMC is less abrupt than the PI and LAMDA-PID with few oscillations, which is a great advantage since there is no considerable effort required from the control valve, concluding that the proposed controller shows an outstanding behavior in terms of disturbance rejection.



(a)



(b)

Figure 5.11. (a) Comparative outlet temperature of the CSTR under disturbances, (b) applied control actions

5.2.1.2 Case Study 2

The system studied consists of a mixing of two fluids inside a tank where the volume of the tank varies freely without overflowing (see Figure 5.12). The system has a hot water stream $W_1(t)$ that mixes with a cold water stream $W_2(t)$ manipulated through a valve (actuator). The resulting mixture gives an output water stream $W_3(t)$, which must be at the desired temperature. The temperature transmitter is

installed at a distance of 125 $[ft]$ from the tank outlet, and has been calibrated to operate in a range of 100 to 200 $[°F]$. The distance between the tank outlet and the location of the temperature transmitter generates a time delay in the measurement. For the system analysis, the following considerations are accepted:

- The tank contents are well mixed.
- The liquid volume in the tank varies without overflowing.
- The pipe and the tank are well insulated.
- The main disturbance of the system is the hot stream $W_1(t)$.

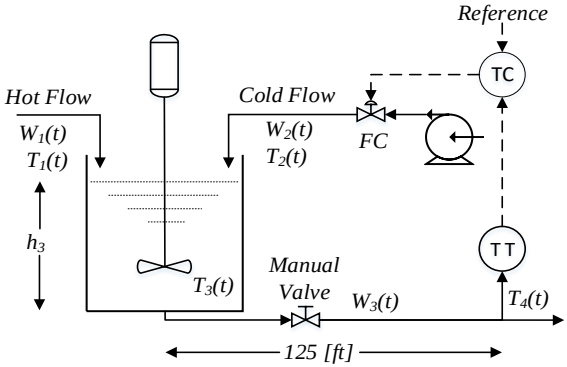


Figure 5.12. Studied process (Mixing Tank)

The mathematical equations that describe the dynamic behavior of the system are:

- Energy balance in the mixing tank

$$W_1(t)Cp_1T_1(t) + W_2(t)Cp_2T_2(t) - W_3(t)Cp_3T_3(t) = A_3Cv_3\frac{d\big(h_3(t)T_3(t)\big)}{dt} \qquad (5.16)$$

- Mass balance in the mixing tank

$$W_1(t) + W_2(t) - W_3(t) = A_3 \frac{dh_3(t)}{dt} \tag{5.17}$$

- Manual valve

$$W_3(t) = 11.8685 C_{VL3} \sqrt{h_3(t)} \tag{5.18}$$

- Pipe delay (between the location of temperature sensor) and the mixing tank

$$T_4(t) = T_3(t)\big(t - t_0(t)\big) \tag{5.19}$$

- Time delay (dead time)

$$t_0(t) = \frac{LA\varrho}{W_3(t)} \tag{5.20}$$

- Temperature transmitter

$$\frac{dTO(t)}{dt} = \frac{1}{\tau_T}\left[\frac{T_4(t) - 100}{100} - TO(t)\right] \tag{5.21}$$

- Control valve position

$$\frac{dV_p(t)}{dt} = \frac{1}{\tau_{V_p}}\left[m(t) - V_p(t)\right] \tag{5.22}$$

- Valve equation

$$W_2(t) = \frac{500}{60} C_{VL} V_p(t) \sqrt{G_f \Delta P_v} \tag{5.23}$$

where $W_1(t)$: mass flow of hot stream $[lb/min]$, $W_2(t)$: mass flow of cold stream $[lb/min]$, $W_2(t)$: mass flow of the output stream $[lb/min]$, C_p: liquid heat capacity at constant pressure, $[Btu/lb - °\mathrm{F}]$, C_v: liquid heat capacity at constant volume $[Btu/lb - °\mathrm{F}]$, h_3: tank content level $[ft]$, A: mixing tank cross-section $[ft^2]$, $T_1(t)$: hot flow temperature $[°F]$,

$T_2(t)$: cold flow temperature $[°F]$, $T_3(t)$: liquid temperature in the mixing tank $[°F]$, $T_4(t)$: temperature $T_3(t)$ considering the delay t_0 $[°F]$, t_0: dead time $[min]$, ϱ: density of the mixing tank contents $[lb/ft^3]$, C_{VL}: valve flow coefficient $[gpm/psi^{1/2}]$, $TO(t)$: transmitter output signal normalized from 0 a 1 $[p.u]$, $V_p(t)$: valve position, from 0 (closed valve) to 1 (open valve), $m(t)$: the fraction of controller output, from 0 to 1 $[p.u]$, G_f: specific gravity, ΔP_v: pressure drop across the valve $[psi]$, τ_T: time constant of the temperature sensor $[min]$, τ_{V_p}: time constant of the control valve $[min]$, A: pipe cross-section $[ft^2]$, L: pipe length $[ft]$.

Table 5.13 shows the parameters in steady-state at the desired operating point of the mixing tank.

Table 5.13. Steady-state values of the mixing tank

Variable	Value	Variable	Value
W_1	$250\ lb/min$	h_3	$4.26509\ ft$
W_2	$191.17\ lb/min$	C_{VL3}	$18\ gpm/ft^{1/2}$
W_3	$441.17\ lb/min$	C_{VL}	$gpm/psi^{1/2}$
Cp_1	$0.8\ Btu/lb-°F$	TO	$0.5\ p.u.$
Cp_2	$1.0\ Btu/lb-°F$	V_p	0.478
Cv_3	$0.9\ Btu/lb-°F$	ΔP_v	$16\ psi$
Cp_3	$0.9\ Btu/lb-°F$	m	$0.478\ p.u.$
T_1	$250\ °F$	G_f	1
T_2	$50\ °F$	τ_T	$0.5\ min$
T_3	$150\ °F$	τ_{V_p}	$0.4\ min$
ρ	$62.4\ lb/ft^3$	A	$0.2006\ ft^2$
A_3	$3.51692\ ft^2$	L	$125\ ft$

From [127], the model can be approximated to a FOPDT as presented in (5.7). In order to observe the behavior of the parameters K, τ and t_0, the procedure presented in [156] has been followed, in which it is proposed to vary the signal

$m(t)$ applied to the valve in successive step changes, from 0.1 to 0.9. Figure 5.13 shows how these parameters change as a function of the input signal $m(t)$, it is varying ascending (black) and descending (red).

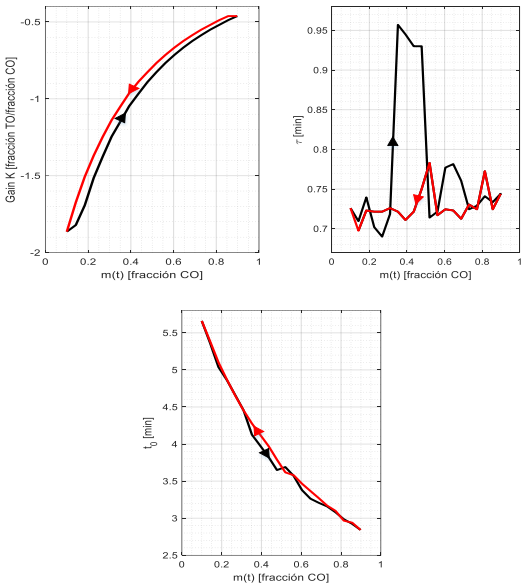


Figure 5.13. K, τ, and t_0 variations as $m(t)$ function for the mixing tank

As seen in Figure 5.13, the parameters change over the entire range of action of $m(t)$, increasing the non-linearity of the system, which is complex to model, so, it is considered a highly nonlinear model due to the time delay and the variation of the parameters, an ideal case study to test the LSMC proposal.

For the design of the controller, it is used the procedure from (5.7)-(5.15) presented in the case study 1 (CSTR), obtaining a second-order system.

Figure 5.13 has shown that the gain of the process is negative $K < 0$ for the entire variation range of $m(t)$, then $\frac{K}{\tau t_0} < 0$, so, based on (4.21), it is concluded that $b(X(t), t) < 0$. Thus, the rule tables to be used in this case study are presented in Tables 4.1 and 4.3. Figures 5.14 and 5.15 show the implemented rules to obtain the discontinuous and continuous control actions for this control system.

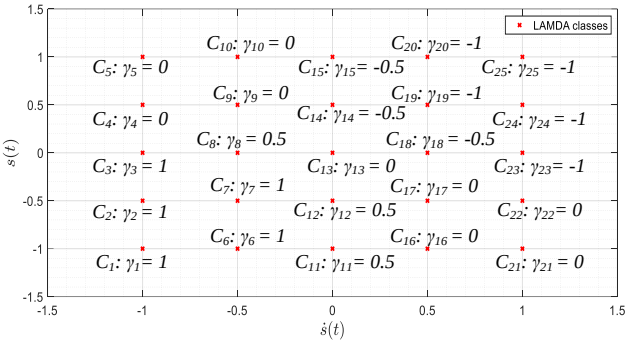


Figure 5.14. Classes and rules for discontinuous control action u_d based on $\dot{s}(t)$ and $s(t)$ for the mixing tank

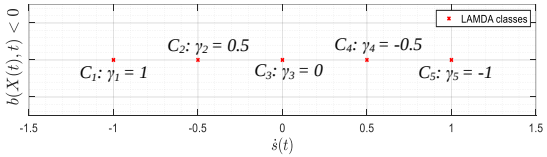
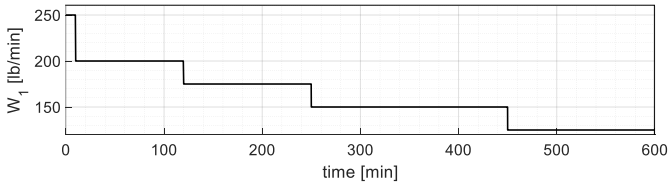


Figure 5.15. Classes and rules for continuous control action u_c based on $\dot{s}(t)$ for the mixing tank

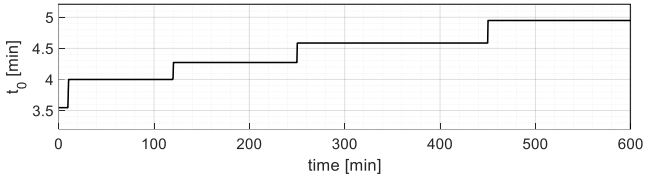
For this experiment, the controllers tested in the case study 1 are evaluated again. A PID, a conventional SMC, LAMDA-PID, and the LSMC controller. From [56] a PID is recommended when $t_0 > \tau/4$ as shown in Figure 5.13. The controller parameters have been tuned considering the method of Dahlin synthesis, obtaining $K_C = -0.17$, $\tau_I = 0.1$ and $\tau_D = 1.7$. The SMC controller parameters have been tuned considering the method proposed in [127], these are: $\lambda_0 = 0.60$, $\lambda_1 = 1.55$, $K_D = 0.25$, $\delta = 0.71$. The LAMDA-PID controller has the parameters $k_p = 0.25, k_i = 0.4, k_d = 2.5 \times 10^{-5}$ and for the LSMC, experimentally, it has been set $\lambda = 1$, $k_1 = 2.5 \times 10^{-5}$, $k_2 = 0.25$, $k_c = 5$ and $k_d = 0.55$. These values have been obtained empirically to decrease the ISE of the system.

The PSO parameterization is set to: $\#particles = 500$, $stalltime\ limit = 100$, $\#variables = 5$, $constraint\ 0 \leq m(t) \leq 1$. As result, the optimized values are: $\lambda = 0.814$, $k_1 = 2.34 \times 10^{-5}$, $k_2 = 0.284$, $k_c = 4.625$ and $k_d = 0.785$.

Figure 5.16a shows the change in the hot water stream W_1 from $250\ [lb/min]$ to $125\ [lb/min]$. The variation of this parameter is considered as a disturbance that changes the dynamics of the process. As an example, Figure 5.16b shows how the parameter t_0 is affected by the changes of W_1 for the system in open loop.



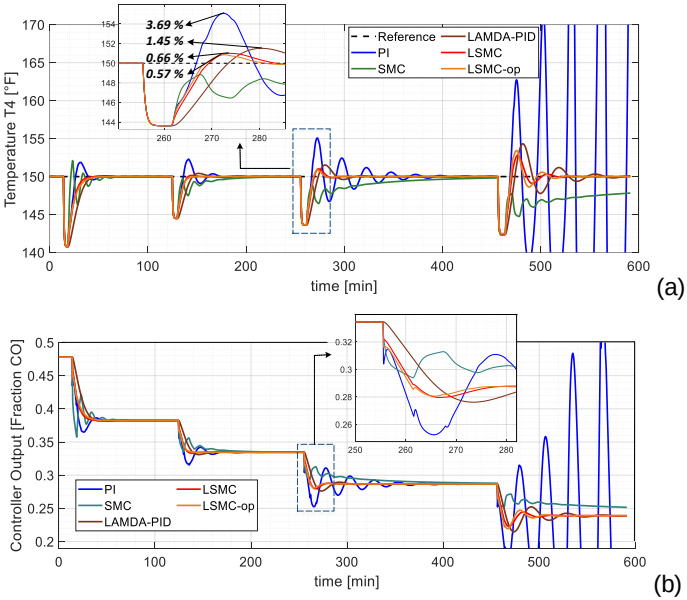
(a)



(b)

Figure 5.16. (a) Change of W_1, (b) change of dead time t_0

The results of the application of the controllers PID, SMC, LAMDA-PID, and LSMC controlling the studied mixing tank under the disturbances presented above are shown in Figure 5.17, in which the system outlet temperature and the applied control actions are presented.



(a)

(b)

Figure 5.17. (a) Comparative outlet temperature of the mixing tank, (b) applied control actions

As seen in Figure 5.17a, the LSMC controller regulates the output temperature effectively at $150\,[°F]$, with a small overshoot and in a short time, e.g., see the small overshoot (1.87%) when W_1 decreases to $125\ [lb/min]$, while the other proposals become oscillatory, unstable, or do not reach the reference.

The LAMDA-PID approach can regulate the process adequately with a moderate presence of oscillations. The PID controller can regulate the system in the disturbance at $10\,[min]$; however, it oscillates as W_1 decreases. In the disturbance that occurs at $450\ [min]$ ($W_1 = 125\ [lb/min]$), it is observed that the controller is not able to control the system, becoming unstable.

In the case of SMC, the process output is regulated during the first three changes of W_1. It is observed that when $W_1 = 150\ [lb/min]$, then the response of this controller is very slow; however, it reaches the reference. When $W_1 = 125\ [lb/min]$, then it is observed that the controller considerably decreases its performance and degrades without reaching the reference during the simulation time.

The zoom in Figure 5.17 shows in detail the behavior of the system output and control actions, where it is shown that LSMC reaches the reference quickly and with a smoother control action than the other proposals.

The bars of Figure 5.18 summarizes the values of overshoot and settling time obtained in the disturbance at time $250\ min$ (where all the controllers are stable) and the ISE of the entire simulation for the comparison.

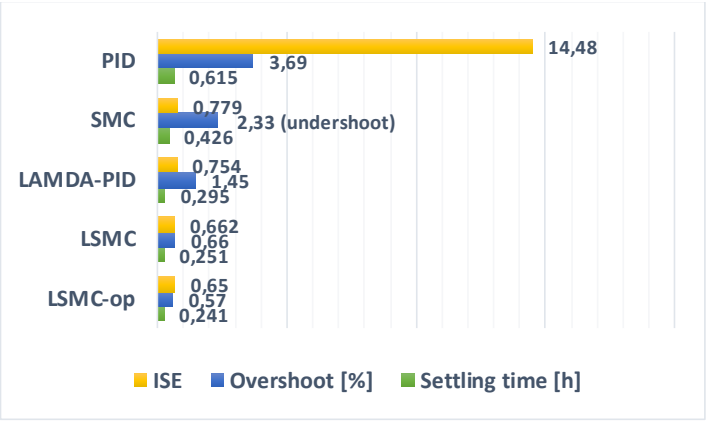


Figure 5.18. Comparative values of performance indexes of the controllers applied to the mixing tank

Figure 5.18 shows that the controller with the best performance is LSMC (0.662) with respect to the PID, SMC and LAMDA-PID controllers. This index is consistent with the values of settling time $(0.251h)$ and overshoot (0.66%). Considering that a lower ISE implies a better performance of the controller because the error converges to zero faster, it can be noted that the performance of the LSMC proposal is the best, without the need for a recalibration of the controller, a process that is required for PID and SMC, to avoid the degradation of their performances.

5.2.1.2.1 Sensitivity Analysis

Considering that the mixing tank case study is the most complex to control due to the variability of the model parameters which increases its nonlinearity, in this subsection, the sensitivity analysis of the LSMC proposal is presented, in order to observe how the number of classes

affect the performance of the controller. For general purposes, 5 classes ($c = 5$) for $s(t)$ and 5 classes for $\dot{s}(t)$ have been established, as detailed in Tables 4.1, 4.2, and 4.3. However, in this sub-section, the sensitivity of the controller is analyzed for a different number of classes. This experiment shows a comparison when three and seven classes are set in each variable, as follows:

- Three classes ($c = 3$) for $s(t)$ and three for $\dot{s}(t)$, defined as N: Negative, ZE: zero, and P: Positive. As detailed in the procedure of section 3, the classes and the control output are standardized between [-1,1], this is $N = -1$, $ZE = 0$, and $P = 1$.

- Seven classes ($c = 7$) for $s(t)$ and seven for $\dot{s}(t)$, defined as NB: Negative Big, NM: Negative Medium, NS: Negative Small, ZE: zero, PS: Positive Small, PM: Positive Medium, and PB: Positive Big. Also, the corresponding values are standardized between [-1,1]; this is: $NB = -1, NM = -0.66, NS = -0.33, ZE = 0, PS = 0.33, PM = 0.66$, and $PB = 1$.

In order to make a fair comparison, the input and output constants of the controller have been set at the previously calibrated values for 5 classes in each variable, as presented in subsection 5.2.1.2. These are $\lambda = 1$, $k_1 = 2.5 \times 10^{-5}$, $k_2 = 0.25$, $k_c = 5$ and $k_d = 0.55$.

The simulation of the mixing tank with controllers that handle different class numbers is shown in Figure 5.19. The graph presents the system output (showing the ISE) and the control actions obtained in each case.

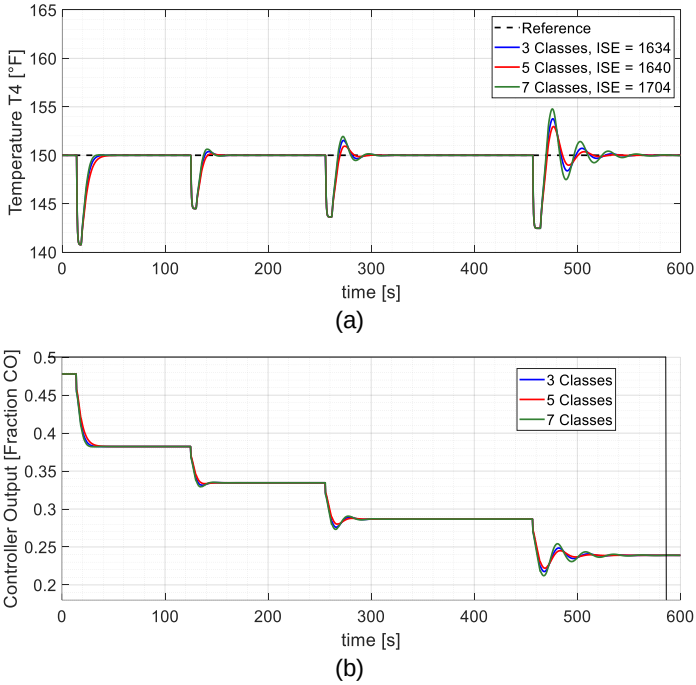


Figure 5.19. Controllers with different number of classes (a) outlet temperature of the process (b) applied control actions

Figure 5.19 shows that the responses of the controllers with 3 and 7 classes in $s(t)$ and $\dot{s}(t)$ present greater overshoot, and therefore, a more abrupt response with respect to the controller with 5 classes. The ISE for the controller with 3 classes is reduced by around 0.36% concerning the controller with 5 classes, which has a smoother response, being this its great advantage. The controller's response with 7 classes is more oscillatory; therefore, its ISE increases by around 3.9%, concerning the controller with 5 classes. Based on the results presented, it can be determined that

the controller's sensitivity varies minimally by changing the number of classes, without observing the degradation of the controller, or considerable changes in its behavior. The controller's best performance with 5 classes is because the input and output constants were calibrated with this number of classes. It is considered that better performance could be obtained in the case of 7 classes, by properly calibrating their centers and constants, which is a more complex process due to the existence of a greater number of variables, which would imply more time in the design and calibration stages.

5.2.1.3 Comparative Analysis of LSMC and ZLSMC

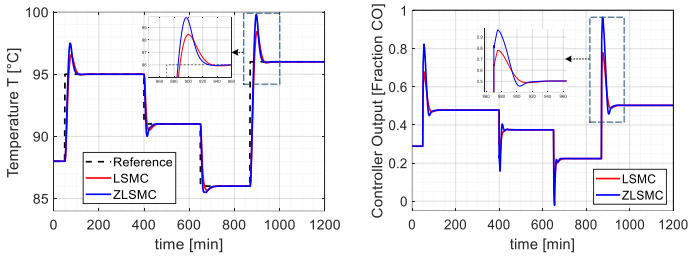
In this subsection, the two previous case studies are evaluated with the LSMC and ZLSMC controllers under heuristic calibration in each case, since the improvements with PSO consume more machine time and as has been observed, they have not represented a considerable improvement in the experiments.

The LSMC and ZLSMC controllers have been designed under the procedure described in subsections 4.2 and 4.4 respectively and have been set with the same scaling gains to obtain a fair comparison among them.

5.2.1.3.1 CSTR Process

In this process the parameters of LSMC and ZLSMC, empirically have been set to the following values: $\lambda = 0.144, k1 = 2.5 \times 10^{-4}, k2 = 1.2, k_c = 1$ and $k_d = 3$.

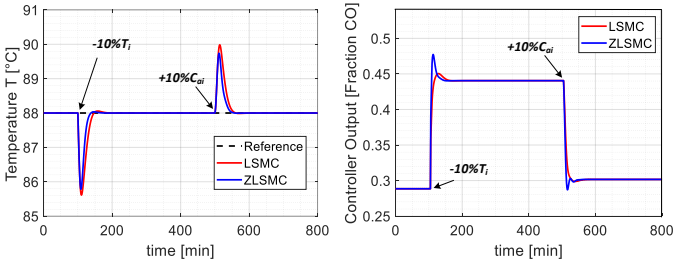
Figure 5.20 shows the temperature output $T(t)$ when step changes are applied to the reference to observe the response of the two controllers.



(a) (b)

Figure 5.20. (a) Comparative outlet temperature of the CSTR, (b) applied control actions of LSMC and ZLSMC

Figure 5.21 shows the temperature output $T(t)$ when disturbances are applied. The first disturbance is applied at 100 min, reducing by 10% the temperature of the feed $Ti(t)$ and the second disturbance is applied at 500 min increasing at 10% the concentration of the reactant in the feed.

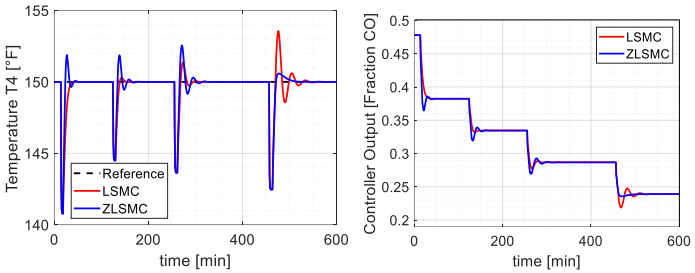


(a) (b)

Figure 5.21. a) Comparative outlet temperature of the CSTR under disturbances, (b) applied control actions of LSMC and ZLSMC

5.2.1.3.2 Mixing Tank Process

In this process the parameters of LSMC and ZLSMC, empirically have been set to the following values:: $\lambda = 1$, $k_1 = 2.5 \times 10^{-5}$, $k_2 = 0.25$, $k_c = 5$ and $k_d = 0.55$. Figure 5.22 shows the response of the system when the changes in the hot water stream W_1 presented in Figure 5.16 are applied.



(a) (b)

Figure 5.22. a) Comparative outlet temperature of the mixing tank, (b) applied control actions of LSMC and ZLSMC

Table 5.14 presents the ISE values and the percentage of improvement with respect to the best value. The results show that ZLSMC is always better than LSMC. In the case of the CSTR process at reference changes, the improvement is minimal (0.07%), however the performance of ZLSMC improves considerably (63.3%) when the disturbances are applied to the system, which shows that the Z-numbers theory in this process produces a less impulsive response, reaching the reference more quickly due to the fact that in the design stage the term "U" of reliability is considered when errors are large. In the other hand, for the mixing tank, ZLSMC is better than LSMC in a lower percentage (3.10%), however the strong point to highlight in

this case study is in the disturbance at $450\ min$, since as observed in the Figure 5.22, the control action of ZLSMC is smoother than the LSMC, this causes the overshoot to be minimal and without oscillation, which is a considerable improvement in terms of power consumption.

Table 5.14. Comparative ISE values among LSMC and ZLSMC

Process	Controller	ISE	Δ
CSTR (Reference change)	LSMC	5.709	0.07%
	ZLSMC	5.705	
CSTR (applied disturbance)	LSMC	0.3800	63.3%
	ZLSMC	0.2327	
Mixing Tank	LSMC	0.6636	3.10%
	ZLSMC	0.6437	

5.2.2 Adaptive LAMDA experiments

To validate the proposed controller, are addressed three case studies, each with different interesting properties, these are: the temperature control of the mixing tank, the regulation problem of an HVAC and the tracking trajectory of a mobile robot. In the experiments, it is demonstrated that the control strategy is experimentally stable and can be applied in systems with different dynamics. The results of these experiments are analyzed and compared with other fuzzy intelligent controllers that do not require the exact model of the plant to be designed, such as Fuzzy-PI (rule-based) and the LAMDA-PI (class-based [124],[154]). These two methods are static (non-adaptive) and are designed and calibrated based on the designer's expertise, which is generally complex and time-consuming. The aforementioned process is not required by the adaptive method, which is the main advantage, especially when the

system has unknown or variable dynamics. Additionally, the comparison is made with the online inverse learning control with ANFIS [77]. Comparative analysis allows identifying the advantages of the proposed method in the different systems. In the case of the adaptive schemes of LAMDA and ANFIS, the procedure for the validation is shown in Figure 5.23, which summarizes the training and application stages based on the schemes of Figures 4.6 and 4.7. The black dashed lines represent the learning and controller configuration parameters and the definition of the inputs. The solid lines are the sequential operation of the proposal in the tests.

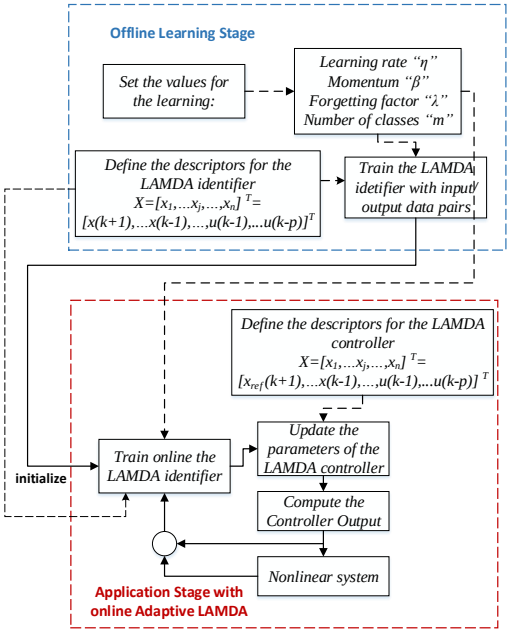


Figure 5.23. Implementation scheme of the online learning adaptive controllers

5.2.2.1 Case Study 1

HVAC systems are complex structures consisting of chillers, heat pumps, heating/cooling coils, boilers, air handling, thermal storage and liquid/air distribution units. It is a MIMO (Multiple-Input Multiple-Output) system with many variables whose modeling and dynamic study is complex due to its nonlinear characteristics [157]. Neuro-fuzzy systems are widely used in complex processes for modeling and control, however, its application in the HVAC systems is very limited [8]. The adaptive approach is simulated in the HVAC system presented by Arguello-Serrano and Velez-Reyes [158]. The main control objective in this simulation is to solve a regulation problem, analyzing and validating the proposed controller to abrupt disturbances in the thermal space variables (Zone 3 in Figure 5.24), these are: Temperature (T_3 $[°F]$) and Humidity Ratio (W_3 $[lb/lb]$).

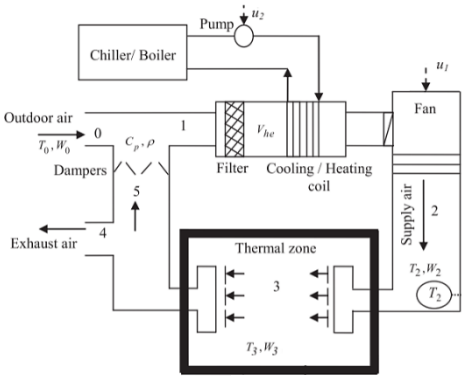


Figure 5.24. Block diagram of a simple HVAC system

The system operation is described as follows, outdoor air flows into the system mixing 25% of it with 75% of the

returning air, expelling the rest. The mixed air passes through a filter to the heat exchanger, where it is conditioned to the set point. The conditioned air is propelled to the thermal zone with a fan. The system requires to control the variables T_3 and W_3, simultaneously, based on thermal loads, by varying the fan speed, u_1, to regulate the airflow rate and the cold-water pumping rate, u_2, from the chiller to the heat exchanger [154]. The differential equations of energy and mass balances known from the conventional mathematical model of HVAC systems required for the simulations are:

$$\dot{T}_3 = \frac{f}{V_s}(T_2 - T_3) - \frac{h_{fg}}{C_p V_s}(W_s - W_3) + \frac{1}{0.25 C_p V_s}\left(Q_0 - h_{fg} M_0\right) \quad (5.24)$$

$$\dot{W}_3 = \frac{f}{V_s}(W_s - W_3) + \frac{M_0}{\rho V_s} \quad (5.25)$$

$$\dot{T}_2 = \frac{f}{V_{he}}(T_3 - T_2) - \frac{0.25f}{V_{he}}(T_0 - T_3) - \frac{f h_w}{C_p V_{he}}(0.25W_0 + 0.75W_3 - W_s) - 6000\frac{gpm}{\rho C_p V_{he}} \quad (5.26)$$

where, h_w is the enthalpy of liquid water, W_0 is the humidity ratio of outdoor air, h_{fg} is the enthalpy of water steam, V_{he} is the volume of the heat exchanger, W_s is the humidity ratio of supply air, W_3 is the humidity ratio of Zone 3, C_p is the specific heat of air, T_0 is the temperature of outdoor air, M_0 is the moisture load, Q_0 is the sensible heat load, T_2 is the temperature of supply air, T_3 is the temperature of Zone 3, V_s is the volume of Zone 3, ρ is the air mass density, f is the

volumetric flow rate of air (ft^3/min), and gpm is the flow rate of chilled water (gal/min). The assumptions made in the derivation of this mathematical model are also detailed in the study of Arguello-Serrano and Velez-Reyes [158].

Representing the system in state space notation for designing the control system, let be $f = u_1$ and $gpm = u_2$ the control actions that modify the target variables $x_1 = T_3$, $x_2 = W_3$, $x_3 = T_2$, $y_1 = T_3$, $y_2 = W_3$. The following parameters are defined to complete the model: $\alpha_1 = 1/V_s$, $\alpha_2 = h_{fg}/C_p V_s$, $\alpha_3 = 1/\rho C_p V_s$, $\alpha_4 = 1/\rho V_s$, $\beta_1 = 1/V_{he}$, $\beta_2 = 1/\rho C_p V_{he}$, $\beta_3 = h_w/C_p V_{he}$. The mathematical model from (5.24)-(5.26) can be re-written in the new form as:

$$\dot{x}_1 = u_1\alpha_1 60(x_3 - x_1) - u_1\alpha_2 60(W_s - x_2) + \alpha_3\left(Q_0 - h_{fg}M_0\right) \quad (5.27)$$

$$\dot{x}_2 = u_1\alpha_1 60(W_s - x_2) + \alpha_4 M_0 \quad (5.28)$$

$$\dot{x}_3 = u_1\beta_1 60(x_1 - x_3) + u_1\beta_1 15(T_0 - x_1) - u_1\beta_3 60(0.25W_0 + 0.75x_2 - W_s) - 6000u_2\beta_2 \quad (5.29)$$

$$y_1 = x_1 \quad ; \quad y_2 = x_2 \quad (5.30)$$

Tables 5.15 and 5.16 contain the numerical values chosen for the simulation and the system parameters at the operating point, respectively.

Table 5.15. Numerical values for system parameters

$\rho = 0.0074\ [lb/ft^3]$	$C_p = 0.24\ [Btu/lb°F]$
$T_{ref} = 55\ [°F]$	$T_{3ref} = 71\ [°F]$
$W_{3ref} = 0.0088\ [lb/lb]$	$V_{he} = 60.75\ [ft^3]$
$f_{ref} = 17{,}000\ [ft^3/min]$	$V_s = 58{,}464\ [ft^3]$
$W_s = 0.007\ [lb/lb]$	

Table 5.16. Numerical vales for system parameters at the operating point

$x_1^0 = 71\ [°F]$	$x_2^0 = 0.0092\ [lb/lb]$
$x_3^0 = 55\ [°F]$	$T_0^0 = 85\ [°F]$
$W_0^0 = 0.0018\ [lb/lb]$	$M_0^0 = 166.06\ [lb/hr]$
$u_1^0 = 17{,}000\ [ft^3/min]$	$u_2^0 = 58\ [gpm]$
$Q_0^0 = 289{,}897.52$	$W_s^0 = 0.007\ [lb/lb]$

To implement the LAMDA controllers in each of the control variables, it is necessary to analyze if a decoupling stage is required to implement independent controllers for each variable. The process to identify the correlation between inputs and outputs is based on the procedure of reaction curves to obtain the transfer functions, applying a step at one of the inputs, and monitoring the response at the outputs, obtaining the numerical values in the form of FOPDT system. The detailed procedure to obtain the transfer functions is presented in [154]. The linearized model can be represented by the $G(s)$ matrix:

$$X(s) = G(s)U(s) \tag{5.31}$$

$$G(s) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{9.8164 \times 10^{-4}\, e^{-0.0016}}{0.2137s + 1} & \dfrac{-1.3223\, e^{-0.0012s}}{0.2301s + 1} \\ \dfrac{-1.1764 \times 10^{-7}\, e^{-0.0011s}}{0.0527s + 1} & 0 \end{bmatrix} \tag{5.32}$$

From (5.32), the gains of the transfer function are obtained to form the gain matrix K.

$$K = \begin{bmatrix} 9.8164 \times 10^{-4} & -1.3223 \\ -1.1764 \times 10^{-7} & 0 \end{bmatrix} \tag{5.33}$$

The relative gain array RGA [159] (Bristol's matrix) is used to measure the interaction between the inputs and outputs in a multivariate process, and it is defined as:

$$RGA(K) = \Lambda(K) \triangleq K \times (K^{-1})^T \tag{5.34}$$

The operator × denotes the element-by-element multiplication:

$$\Lambda(K) = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{5.35}$$

$\Lambda(K)$ shows the dependence between the inputs and outputs. Based on these terms, the decoupling stage is not necessary for the control. Due to the HVAC system characteristics and the resulting parameters of $\Lambda(K)$, the control design with two independent LAMDA controllers, one for the temperature x_1 and another for the humidity ratio x_2, is feasible.

$$u_2 \rightarrow x_1 \; and \; u_1 \rightarrow x_2 \tag{5.36}$$

Finally, to proceed with the simulation of the proposed control, the HVAC system is discretized by applying the Euler method to equations (5.27)-(5.29) considering the sample time T_s:

$$\begin{aligned} x_1(k+1) = T_s[&u_1(k)\alpha_1 60\big(x_3(k) - x_1(k)\big) \\ &- u_1(k)\alpha_2 60\big(W_s - x_2(k)\big) \\ &+ \alpha_3\big(Q_0 - h_{fg}M_0\big)] + x_1(k) \end{aligned} \tag{5.37}$$

$$x_2(k+1) = T_s\big[u_1(k)\alpha_1 60\big(W_s - x_2(k)\big) + \alpha_4 M_0\big] + x_2(k) \tag{5.38}$$

$$x_3(k+1) = T_s[u_1(k)\beta_1 60\big(x_1(k) - x_3(k)\big) + u_1(k)\beta_1 15\big(T_0 - x_1(k)\big) - u_1(k)\,\beta_3 60(0.25W_0 + 0.75x_2(k) - W_s) - 6000u_2(k)\beta_2] + x_3(k) \quad (5.39)$$

Figure 5.25 shows the operational scheme of the control system with two separated control loops in the application stage, to regulate the two variables in the thermal space of the system.

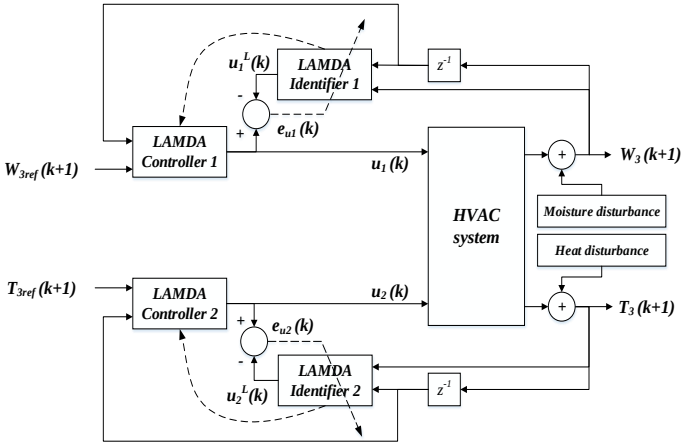


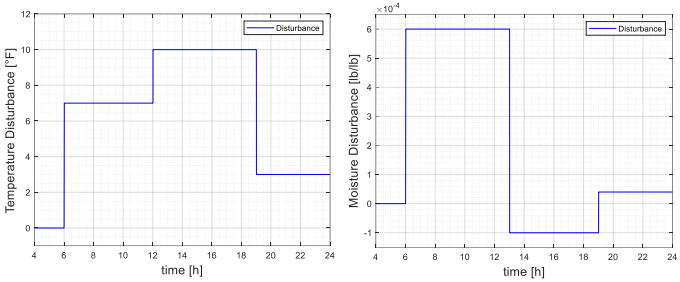
Figure 5.25. Adaptive control structure for the HVAC system

For the training stage, the LAMDA Identifier 1 uses two inputs $[W_3(k+1); W_3(k)]$ (because it has been considered a first-order plant based on [154]), each with two classes, the design parameters are $\eta_1 = 0.02$, $\beta_1 = 0.01$, $\lambda_1 = 0.997$. LAMDA Identifier 2 uses two inputs $[T_3(k+1); T_3(k)]$, each with two classes, and the design parameters are $\eta_2 = 0.95$, $\beta_2 = 0.01$, $\lambda_2 = 0.997$. The sampling period of the simulation

is $T_s = 1.2\,min$. A random input is applied to the plant that consists of 60 different random values of $54\,min$ duration.

In Figure 5.26, the online learning block (LAMDA Identifier 1) is placed between the control action u_1 and the output Humidity Ratio W_3, to learn the inverse model of the system using current and past information. The system output W_3 and its previous state are used as inputs for the identifier, in order to minimize the error $e_{u1}(k) = u_1(k) - u_1^L(k)$, where u_1^L is the output of the LAMDA identifier. The minimization of $e_{u1}(k)$ allows adjusting the parameters of the LAMDA model, which are updated in the controller at every sample time. The procedure described above is similarly applied to control the temperature T_3 of the Thermal Zone, considering the minimization of the error $e_{u2}(k) = u_2(k) - u_2^L(k)$, where $u_2^L(k)$ is the output of the LAMDA Identifier 2.

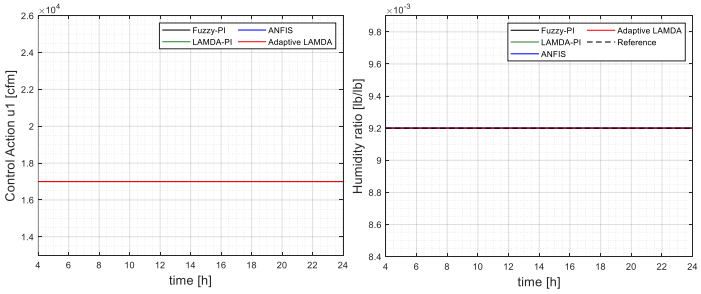
The performance of Adaptive LAMDA controller is analyzed by evaluating its response in the presence of abrupt disturbances, to test the robustness. The IAE is compared with the controllers Fuzzy-PI, LAMDA-PI and ANFIS. Fuzzy-PI and LAMDA-PI were designed based on the expertise of the plant. ANFIS and LAMDA based on online learning were set with the same values for the parameters applied to the learning stage. Two types of disturbances are applied to the HVAC system separately, to observe the behavior of each control variable in the Thermal Zone 3: Heat and Humidity Ratio (see Figures 5.47a and 5.47b, respectively).



(a) (b)

Figure 5.26. (a) Heat disturbance signal, (b) Moisture disturbance signal applied to robustness analysis

First, only the temperature disturbance is applied to the plant. Figure 5.27a shows that the control action u_1 stays at 17000 $[cfm]$, while the Humidity Ratio stays at 0.0092 $[lb/lb]$. Thus, the temperature disturbance does not affect W_3, as shown in (5.37)-(5.39).

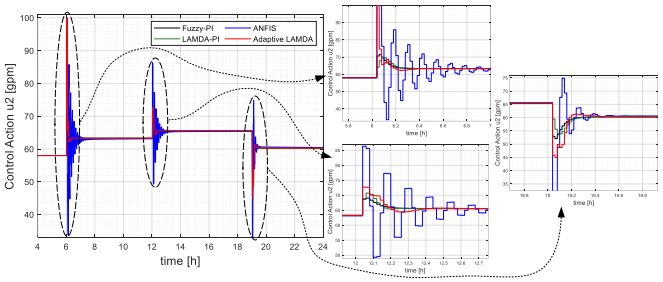


(a) (b)

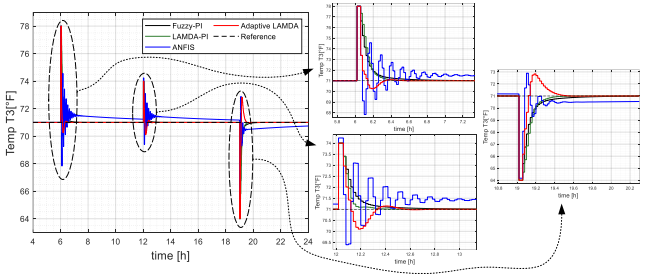
Figure 5.27. Comparative results with temperature disturbance: (a) control action u_1, (b) Humidity Ratio W_3

Figure 5.28a shows the behavior of the control action u_2 and Figure 5.28b shows the variation of the temperature T_3 when the temperature disturbance is applied.

The calibration of the non-adaptive methods in this system has been complex, and it takes a lot of time for this process because there are two controllers and several parameters to set, such as gains, the classes and their values. These methods control the plant properly when abrupt disturbances (the worst conditions) are applied to the output temperature, as shown in Figure 5.28.



(a)



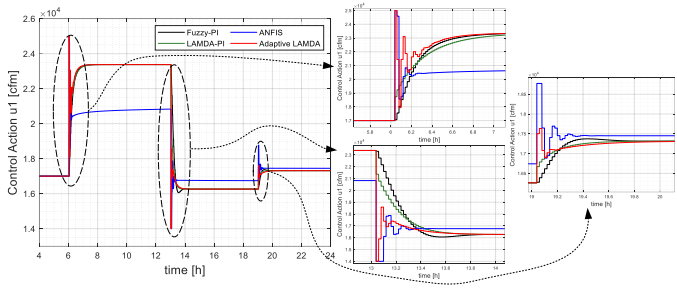
(b)

Figure 5.28. Comparative results with temperature disturbance: (a) control action u_2, (b) Temperature T_3

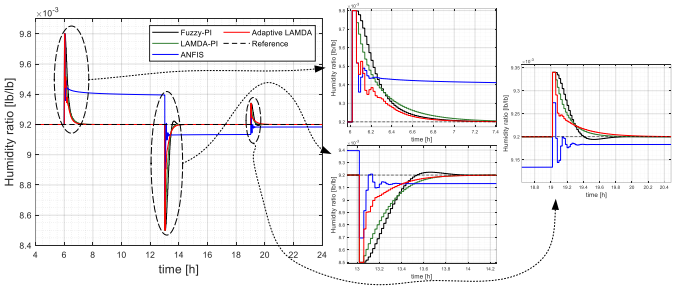
The adaptive proposal avoids the problems exposed in the design of the non-adaptive controllers. The control signal of Adaptive LAMDA is abrupt with respect to the Fuzzy-PI and LAMDA-PI controllers, due to the learning parameters

selected for this experiment. See the zoom in Figure 5.28, where the effectiveness of the proposal is evaluated qualitatively, highlighting the transient in the response of the controllers, and observing that the method responds quickly without error in steady-state. The ANFIS control being the most similar to Adaptive LAMDA, has a fairly abrupt and oscillatory response, which consequently leads to greater overshoot and presents a steady-state error ($\pm 0.5[°F]$).

In the next experiment, the moisture disturbance is applied to the plant in order to analyze how the variables T_3 and W_3 are affected, and how the controllers are able to regulate them. Figure 5.29a shows the control action u_1, and Figure 5.29b shows the behavior of Humidity Ratio W_3 for all the analyzed controllers. In Figure 5.30a is presented the control action u_2 and in Figure 5.30b the behavior of the Temperature T_3 for all the analyzed controllers is shown.

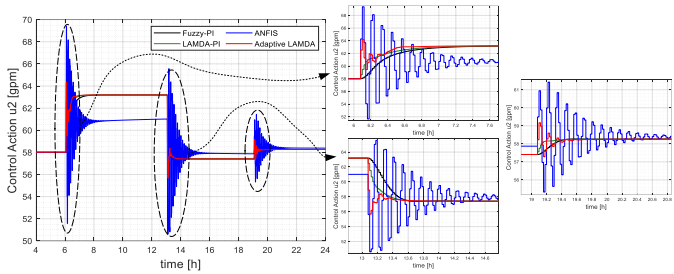


(a)

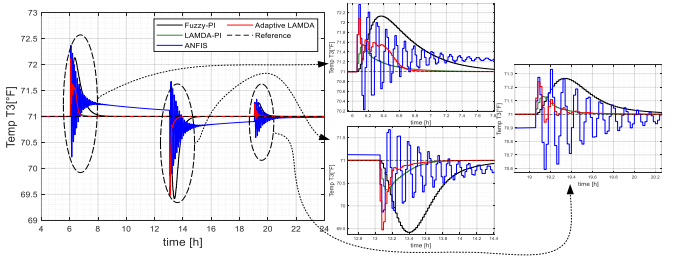


(b)

Figure 5.29. Comparative results with moisture disturbance: (a) control action u_1, (b) Humidity Ratio W_3



(a)



(b)

Figure 5.30. Comparative results with relative humidity disturbance: (a) control action u_2, (b) Temperature T_3

Figures 5.29 and 5.30 show that the abrupt moisture disturbance applied to the Humidity Ratio, affects the variables W_3 and T_3. For the Humidity Ratio, it can be seen that the output of the Adaptive LAMDA control presents few oscillations in the transient response, and it is quick to reach the reference (fast convergence), without overshoot because the control action is not abrupt. This demonstrates that the online learning performed by the algorithm when the system is subjected to disturbances is adequate (see the zoom in Figures 5.29 and 5.30). The convergence of the approach is better than the non-adaptive methods and the transient response is faster. The calibration of the non-adaptive methods in this case has an extra complexity degree due to the interaction between the input and output variables, which is avoided with Adaptive LAMDA. It is also observed that this approach is much better than the ANFIS controller, which is oscillatory and not able to reach the reference, presenting an error in a steady-state of around $\pm 0.2 \times 10^{-3} [°lb/lb]$, especially with the disturbances at time $6h$ and $13h$. The selected learning parameters of the algorithm are adequate for the HVAC system, and the algorithm works very well with only two classes per descriptor, reducing the computational time.

Figure 5.30 shows that Adaptive LAMDA can regulate the Temperature T_3 properly. Control actions of Adaptive LAMDA are less abrupt than Fuzzy-PI controller (black line), and stabilize the system in a very short time as desired in control systems, which allows to conclude that the selection of learning parameters is adequate for rapid convergence without overshoot in response, such as LAMDA-PI (method

that requires a complex calibration). ANFIS, as in the previous cases, is the controller with the most oscillatory response. In this case, the steady-state error is around $\pm 0.25°[F]$ for the two initial disturbances, and for a disturbance at time $19h$ the error is zero, but reaching this value in a longer time compared to the other proposals, which is not useful in these systems.

A quantitative analysis, computing the IAEs after applying the temperature and moisture disturbances to the plant, is shown in Table 5.17.

For the temperature disturbance, the Adaptive LAMDA controller is the best (minimum value). For the moisture disturbance, Adaptive LAMDA is the best controlling this variable, and the second-best to control the temperature, with the advantage that it does not require a tuning method for the parameters and a previous knowledge of the plant.

The LAMDA-PI controller presents competitive results due to the fact that an exhaustive knowledge engineering has been used to establish the classes and rules on which the control actions are defined, which is a process that requires time and must be properly calibrated. It is not required by Adaptive LAMDA, being this its main advantage.

In the case of the temperature disturbance, since the humidity ratio variable is not affected, there is no reference change in that variable, as shown Figure 5.27. Because of this, the IAE in controller 1 is zero in all cases.

Table 5.17. Numerical values for IAE for the HVAC experiments

IAE computed with Temperature Disturbance				
Contro -ller	**Fuzzy-PI**	**LAMDA-PI**	**ANFIS**	**Adaptive LAMDA**
1	0	0	0	0
2	2.04	1.911	7.689	1.529
IAE computed with Moisture Disturbance				
	Fuzzy-PI	**LAMDA-PI**	**ANFIS**	**Adaptive LAMDA**
1	3.7×10^{-4}	3.4×10^{-4}	1.9×10^{-3}	2.2×10^{-4}
2	1.879	0.377	2.912	0.492

5.2.2.2 Case Study 2

Here, to test the controller it is used a similar mixing tank presented in subsection 5.2.1.2. It consists of mixing two fluids inside a tank where the liquid volume in the tank is considered constant $V = 15\ ft^3$ and manual valve is not placed therefore $W_3(t) = 0$. The mathematical equations that describe the dynamic behavior of the system and change with respect to the case study analyzed above are:

- Energy balance in the mixing tank

$$W_1(t)Cp_1T_1(t) + W_2(t)Cp_2T_2(t) - \big(W_1(t) + W_2(t)\big)Cp_3T_3(t) = V\varrho Cv_3\frac{dT_3(t)}{dt} \quad (5.40)$$

- Time delay (dead time)

$$t_0(t) = \frac{LA\varrho}{W_1(t) + W_2(t)} \quad (5.41)$$

As proposed [127], the system can be approximated to a FOPDT as shown in (5.7) and modeling the dead time using

a first-order Taylor series approximation, the model of the process in the time domain is:

$$\frac{X(s)}{U(s)} \cong \frac{K}{(\tau s+1)(t_0 s+1)} \tag{5.42}$$

Resulting in a second-order system, in a discrete-time:

$$\frac{X(z)}{U(z)} \cong \frac{a_t z+b_t}{c_t z^2+d_t z+f_t} \tag{5.43}$$

where a_t, b_t, c_t, d_t, f_t are functions dependent on K, τ, t_0. The discrete-time model is variable and it depends on these parameters. Developing (5.43), considering the time k:

$$u(k) = (1/a_t)[c_t x(k+1) + d_t x(k) + f_t x(k-1) - b_t u(k-1)] \tag{5.44}$$

It is important to clarify that if the plant model is completely unknown, the number of previous states (p and q) of both x and u could be taken experimentally, until obtaining an adequate adjustment in the training stage.

For the initial training process, the algorithm parameters have been set with the following values $\eta = 0.00025$, $\beta = 0.001$, $\lambda = 1$, and a sampling period $T_s = 0.4\ min$. The inputs are $[x(k+1); x(k); x(k-1); u(k-1)]$, each with two classes. A random input is generated for the plant that consists of a sinusoidal signal from 0 to $200\ min$, and 100 different random step values of $143.6\ min$ duration. Figure 5.31 presents a comparison between the two techniques that require a training stage (ANFIS and Adaptive LAMDA). It is observed that LAMDA presents less error with better adjust to the actual data $u(t)$ applied to the system.

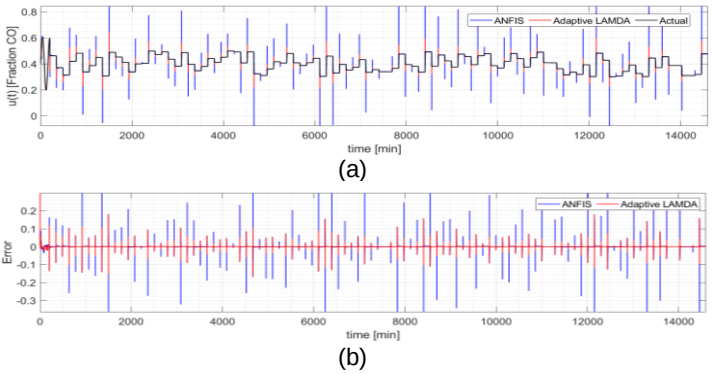


Figure 5.31. Comparison of learning algorithms a) adjust in the training stage with actual output data b) error

Once LAMDA has been trained, the proposed controller is tested in the plant under disturbances produced by varying hot stream $W_1(t)$. As shown (5.40), the dead time changes depending on W_1. The changes of the hot stream and the dead time have been presented in Figure 5.16. The variable dead time causes the dynamics of the system also changes, so it is appropriate to use an adaptive method to test the adaptive controller.

The complete control scheme for the mixing tank with variable dynamic, is presented in Figure 5.32, which shows the online identification and the control blocks. The LAMDA identifier inputs based on (5.44) are $[x(k+1); x(k); x(k-1); u(k-1)]$, and in the LAMDA controller the inputs are $\left[x_{ref}(k); x(k); x(k-1); u(k-1)\right]$.

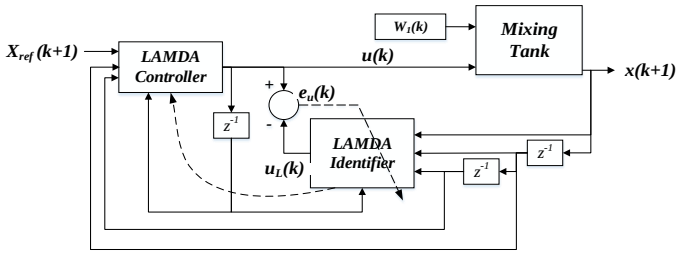
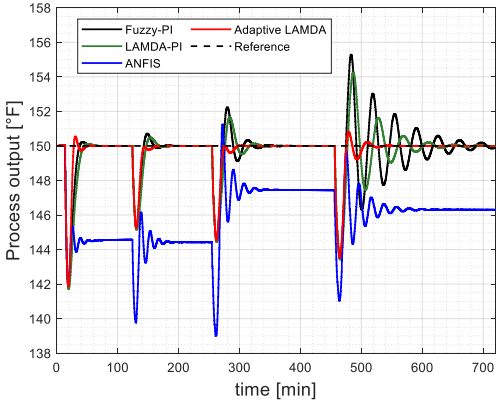


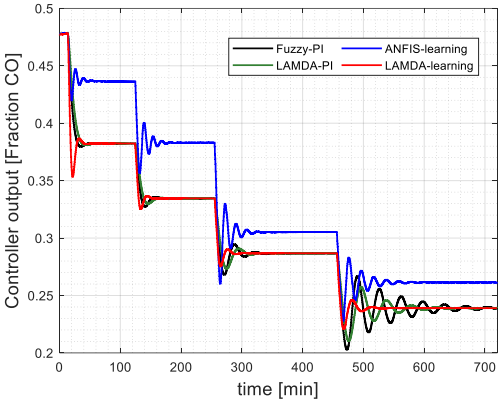
Figure 5.32. Adaptive control structure for the Mixing Tank

The tests are carried out for the controllers Fuzzy-PI, LAMDA-PI, ANFIS and the Adaptive LAMDA. Figure 5.33 shows qualitatively the effectiveness of the proposal. The Adaptive LAMDA convergence to the reference is faster when W_1 changes abruptly, taking into account that the controller design has not required the plant model or a calibration stage, as in the case of the Fuzzy-PI and LAMDA-PI controllers. It has been observed that the proposed learning algorithm with two classes per descriptor is able to control the system properly, with the advantage that the computational time is less with respect to the use of more classes. Although techniques without learning respond well to the control of this system, they have an oscillatory response with higher overshoot, especially in the case of the disturbance at time $450min$, which causes the system dead time to change abruptly (see Figure 5.33a). At this point, the adaptive LAMDA corrects it in less time without requiring additional calibration, since it adapts to these changes automatically with the LAMDA identifier block that learns online. Disturbances at time $10\ min$, $120\ min$ and $250\ min$ are quickly corrected with minimal overshoot by Adaptive LAMDA with excellent performance. Additionally, it can be observed that the non-learning methods (Fuzzy-PI and

LAMDA-PI) degrade their response considerably as the plant changes, which is an important aspect of the performance of the system. Adaptive LAMDA does not degrade, and its control action is smoother than the other controllers, which is an important advantage since in the real system the actuator is not overstressed (see Figure 5.33b). In the case of ANFIS, it is observed that it maintains an error in a steady-state, that is, the algorithm is not able to reach the reference. One solution would be to place an additional integration stage to correct it, which would increase the computational time and the complexity of the controller design.



(a)



(b)

Figure 5.33. (a) Comparative response of the system (temperature) (b) Applied control actions

The effectiveness of the obtained results is quantitatively evaluated by the computation of the IAE presented in Table 5.18. Additionally, the percentage change Δ from the best IAE value is computed to observe the improvement in performance terms.

Table 5.18. IAE of the controllers applied to the mixing tank process

Index	Fuzzy PI	LAMDA PI	ANFIS	Adaptive LAMDA
IAE	5.719	5.185	34.82	2.809
Δ	68.25%	59.44%	170.1%	-

The index with the lowest value is Adaptive LAMDA, because it reaches the reference quickly and with lower overshoot. In the presence of disturbances, it can be seen that the adaptive proposal is better at around 60%, with

respect to Fuzzy-PI and LAMDA-PI, and in 170 % with respect to ANFIS (the most similar approach) since this method is not able to reach the reference. These percentages show the potential of the learning algorithm in these types of systems.

5.2.2.3 Case Study 3

Finally, to validate the proposed controller in tracking trajectory tasks, its application in a mobile robot is presented. Trajectory control of a mobile robot is one of the objectives to be achieved in the field of autonomous robotics, due to the large number of associated applications as: risky or hazardous tasks for the humans, defense, medical, automation of industries and processes, among others [160]. Because the dynamic models of these systems are complex to obtain, or could present errors, it is necessary to design robust controllers that can compensate for these problems. For this reason, Adaptive LAMDA is applied to these systems, in which the algorithm will learn from the dynamics of the system (which is completely unknown) for the development of the controller based on the inverse model, without requiring previously the dynamic robot model, in order to apply it to the task of tracking different trajectories in a robot simulation environment.

5.2.2.3.1 Robot Model

The unicycle type robot is widely used in the field of automatic control due to its fast and nonlinear dynamics. Figure 5.34 shows a representation of the robot, where v and ω are the linear and angular velocities, respectively, h is the point of interest with x, y coordinates in the XY plane, ψ is the orientation of the robot, a is the distance between h

and the central point of the virtual axis B that connects the wheels, and r_1 is the radius of the wheels.

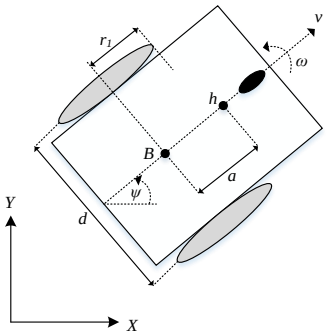


Figure 5.34. Parameters of the unicycle-like mobile robot

The complete mathematical representation of the mobile robot consists in the kinematic and the dynamic model. The general discretized kinematic model, assuming that the disturbance term is a zero vector and considering that T_s is the sample time [161], is:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \psi(k+1) \end{bmatrix} = T_s \begin{bmatrix} cos\psi\,(k) & -a\,sin\psi(k) \\ sin\,\psi\,(k) & a\,cos\psi(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} x(k) \\ y(k) \\ \psi(k) \end{bmatrix} \tag{5.45}$$

In [19] is proposed the application of two controllers, one of them based on feedback linearization for the robot kinematics, and the other one based on the dynamics. For the design, the dynamic model as unknown (black box) is considered, thus, its identification and control is done with the Adaptive LAMDA, which is the main contribution in this experiment.

5.2.2.3.2 Kinematic Controller

The kinematic controller (5.46) is based on the robot kinematics (5.45), considering the coordinates of the point of interest $[x, y]^T$. The control law is:

$$\begin{bmatrix} v_{ref}^c(k) \\ \omega_{ref}^c(k) \end{bmatrix} = \begin{bmatrix} \frac{cos\,\psi\,(k)}{T_s} & \frac{sin\,\psi\,(k)}{T_s} \\ -\frac{1}{a}\frac{sin\,\psi\,(k)}{T_s} & \frac{1}{a}\frac{cos\,\psi\,(k)}{T_s} \end{bmatrix} \times \begin{bmatrix} x_{ref}(k+1) + l_x\,tanh\left(\frac{k_x}{l_x}e_x(k)\right) - x(k) \\ y_{ref}(k+1) + l_y\,tanh\left(\frac{k_y}{l_y}e_y(k)\right) - y(k) \end{bmatrix} \quad (5.46)$$

where $a > 0$, $[v_{ref}^c(k) \quad \omega_{ref}^c(k)]^T$ is the output of the kinematic controller, $e_x(k) = x_{ref}(k) - x(k)$, and $e_y(k) = y_{ref}(k) - y(k)$ are the position errors in the X and Y axis respectively, $k_x > 0$, $k_y > 0$ are the gains of the controller, $l_x, l_y \in \mathbb{R}$ are saturation constants. The $\tanh(\cdot)$ function is added to avoid a saturation of the control actions in the case of large position errors [162]. In the stability analysis, perfect velocity tracking is considered, $v_{ref}^c(k) \equiv v(k)$ and $\omega_{ref}^c(k) \equiv \omega(k)$. By replacing (5.46) in (5.45), the closed-loop equation is:

$$\begin{bmatrix} e_x(k+1) \\ e_y(k+1) \end{bmatrix} + \begin{bmatrix} l_x\,tanh\left(\frac{k_x}{l_x}e_x(k)\right) \\ l_y\,tanh\left(\frac{k_y}{l_y}e_y(k)\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.47)$$

Defining the output error vector as $\tilde{h}(k) = [e_x(k) \quad e_y(k)]^T$, then (5.47) can be written as:

$$\tilde{h}(k+1) = -\begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x} e_x(k)\right) \\ l_y \tanh\left(\frac{k_y}{l_y} e_y(k)\right) \end{bmatrix} \tag{5.48}$$

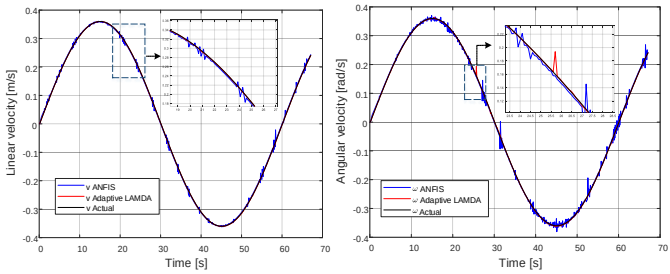
In [161] has been selected the Lyapunov's candidate function for the kinematic control law as $V(k) = \frac{1}{2}\tilde{h}^T(k)\tilde{h}(k)$. In the cited paper is demonstrated the stability of the kinematic controller for tracking trajectories if the parameters are set as $k_x > 0$, $k_y > 0$, $l_x > 0$ and, $l_y > 0$, then $\tilde{h}(k) \to 0$ for $k \to \infty$.

5.2.2.3.3 Dynamic Controller

The design of the dynamic controller is complex because a large number of parameters corresponding to the actuation mechanisms and physical variables of the robot must be considered in real-time. For this reason, non-adaptive controllers (Fuzzy-PI and LAMDA-PI) are not tested in this experiment since their calibration is complex and time-consuming, therefore, the adaptive methods are appropriate in this system. The following results are for ANFIS and Adaptive LAMDA since they are proposals that can learn about the dynamic of the system and do not require parametric calibration for the design.

In this case study, the benefits of the Adaptive LAMDA are clearly appreciated since the algorithm learns the dynamics of the system, which is considered as unknown and variable. The method is used to model it as follows: in the training

stage, LAMDA is applied to learn the inverse dynamic model based on the scheme of Figure 4.6a. The controller takes as input information the computed reference values in the output of the kinematic controller $[v_{ref}^{c} \quad \omega_{ref}^{c}]^{T}$ and the measured variables of the robot $[v \quad \omega]^{T}$. With this information, the identifier updates the internal parameters of the LAMDA model in the controller (obtaining the inverse model) in each sample time. For the training stage, the LAMDA Identifier 1 uses two inputs $[v(k+1); v(k)]$, each with two classes, $\eta_1 = 0.08$, $\beta_1 = 0.01$, $\lambda_1 = 0.999$. LAMDA Identifier 2 uses two inputs $[\omega(k+1); \omega(k)]$, each with two classes, $\eta_2 = 0.08$, $\beta_2 = 0.01$, $\lambda_2 = 0.999$. The sampling period of the simulation is $T_s = 0.1\, seg$. A sinusoidal input of 670 samples is generated for the system. Figure 5.35 shows the comparison of learning algorithms in the robot, which shows a better fit to the real values of linear and angular velocity of the adaptive LAMDA with respect to ANFIS.



(a) (b)

Figure 5.35. Comparison of learning algorithms in mobile robot (a) linear velocity, (b) angular velocity

In the application, the controller computes the output $\left[v_{ref}^{d}(k) \quad \omega_{ref}^{d}(k)\right]^{T}$ necessary to bring the system to the reference. The proposed structure, with an external

kinematic controller and an internal dynamic controller based on the adaptive methods (LAMDA or ANFIS), is the cascade scheme shown in the block diagram of Figure 5.36.

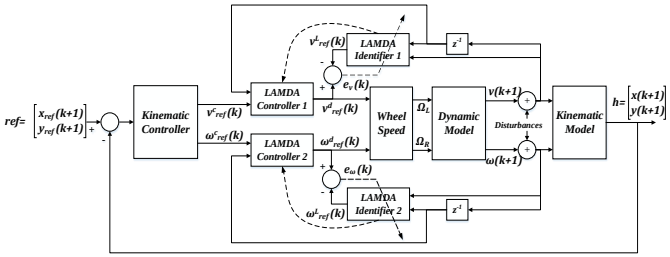


Figure 5.36. Adaptive control structure for a mobile robot

In the scheme, the online learning block (LAMDA Identifier 1) is placed between the control action computed by the dynamic controller $v_{ref}^{d}(k)$ and the measured variable v, to learn the inverse dynamics of the system using current and past information. The linear velocity v and its previous state are used as inputs for the identifier, in order to minimize the error $e_v(k) = v_{ref}^{d}(k) - v_{ref}^{L}(k)$, where $v_{ref}^{L}(k)$ is the output of the identifier. The minimization of e_v allows adjusting the parameters of the LAMDA model that are updated in the controller, at every sample time. The procedure described above is similarly applied to control the angular velocity ω, considering the minimization of the error $e_\omega(k) = \omega_{ref}^{d}(k) - \omega_{ref}^{L}(k)$, where $\omega_{ref}^{L}(k)$ is the output of the LAMDA Identifier 2. The control variables are the motor velocities, then it is necessary to compute the speed for the left and right wheels Ω_L and Ω_R, respectively, based on the values of $\begin{bmatrix} v_{ref}^{d}(k) & \omega_{ref}^{d}(k) \end{bmatrix}^T$. These relations are given by:

$$\Omega_L = \frac{2v_{ref}^d(k) - d\omega_{ref}^d(k)}{2r_1} \quad and$$

$$\Omega_R = \frac{2v_{ref}^d(k) + d\omega_{ref}^d(k)}{2r_1} \tag{5.49}$$

The proposed controllers are tested on a Pioneer 3DX robot [163] inside the Virtual Robot Experimentation Platform (V-REP). VREP allows simulating robotic systems considering their kinematics, dynamics and the physic of the environment [164]. The versatility of this software is linked to the availability of plug-ins to connect with other computational tools, such as Matlab, where the algorithms have been programmed. The main user interface of V-REP with the Pioneer 3DX robot is shown in Figure 5.37.

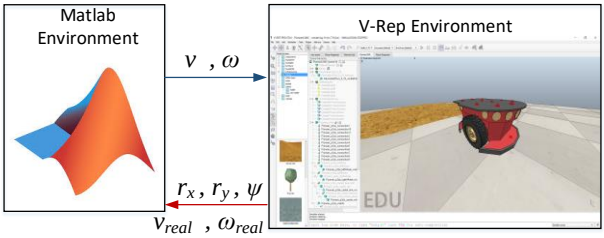


Figure 5.37. V-REP scene showing the Pioneer 3DX robot

The performance of the Adaptive LAMDA controller is tested against the ANFIS, such that the operating modes are similar to make a fair comparison. In this case study, the aim is to perform the trajectory control of the Pioneer 3DX applied in three different paths, applying a load to the robot (as a disturbance) to modify its dynamics and analyze the performance of the controllers. Graphical and numerical comparisons are performed to test the performance and

effectiveness of the algorithms in this control task (see IAE in Table 5.18).

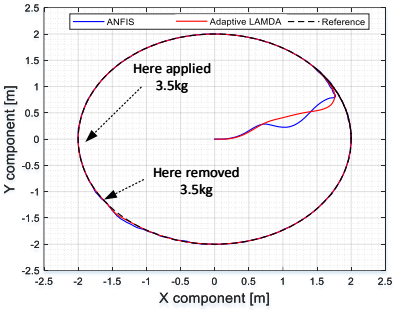
Three trajectories are tested: Circular (5.50), Lenmiscate curve (5.51) and Square (5.52). The starting point of the robot in all cases is in the coordinate $(x, y) = (0,0)m$.

$$\begin{cases} x_{ref}(k) = 2\,cos(0.033\pi k T_0) \\ y_{ref}(k) = 2\,sin(0.033\pi k T_0) \end{cases} \tag{5.50}$$

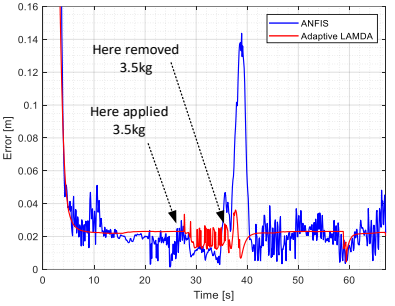
$$\begin{cases} x_{ref}(k) = 1.2sin(0.063\pi k T_0) \\ y_{ref}(k) = 2sin(0.0315\pi k T_0) \end{cases} \tag{5.51}$$

$$\begin{cases} x_{ref}(k) = 1.5\ \forall\ kT_o \in [0,15]\ ; (4.5 - 0.2kT_o)\ \forall\ kT_o \in [15,30]; \\ \quad -1.5\ \forall\ kT_o \in [30,45];\ (-10.5 + 0.2kT_o)\ \forall kT_o \in [45,60] \\ y_{ref}(k) = (-1.5 + 0.2kT_o)\ \forall\ kT_o \in [0,15]\ ; 1.5\ \forall\ kT_o \in [15,30]; \\ \quad (7.5 - 0.2kT_o)\ \forall\ kT_o \in [30,45];\ -1.5\ \forall\ kT_o \in [45,60] \end{cases} \tag{5.52}$$

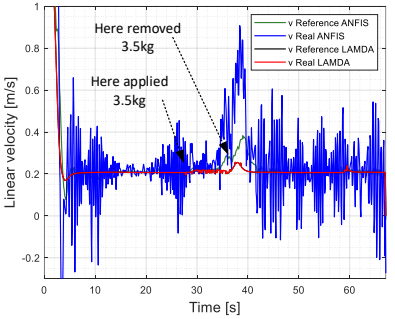
Figures 5.38, 5.39 and 5.40 show the comparative results for the Circular, Lenmiscate and Square trajectories, respectively, in which the response of the ANFIS and Adaptive LAMDA controllers are shown, as well as the position error in each trajectory. Additionally, the linear and angular speed references are shown, with the respective real values reached by the robot. During the simulation of this experiment, a $3.5kg$ load is added and removed on the robot at different time instants, affecting its dynamics to analyze the controllers' response.


ANFIS
Adaptive LAMDA
Reference
Here applied
3.5kg
Here removed
3.5kg
Y component [m]
X component [m]

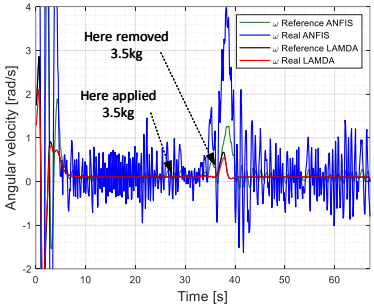
(a)


ANFIS
Adaptive LAMDA
Here removed
3.5kg
Here applied
3.5kg
Error [m]
Time [s]

(b)

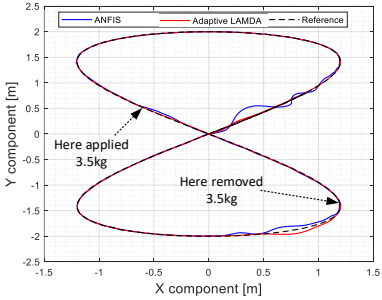

v Reference ANFIS
v Real ANFIS
v Reference LAMDA
v Real LAMDA
Here removed
3.5kg
Here applied
3.5kg
Linear velocity [m/s]
Time [s]

(c)

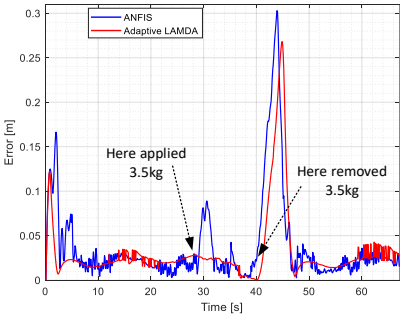


(d)

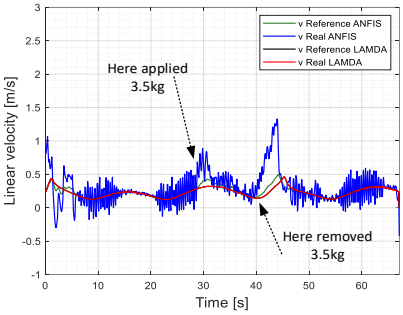
Figure 5.38. (a) Circular trajectory followed by the robot, (b) instantaneous quadratic error of the robot position, speeds of the robot and control actions (c) linear and (d) angular velocity



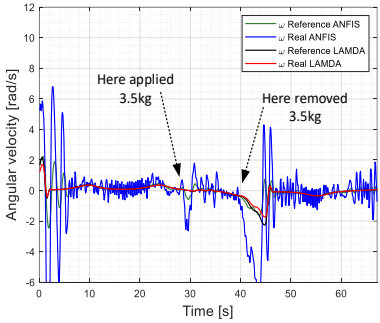
(a)



(b)

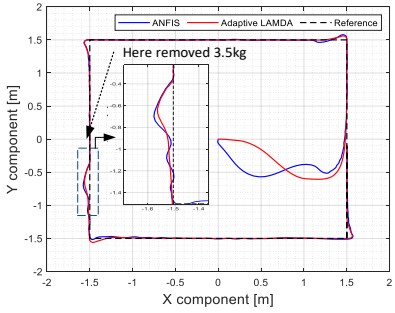


(c)

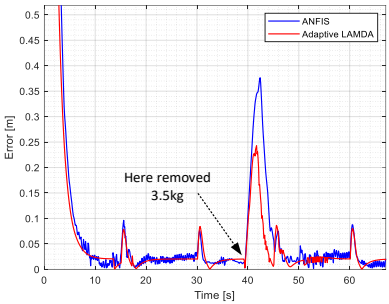


(d)

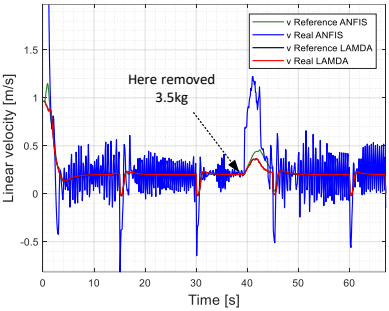
Figure 5.39. (a) Lenmiscate trajectory followed by the robot, (b) instantaneous quadratic error of the robot position, speeds of the robot and control actions (c) linear and (d) angular velocity



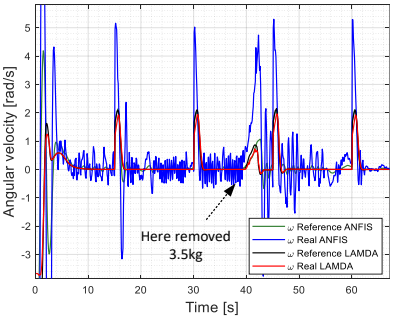
(a)



(b)



(c)



(d)

Figure 5.40. (a) Square trajectory followed by the mobile robot, (b) instantaneous quadratic error of the robot position, speeds of the robot and control actions (c) linear and (d) angular velocity

Figures 5.38a, 5.39a and 5.40a show the paths followed by the mobile robot. The results show qualitatively that the Adaptive LAMDA provides finer and more efficient control with respect to ANFIS (the two methods designed with the same learning parameters), getting a smaller distance error with respect to the references, and especially, under disturbances. The instantaneous quadratic error of the robot position controlled by the Adaptive LAMDA has an average of $2cm$ (see Figures 5.38b, 5.39b and 5.40b), a value considered acceptable taking into account that the dynamic of the controller is not based on the mathematical model of the system. In the case of the square, it is observed that the errors in the corners reach values of 9cm due to the abrupt changes in the robot's orientation, but as observed, they are quickly corrected by the LAMDA controller.

The linear speed in all the tested trajectories is around 0.2m/s (see Figures 5.38c, 5.39c and 5.40c), references reached by the Adaptive LAMDA controller with fast convergence, showing to be more efficient qualitatively than ANFIS due to the softer response and better in quantitative terms if it is analyzed the IAE values in Table 5.18. Then, it is clear that the learning parameters set for the linear speed controller are adequate because Adaptive LAMDA does not show oscillations in the control actions, while ANFIS presents a large number of oscillations, with amplitudes around ±0.4 m/s, which is excessive if compared to established references that can damage the actuators by the applied energy variations.

In the case of angular velocity, the tests show that the Adaptive LAMDA presents smooth control actions again.

The ANFIS proposals have oscillations around the reference of $\pm 5 rad/s$. This behavior can be seen in the Figures 5.38d, 5.39d and 5.40d, where it is observed that the proposal converges faster to the references than ANFIS. Also, the learning parameters set for the angular speed controller are adequate because the method does not show excessive oscillations in the control actions. LAMDA is much better than ANFIS because the control action is smooth, implying that the actuators are not abruptly actuated to reach the reference in steady-state. The reduction of the oscillations is considerable, which is one of the strong points to be highlighted by LAMDA. Finally, from the results obtained through this experiment, it has been possible to analyze the performance of the Adaptive LAMDA tracking controller applied to the dynamic model of the mobile robot, demonstrating its ability to follow the established speed references, and therefore, the desired trajectories. From the quantitative point of view, in this experiment, it is observed the benefits in performance terms of the Adaptive LAMDA with respect to the ANFIS controller, as can be observed in the results of IAE of Table 5.19 for all the paths. All these performance improvements are the result of two important factors, the use of aggregation operators in the GAD computation and the adjustment of the exigency parameter, which adapts to system variations online.

Table 5.19. Numerical values for IAE for the mobile robot experiments

IAE for trajectory tracking of a mobile robot			
Trajectory	**ANFIS**	**Adaptive LAMDA**	$\Delta\%$
Circle	5.725	5.038	12.77
Lenmiscate	2.495	2.101	17.15
Square	7.632	6.510	15.87

In all cases, the performance of LAMDA is better in a percentage greater than 12% over the ANFIS controller, when they were tested in the different trajectories. Under disturbances that affect the dynamic of the system, the proposed controller is the least affected and the one that converges more quickly towards the reference, which allows to validate the approach in fast dynamic robotic systems.

5.2.3 General comparative analysis among LAMDA controllers

Once the different proposals for LAMDA controllers have been tested, this subsection makes a comparative analysis of all the approaches addressed, to identify their behavior in the mobile robot case study. As detailed in subsection 5.2.2.3, for the tracking trajectory, a cascade control strategy composed of two controllers is used, the external controller based on a feedback linearization for the robot kinematics presented in (5.2.2.3.1), and the internal controller applied to the robot dynamics where the LAMDA approaches are tested. Two trajectories are examined for the evaluation of the methods, and the results of the experiments are compared qualitatively and quantitatively to determine which proposal performs the most accurate control and presents the best results in terms of performance.

5.2.3.1 Rule-based LAMDA applied to the dynamic model

Due to the large number of parameters involved in the dynamic model, such as physical variables and actuation mechanisms, its model identification and controller design is complex. A simple method to obtain an approximate model is through the reaction curve, in which an input signal is

applied to the system, and the behavior of the output is evaluated. The controllers are in charge of receiving the linear and angular value references and computing the necessary control actions to follow the desired trajectory.

Figure 5.41 shows the platform responses when a step signal is applied to the linear and angular speeds; the responses look like FOPDT models.

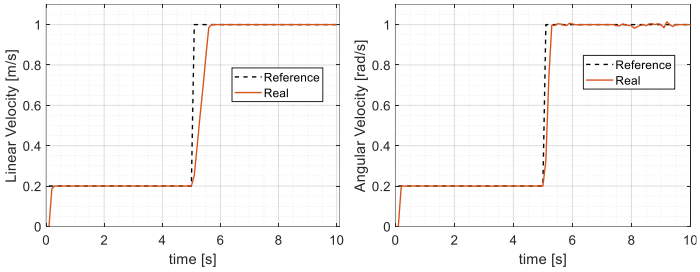


Figure 5.41. Step response (a) Linear and, (b) Angular velocity

The parameter identification performed in Figure 5.41 gives the following values for the linear velocity: $K_v = 1$, $\tau_v = 0.224\ sec.$, $t_{0v} = 0.144\ sec.$; and for the angular velocity: $K_\omega = 1$, $\tau_\omega = 0.116\ sec.$, $t_{0\omega} = 0.0856\ sec.$

With the knowledge of the approximate model's characteristic parameters, it is proposed the design of Rule-based LAMDA (LAMDA-PID) controllers. The inputs of the controllers are e_v, $\dot{e}_v$, e_w and $\dot{e}_w$, where e is the error obtained from the subtraction of the reference and the current system output and $\dot{e}$ is its derivative. These variables are selected to drive the system to the desired zero states, where the errors of linear and angular speeds and their derivatives are zero.

Centers of fuzzy classes C_k and their respective parameters in the consequent γ_k, are shown in Figure 5.42 to represent the analytic expression that summarizes the fuzzy logic inference presented in (4.1).

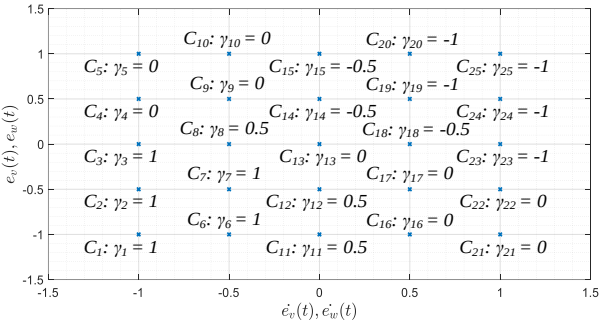


Figure 5.42. Defined classes and outputs for the linear and angular velocities

Considering that they are the training data for LAMDA operation, 25 classes are defined for each controller, setting the centers as a combination of the following sets:

$$e_v = [-1, -0.5, 0, 0.5, 1]\left[\frac{m}{s}\right]; \dot{e}_v = [-1, -0.5, 0, 0.5, 1]\left[\frac{m}{s^2}\right] \quad (5.53)$$

$$e_\omega = [-1, -0.5, 0, 0.5, 1]\left[\frac{rad}{s}\right]; \dot{e}_\omega = [-1, -0.5, 0, 0.5, 1]\left[\frac{rad}{s^2}\right] \quad (5.54)$$

The dynamic controller computes the output $\left[v_{ref}^d(t) \quad \omega_{ref}^d(t)\right]^T$ necessary to bring the system to the reference. The proposed structure, with an external kinematic controller and an internal dynamic controller based on the LAMDA-PID is shown in Figure 5.43. The

scheme shows blocks with the scaling gains kp_1, kd_1, ki_1, kp_2, kd_2 and ki_2 for tuning the response of the controllers.

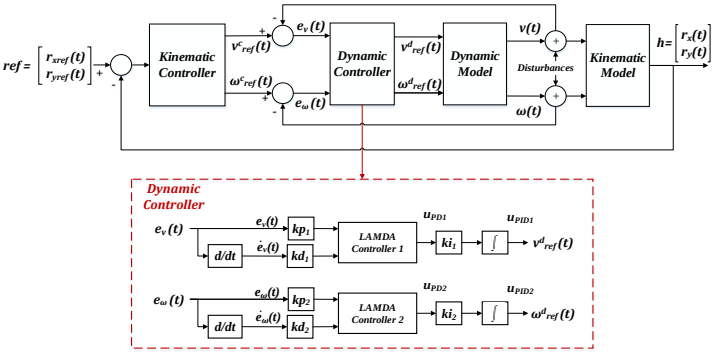


Figure 5.43. Control scheme for trajectory tracking of a mobile robot using LAMDA-PID

5.2.3.2 LSMC and ZLSMC applied to the dynamic model

The design of the LSMC linear and angular speed controllers is similar. Thus, to summarize the dynamic controller's design, only the procedure for the linear speed is detailed in this subsection. Consider the FOPTD for the linear velocity with the form of (5.7):

$$\frac{X_v(s)}{U_v(s)} = \frac{K_v e^{-t_{0v}s}}{\tau_v s + 1} \tag{5.55}$$

Modeling of dead time t_{0v} with a first-order Taylor series approximation [127]:

$$e^{-t_{0v}s} \cong \frac{1}{t_{0v}s + 1} \tag{5.56}$$

Substituting (5.56) into (5.55), it is obtained:

$$\frac{X_v(s)}{U_v(s)} \cong \frac{K_v}{\tau_v t_{0v} s^2 + (\tau_v + t_{0v})s + 1} \quad (5.57)$$

Solving (5.57) in the time domain:

$$\tau_v t_{0v} \ddot{x} + (\tau_v + t_{0v})\dot{x} + x - K_v u = 0 \quad (5.58)$$

The system represented in state-space, where $x_1 = x$, is:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{(\tau_v + t_{0v})}{\tau_v t_{0v}} x_2 - \frac{1}{\tau_v t_{0v}} x_1 + \frac{K_v}{\tau_v t_{0v}} u \end{aligned} \quad (5.59)$$

The form of expression (5.59) is similar to the presented in (4.5). So, the procedure for designing a stable LSMC controller (as detailed in subsection 4.2.1) is feasible.

As shown in (5.59), the system corresponds to a second-order model ($n = 2$). From (4.9), $s(t)$ becomes:

$$s(t) = \dot{e}_v(t) + 2\lambda e_v(t) + \lambda^2 \int e_v(t)\, dt \quad (5.60)$$

The derivative of (5.60) becomes:

$$\dot{s}(t) = \ddot{e}_v(t) + 2\lambda \dot{e}_v(t) + \lambda^2 e_v(t) = 0 \quad (5.61)$$

For $n = 2$ in (4.7):

$$\ddot{e}_v(t) = \dot{x}_{d2}(t) - \dot{x}_2(t) \quad (5.62)$$

Moreover, replacing (5.59) and (5.62) in (5.61):

$$\begin{aligned} \dot{s}(t) = \dot{x}_{d2}(t) + \frac{(\tau_v + t_{0v})}{\tau_v t_{0v}} x_2 + \frac{1}{\tau_v t_{0v}} x_1 - \frac{K_v}{\tau_v t_{0v}} u \\ + 2\lambda \dot{e}_v(t) + \lambda^2 e_v(t) = 0 \end{aligned} \quad (5.63)$$

Considering the values of the terms previously identified, $(K_v/\tau_v t_{0v}) > 0$, and based on (4.5), it is concluded that $b(X, t) > 0$. Thus, the rule tables presented in Tables 4.1 and 4.2 are used. Figure 5.44 shows the classes defined for

continuous control action. For example, if $\dot{s}(t) = 0.5$, then the Class 4 is activated, resulting in a control action $u_{ncv} = 0.5$ with the aim in order to satisfy $\dot{s}(t) = 0$.

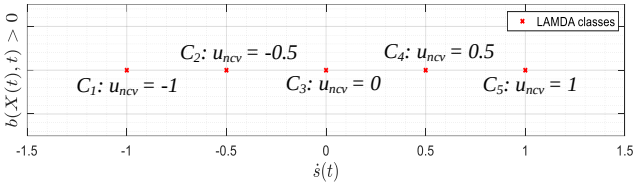


Figure 5.44. Classes and rules for continuous control action $\boldsymbol{u_c}$ based on $\boldsymbol{\dot{s}(t)}$ for the linear velocity of the mobile robot

Figure 5.45 shows the classes defined in the discontinuous control action. For example, if $\dot{s}(t) = -0.5$ and $s(t) = -1$, then the Class 6 is activated, resulting in a control action $u_{ncv} = -1$ in order to satisfy $s(t)\dot{s}(\mathrm{t}) < 0$.

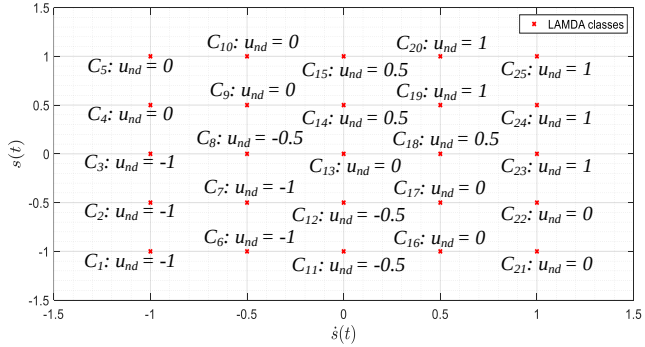


Figure 5.45. Classes and rules for discontinuous control action u_d based on $\dot{s}(t)$ and $s(t)$ for the linear velocity of the mobile robot

Finally, Figure 5.46 shows the block diagram of the proposed general controller. It corresponds to a cascade scheme with an external loop for the kinematic controller and

an internal loop with the dynamic controller composed of two independent controllers based on LSMC, one for linear speed and another for angular speed.

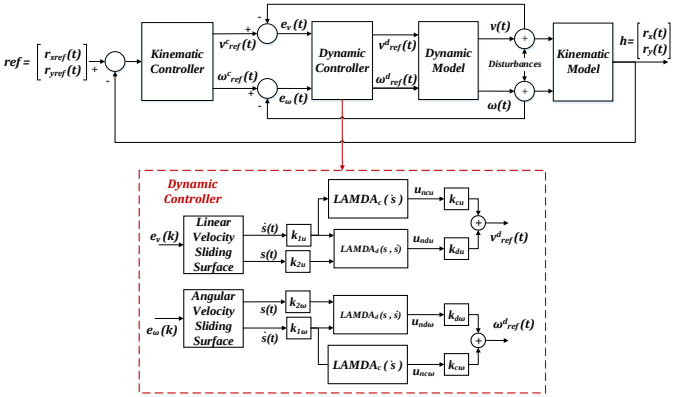


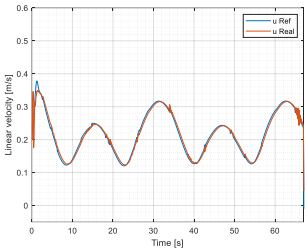
Figure 5.46. Adaptive LAMDA structure for a mobile robot

The ZLSMC has the similar design of LSMC with the reliability presented in Figures 4.9 and 4.10. The Adaptive LAMDA design is presented in section 5.2.2.3, for this reason, the design of these two methods used for the comparison is not detailed.

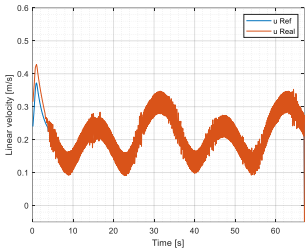
The controllers tested in this section applied to the dynamic model are: Fuzzy-PID, SMC, LAMDA-PID, Adaptive LAMDA, LSMC and ZLSMC. (In order to summarize information, the PID results are not shown, however those results are shown in [165]).

The tested trajectories are the Lenmiscate (5.51), and square (5.52), with the starting point of the robotic platform in the coordinate $(r_x, r_y) = (0m, 0m)$.

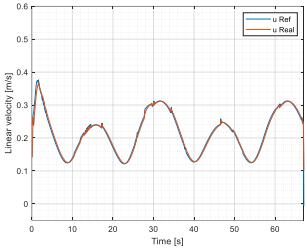
To perform the qualitative analysis, Figures 5.47-5.52 are presented. The response curves for the linear and angular velocities, trajectory error, and trajectory followed by the robotic platform controlled by each proposal mentioned above, are shown in these figures.



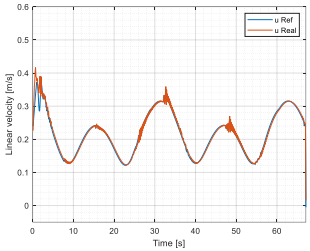
(a)



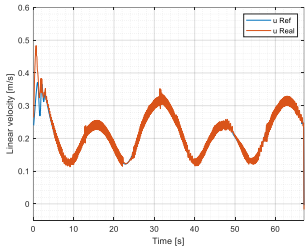
(b)



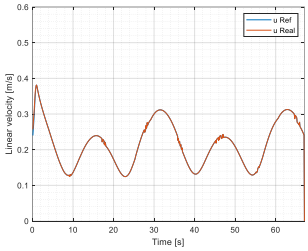
(c)



(d)



(e)



(f)

Figure 5.47. Linear velocity for the Lenmiscate trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA

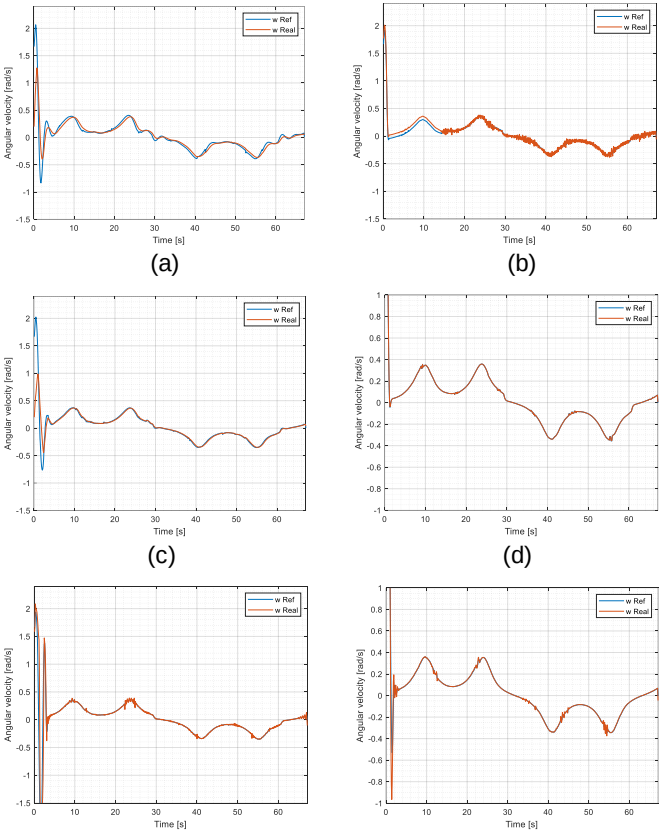
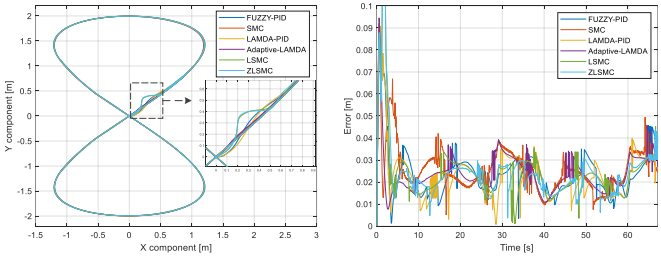
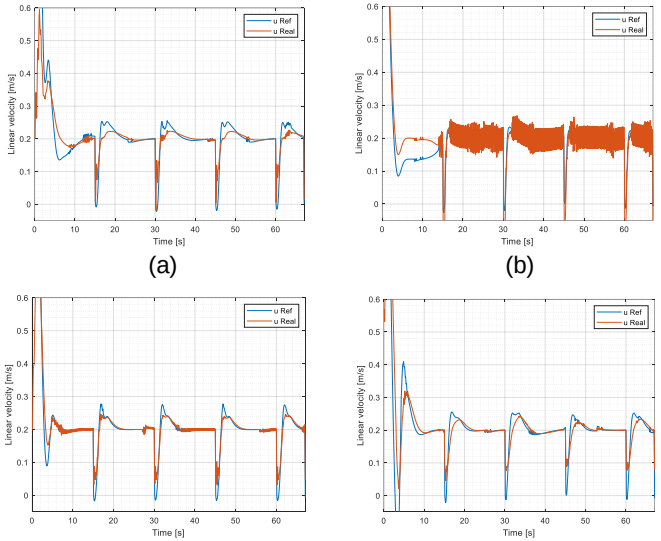


Figure 5.48. Angular velocity for the Lenmiscate trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA



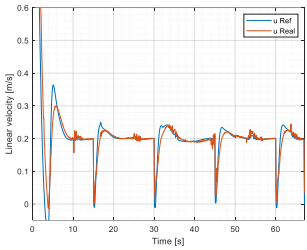
(a) (b)

Figure 5.49. (a) Lenmiscate trajectory performed by the different methods, (b) trajectory distance error

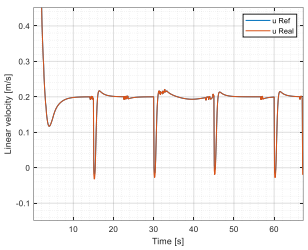


(a) (b)

(c) (d)

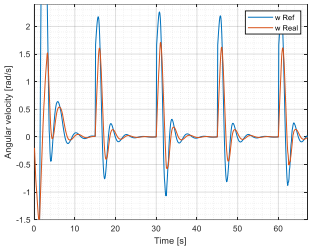


(e)

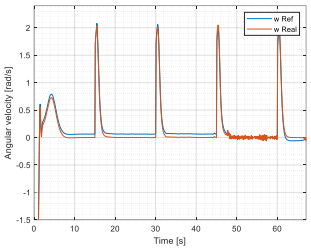


(f)

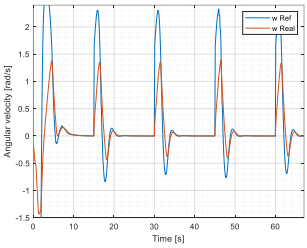
Figure 5.50. Linear velocity for the Square trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA



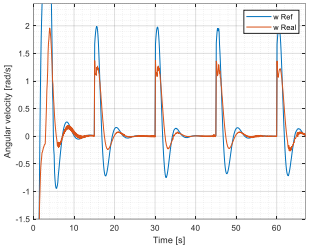
(a)



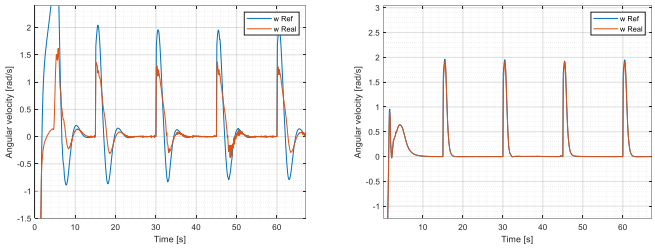
(b)



(c)

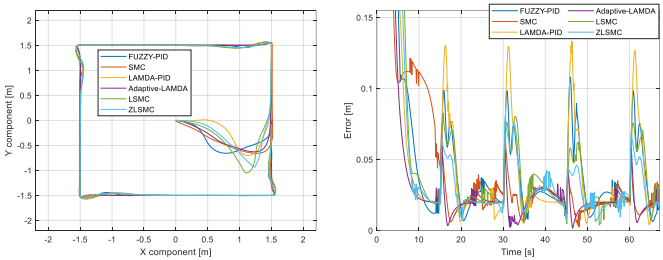


(d)



(e) (f)

Figure 5.51. Angular velocity for the Square trajectory: (a) Fuzzy-PID, (b) SMC, (c) LAMDA-PID, (d) LSMC, (e) ZLSMC, (f) Adaptive-LAMDA



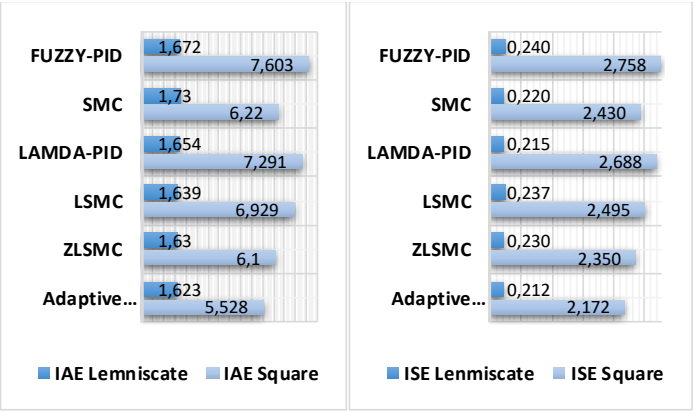
(a) (b)

Figure 5.52. (a) Lenmiscate trajectory performed by the different methods, (b) trajectory distance error

The results show that all the controllers perform the trajectory tracking. From a qualitatively point of view, it is observed the chattering presented in the controller action of SMC (see Figures 5.47b, 5.48b and 5.50b), which is the main drawback of this proposal being able to affect the actuators by the oscillations in a real platform, this effect decreases in the angular velocity of the square (see Figure 5.51b) because the robot follows straight lines.

In both trajectories, the Adaptive LAMDA is the best, reaching the reference in less time with a smooth control action with minimal oscillations in the linear velocity. In the square path, it is possible to observe that LSMC and ZLSMC tries to decrease the error quickly but to see the differences it is necessary to analyze the response quantitatively; however, as the reference points change, then the controller makes the robot change orientation and adjust to the reference smoothly with errors in the corners of the square smaller than the Fuzzy-PID, and LAMDA-PID. The error in most cases is around $2cm$, which corresponds to the 1% in terms of relative error considering the radius and the trajectories' side. In general terms, the Adaptive LAMDA proposal that requires a previous learning stage is the one that best controls the different trajectories. However, it requires more computational time than the other proposals due to the number of calculations required [136].

The quantitative analysis has been carried out based on the IAE, ISE, and ISCO indices. Figure 5.53 shows that the controllers based on LAMDA follows correctly the tested trajectories since ISE and IAE are the minima for the different methods. The Adaptive LAMDA is the best reaching in less time the references and decreasing the IAE and ISE. In the Lenmiscate trajectory, it is observed that LSMC performs an adequate control, being in second place in terms of IAE, and increasing its ISE value due to the initial oscillation that it presents. In the case of the square trajectory, LSMC quickly tries to reduce the position error; this the reason why the ISE is the second best after the Adaptive LAMDA.



(a) (b)

Figure 5.53. Quantitative comparison (a) IAE, (b) ISE

The control effort is measured with the ISCO metric which is computed as:

$$ISCO = \int_0^t u(t)dt \tag{5.64}$$

This parameter integrates the controller output square over time, penalizing large control actions more than smaller ones since the square of a large error will be much more significant. Therefore, the controller that obtains the minimum ISCO performs the best. This index values computed in the tracking trajectories are presented in Figure 5.54 for the linear and angular velocities control action.

The softer control actions are obtained for the controllers Fuzzy-PID, LAMDA-PID and Adaptive LAMDA in each velocity. The SMC controller increases the value of the index due to the chattering present in the control actions. Besides, it is observed that the LSMC controllers make a minimally

greater control effort in the Lenmiscate trajectory. However, in the square trajectory is observed that the control effort is less than the adaptive LAMDA. This characteristic is given by the calibration that has been given to the controller, in which it has been chosen to consider the ISE as the parameterization criterion. However, as shown in the results, the variations of ISCO in all the controllers is minimal, which allows to validate the proposal. It can also be noted that the control action of ZLSMC is a bit more abrupt (in linear speed), which is given by the calculation of the total utility that brings the system to reference more quickly at the cost of using more energy.

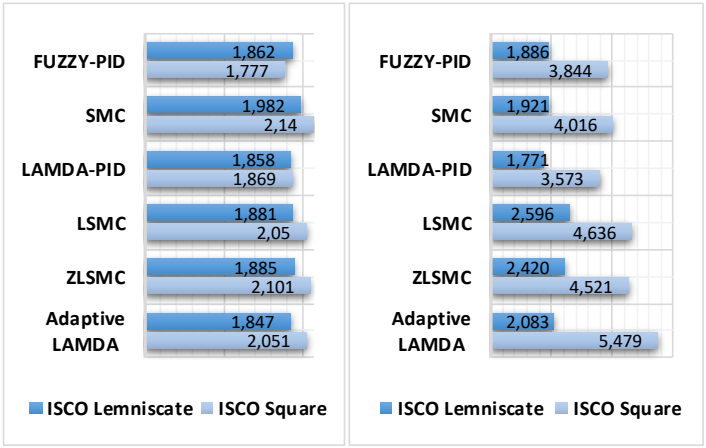


Figure 5.54. Quantitative comparison for the different controllers based on ISCO for the: a) Linear velocity, b) Angular velocity

In general, the best controller in terms of performance is the Adaptive-LAMDA, followed by LSMC and ZLSMC, but it is important to note that Adaptive LAMDA requires a previous learning stage. On the other hand, LSMC is a chattering-free

robust method that does not require the learning stage, which reduces its computational time, but it requires an adequate calibration to obtain competitive results as in the case of LAMDA-PID.

5.3 Computational Complexity

5.3.1 Memory Usage

In this subsection is presented a summary of the results published in the papers [136,138] where LSMC and Adaptive LAMDA are presented with a complete analysis. No emphasis has been placed on analyzing the Rule-based LAMDA since it is considered within the LSMC proposal. The computational complexity of the LAMDA controllers is evaluated in terms of memory usage, computation time and number of operations [166]. The programs are implemented in Matlab R2020a, running on an Intel (R) Core (TM) i7-8750H @ 2.2GHz processor. The computational complexity is computed based on the number parameters of the algorithms, these are l: the number of descriptors (inputs), c: the number of classes in each descriptor.

- Memory usage of LSMC

The number of parameters to compute the controller output $u = u_c + u_d$ is based on the total number of classes $m = c^l$. To compute u_c, LSMC requires only the descriptor $\dot{s}$, that is $l = 1$; hence, the number of parameters to be stored is $2c$. To compute u_d, LSMC requires the descriptors s and $\dot{s}$, that is $l = 2$; hence, the number of parameters to be stored is $2c^2$. Aditionally, the parameters $\sigma_{k,j} = 0.25$, α, λ, and four scaling gains are required. The total number of parameters to be stored in memory $(\#stored\_values)$ is:

$$\#stored\_values = 2c^2 + 2c + 7 \quad (5.64)$$

In terms of Big O the algorithm has the function $f(c)$:

$$f(c) = O(c^2) \quad \forall c \geq 1 \quad (5.65)$$

- Memory usage of Adaptive LAMDA:

The number of parameters to be computed by the algorithm in the learning (antecedent and the consequent) is based on the number of inputs and number of classes in each input, l and c, respectively. The number of parameters in the antecedents $\#parameters_{\theta(k)}$ and $\#parameters_{\theta(k-1)}$ are:

$$\#parameters_{\theta(k)} = 2lc + 1 \quad (5.66)$$

$$\#parameters_{\theta(k-1)} = 2lc + 1 \quad (5.67)$$

The number of parameters of the consequent in the vector $h(k)$ is $\#parameters_{h(k)}$, and the number of parameters of the covariance matrix is $\#parameters_{P(k)}$. These values are:

$$\#parameters_{P(k)} = [m(l+1)]^2 \quad (5.68)$$

$$\#parameters_{h(k)} = m(l+1) \quad (5.69)$$

Each value is stored in 2 bytes of memory [166].

Finally, the total number of stored parameters analyzing from (5.66) - (5.69), gives as result the Big O function $f(ml)$:

$$f(ml) = O(m^2 l^2) \quad \forall m \geq 1 \text{ and } \forall l \geq 1 \quad (5.70)$$

5.3.2 Computation Time

The temporal complexity verifies the time required to compute the control output in each sample time. Tables 5.20 and 5.21 shows the sample average time of a total of 2000 samples, in which the control output has been calculated for a different number of classes in each descriptor "c."

Table 5.20. Computation Time (seconds) of LSMC ($l = 2$)

	Number of classes "c" in each descriptor			
	3	**5**	**7**	**9**
u_c	*8.24-6*	*8.88e-6*	*8.98e-6*	*9.51e-6*
u_d	*8.71e-5*	*1.08e-4*	*1.13e-4*	*1.36e-4*
u	*9.53e-5*	*1.17e-4*	*1.22e-4*	*1.46e-4*

Table 5.21. Computation Time in seconds (s) of Adaptive LAMDA

	Number of classes "c" in each descriptor							
	2		**3**		**4**		**5**	
"l"	Learning	Control	Learning	Control	Learning	Control	Learning	Control
2	*1.40 e-4*	*4.08 e-5*	*1.69 e-4*	*4.48 e-5*	*4.90 e-4*	*5.13 e-5*	*8.82 e-4*	*5.65 e-5*
3	*2.08 e-4*	*5.24 e-5*	*1.10 e-3*	*6.43 e-5*	*1.71 e-3*	*1.18 e-4*	*10.7 e-3 s*	*1.09 e-3*
4	*7.56 e-4*	*5.67 e-5*	*6.64 e-3*	*7.85 e-4*	*93.5 e-3*	*7.84 e-3*	*1.02 e1*	*46.5 e-3*
5	*1.87 e-3*	*9.07 e-5*	*162e -3*	*11.6 e-3*	*852e -3*	*182 e-3*	*NT*	*NT*
6	*6.70 e-3*	*8.74 e-4*	*624e -3*	*134. 9*	*NT*	*NT*	*NT*	*NT*
7	*45.1 e-3*	*4.70 e-3*	*NT*	*NT*	*NT*	*NT*	*NT*	*NT*
8	*343e -3*	*25.8 e-3*	*NT*	*NT*	*NT*	*NT*	*NT*	*NT*

*NT corresponds to not tested.

5.3.3 Number of Operations

The temporal complexity depends on the type of processor and memory characteristics in which the program is executed, for this reason, it is most appropriate to evaluate the number of arithmetic operations (arithmetic complexity) used to solve a problem. Subtraction, addition,

multiplication, division, squared and exponent are considered as basic operations.

The computational complexity of LAMDA controllers, compared to proposals such as conventional Fuzzy (number of discretization of output universe of discourse suggested $(M_{OD}) = 32$ [166]), LAMDA without learning [154], ANFIS, LAMDA and Adaptive LAMDA are presented in Table 5.22, and the asymptotic notation is presented in Table 5.23.

Table 5.22. Arithmetic complexity of the fuzzy control algorithms

	Arithmetic complexity
Conventional Fuzzy [166]	$4089m + 37ml + 31lc + 59l + 21$
Rule-based LAMDA [154]	$m^2 + 6ml + 3m + 5lc - 1$
Adaptive ANFIS [72]	$m^3(4l^3 + 12l^2 + 12l + 5)$ $+m^2(6l^2 + 12l + 10) + 8lc + 46$
LSMC [138]	$30c^2 + 30c + 47$
Adaptive LAMDA [136]	$m^3(4l^3 + 12l^2 + 12l + 5)$ $+m^2(6l^2 + 12l + 9) + m(11l + 2) + 8lc + 48$

Table 5.23. Arithmetic complexity in terms of Asymptotic Notation

	Arithmetic complexity
Conventional Fuzzy [166]	$f(ml) = O(ml) \quad \forall m \geq 1$ and $\forall l \geq 1$
Rule-based LAMDA [154]	$f(m) = O(m^2) \quad \forall m \geq 1$
Adaptive ANFIS [72]	$f(ml) = O(m^3 l^3) \quad \forall m \geq 1$ and $\forall l \geq 1$
LSMC [138]	$f(c) = O(c^2) \quad \forall c \geq 1$
Adaptive LAMDA [136]	$f(ml) = O(m^3 l^3) \quad \forall m \geq 1$ and $\forall l \geq 1$

It is evident that the adaptive proposals are the most complex in computational terms, which is logical due to the

learning algorithm that they incorporate, these being of cubic order with respect to the number of classes. Adaptive LAMDA is similar to ANFIS when analyzing the number of operations that these require for the learning and the computation of the control output, the difference lies in the linear term m, so it cannot be considered a relevant difference. As it has been observed in the results of Table 1, when working with a low number of classes in each descriptor, the algorithm is more efficient and requires less computation time. Therefore, it can be applied in systems with 8 descriptors at a speed of 0.4 seconds in the worst case, which shows the viability in the use of the adaptive controller.

In LSMC, a quadratic exponent in the term of a number of classes in each descriptor is observed. This shows the simplicity, in computational terms, of the algorithm when calculating the control action. The LSMC can be applied in systems with 9 classes at a speed of 0.15 milliseconds, which shows the viability in the application of the controller.

6. DISCUSSION

6.1 Classification Context

LAMDA-HAD is able to work on classification problems improving the performance of LAMDA. In most of the used datasets, LAMDA-HAD obtains results as good as the best classifiers like RF or LDA. LAMDA-HAD has the advantage of creating new classes outside of the training stage being widely superior to LAMDA, since it corrects the errors identified in the original algorithm.

The tests have been performed on datasets with balanced and unbalanced classes, measuring metrics according to these characteristics (Accuracy and F-measure). In scenarios with symmetric classes, the accuracy of LAMDA-HAD is very similar to the best classifiers, and in cases of asymmetry, our algorithm is better when the descriptors are overlapped, their data distribution is similar, or where there are a large number of outliers.

Based on Tables 5.2, 5.3 and 5.4 which summarize the results of the LAMDA-HAD, the performance of this algorithm is comparable to the performance of the classifiers that perform better. They show that LAMDA-HAD has a high performance in tasks of supervised learning improving the performance in relation to the original LAMDA in all benchmarks.

In unbalanced cases, LAMDA-HAD allows detecting the characteristic of asymmetry between the classes. In this way, the proposed algorithm identifies correctly the classes of small number of individuals without assigning its objects to other classes. This can be corrected by balancing the

classes, as has been made in the Wholesale Costumers dataset as demonstrated in [105]. With this correction, the accuracy of LAMDA-HAD is similar to the best classifiers, keeping the values of F-measure where it had the best performance.

LAMDA-HAD has the advantage, compared to LAMDA, of assigning not identified individuals to the NIC more accurately, option that the other analyzed algorithms do not have (see Figure 5.2). The feature of creating and adapting classes of LAMDA is preserved. When HAD parameter is computed, there is more certainty in the assignment when there is no significant difference between the existing classes and the new data that is being entered. This is achieved because it is being compared between the GAD values of the individual and the GAD of each class to validate the classification. Therefore, the possibility of modifying the initial model by temporary changes that could occur is diminished. By adapting the GAD to each class, a better description of the data representing the class is obtained, avoiding that an "outlier" or "temporary change" data drastically modify the general model described by the set of classes.

Finally, LAMDA-HAD presents greater robustness for the classification because HAD decreases the possibility of creating new classes by mistake. However, in an online process many very different data arrive that can modify the number of classes, requiring that the model can adapt to changes in the process.

6.2 Clustering Context

LAMDA-RD in all cases considerably improves the performance of the original algorithm (see the results from Table 5.6 to Table 5.9).

Based on the results of P_C, metric that considers SC and WB-index, it can be noticed that in 4 of the 10 datasets tested (Dim1024, Hepta, Unbalance and s1), LAMDA-RD obtains the best results, in terms of performance, while in Segment (high dimensionality), R15 and Aggregation, is close to the best algorithms. In Postures (benchmark with a large number of samples and high dimensionality) is the best algorithm when compared to other clustering methods focused on data streams, achieving the objective of this work of obtaining a competitive algorithm that in all cases improves the performance of LAMDA.

LAMDA-RD has been tested in balanced and unbalanced datasets with different overlapping. In cases where there is no overlap, the algorithm works as well as KM, KMD and AHT. With an overlap of 9% detailed in [150], as in the case of s1, LAMDA-RD presents the best results, while with an overlap of less than 20% (s2 and a1), the algorithm has an intermediate performance. Also, it is noted that the performance decays in s3, which has a 40% overlap (strong non-Gaussian distribution of feature values), where it is complicated to make an online assignment of elements working in a streaming data scenario, based on distances and densities.

In the context of the data stream scenario, LAMDA-RD, based on the performance metrics of Table 5.10, is widely superior to LAMDA-TP and ADDClustering, since the

proposal has a merging process that allows avoiding the creation of an excessive number of clusters.

LAMDA-RD reaches the best results comparing with other well-known clustering algorithms in benchmarks with overlap <20% (see in [150]) and unbalanced datasets, such as: Dim1024, Segment, Unbalance and s1. While increasing the individuals in the overlap area for instance s2, s3, and a1, the algorithm decreases its performance since it is based on density measurements. LAMDA-RD can work in online mode with streaming data, however, it is not adequate when the dataset has a large number of individuals, because the algorithm's execution time will considerably increase. Aditionally, LAMDA-RD can discover new groups with a low computational cost. The addition of a merging algorithm avoids the creation of an excessive number of poor quality clusters, which is demonstrated by the performance metrics SC, WB_{index} and P_C.

The values d_{nb} and D_t regulate the requirements in the clusters to be merged. Specifically, setting D_t to a value close to zero, then a merging process is made between the nearby clusters, but with dissimilar characteristics. In the other hand, setting D_t to a value close to 1, then the merge is made between nearby and similar clusters.

The robust distance (RD) related to d_{nb} allows improving the quality of the resulting clusters, since this term penalizes the dissimilarity between the individuals and the clusters. These parameters affect the quality of clusters related to the minimum P_C and the final number of clusters created m.

6.3 Control Systems Context

LSMC can be applied in a class of SISO systems presented in (4.5). For its implementation, it is required to know an estimation of the order of the system, which is useful for the design of the sliding surface. The proposal, as shown, is capable of controlling the non-linear systems presented in the simulations, in which the dynamics can be considered as partially known due to the changes that its parameters may present as a function of the physical variables involved in the model (as in the mixing tank), without the need for recalibration since the controller does not degrade, which has allowed to validate its robustness.

The application of LSMC may be feasible in systems with MIMO characteristics (similar in structure to the SISO presented in this work), initially it would be appropriate to consider its application in decoupled MIMO systems in which it is possible to work with independent controllers in each of the variables. For this, there are mathematical tools for decoupling MIMO systems, such as those based on the relative gain array RGA [13]. This method proposes to compute Bristol's matrix used to measure the interaction between the inputs and outputs in a multivariate process control. Based on this information, if necessary, the design of a decoupler that allows controlling the n-variables of the system with n-controllers would make the implementation of the LSMC method feasible and practical.

The most important limitation in terms of applicability is the parameterization and calibration of the scaling gains of the LSMC, which must be analyzed with great attention and meticulousness in various systems to establish initial

equations that allow to have a starting point to obtain these parameters. The formalization of these equations would be of great help and importance in the design stage since in this work, as mentioned above, it has been heuristically calibrated with the ISE minimization process, which it might not be feasible in real processes due to the risk that this implies in case of having an incorrect initial calibration.

Calibration using PSO increases the calibration time since the algorithm must be run n-iterations to minimize the ISE. Similar results have been found when doing heuristic minimization as shown in Figures 5.9-5.11, 5.17-5.18, which is a process that can take less time. For this reason, the formalization of the Adaptive LAMDA algorithm has been proposed, capable of self-adjusting its internal parameters.

Z-numbers based controller produces a less impulsive response, reaching the reference more quickly due to the fact that in the design stage the term "U" of reliability is considered when errors are large. As seen in Figure 5.22, the control action of ZLSMC is smoother than the LSMC, this causes the overshoot to be minimal, which is a considerable improvement in terms of power consumption.

From a qualitative point of view, it has been observed that the Adaptive LAMDA is capable to control systems better than the other proposals in all case studies. In the mixing tank with variable dead time system, it is observed that the algorithm is capable of adapting to changes in the dynamics of the system produced by the variation of the dead time, calculating a less aggressive control action that is capable to take the system to the reference in less time. In the case of the HVAC system, the two variables T_3 and W_3 are

adequately regulated, even when moisture and temperature disturbances are added to the system; the control action of Adaptive LAMDA allows to regulate the system, considering that the algorithm learns directly from the behavior of the plant and has not required calibration, which is indispensable in the non-adaptive methods. Also, it has been observed that the response of the method is better, especially with respect to ANFIS. In the robot tracking trajectory, it is evident that Adaptive LAMDA is much better than ANFIS, it is enough to observe that the control action of the LAMDA proposal is less abrupt and oscillatory, which is an advantage since the actuators may not respond to the control action computed by ANFIS.

The simulations of Adaptive LAMDA have shown that the proposed method is able to control systems even better than other intelligent methods. In the case of the mixing tank, the results have shown that the proposal is better in terms of performance by 60% over non-adaptive methods (LAMDA-PI and Fuzzy-PI), and by 170% over the more similar approach (ANFIS), which maintains steady-state error without reaching the reference (see Table 5.17). In the case of regulation of the HVAC system, it has been possible to observe an excellent performance of Adaptive LAMDA in the control stage, in the presence of temperature and moisture disturbances, considerably improving performance with respect to ANFIS, as shown in the related results with the IAE (see Table 5.16). The proposal presents a smoother and less oscillatory control action that eliminates the steady-state error. On the other hand, in the application of trajectory tracking of the mobile robot, it has been observed that LAMDA achieves the control objective with excellent

performance, but the most important characteristic to note is the shape of the control actions produced by the proposed method, in which it is clearly observed that the oscillations decrease considerably with respect to ANFIS, with errors in the trajectory of smaller magnitude, that is, better IAE (see Table 5.18). The three case studies have shown that the main advantage of the learning-based controllers is that they do not depend on the mathematical model of the plants, which are often complex to obtain and may have modeling errors.

7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The main goal of the thesis has been reached: the formalization of a controller based on the LAMDA fuzzy model. For this purpose, extensions to improve LAMDA in classification and clustering tasks have been proposed.

In the case of classification tasks, LAMDA-HAD has been tested in several benchmarks and its results have been compared with other classifiers like LDA, RF, SVM, DT, among others, showing very similar results for the F-measure and Accuracy metrics, with respect to the best classifiers, being in our experiments LDA and RF. In all tests, LAMDA-HAD is much better than the original LAMDA. Likewise, it has shown that it is the best algorithm when when the classes are well defined (their descriptors correctly characterize each class), with the advantage that LAMDA-HAD does not hide information of unbalanced classes (reducing the performance). LAMDA-HAD is the only algorithm that for different data characteristics (unbalanced classes, overlapping, etc) obtain competitive results.

In the case of clustering tasks, an automatic merge algorithm, called LAMDA-RD, has been formalized. This algorithm analyses the similarity between neighboring clusters to decide if the merge process is carried out or not. It has an additional execution time; however, this problem compensated with the ability of the algorithm to avoid creating an excessive number of clusters. This feature is not possible in LAMDA-TP and in LAMDA. In the comparative study with them, LAMDA-RD significantly improves the performance in terms of the metrics SC, WB_{index} and RI.

LAMDA has the characteristic of making intrinsically a split process, generating new classes when it does not identify the similarity between the individual and the clusters. However, the quality of the clusters generated is not similar to the iterative methods, especially in cases of high overlapping, which LAMDA-RD corrects presenting better results in all the benchmarks. In cases when the overlapping is 0-20%, LAMDA-RD presents results comparable as iterative methods (KM, KMD and FCM), and as the overlapping increases its performance decreases because it is more complex to make an assignment when the elements have characteristics of several clusters in online methods. The advantage of LAMDA-RD is that it can discover new groups with a low computational cost. The addition of a merging algorithm avoids the creation of an excessive number of poor quality clusters.

In the context of control systems, a controller based on LAMDA has been formalized, adding an inference stage in order to take the system to the desired state through the calculation of the GADs. By combining LAMDA with the SMC theory, it has been shown that LSMC is stable and is capable of controlling systems with variable dynamics and model uncertainties.

The design of LSMC is simple and requires only two descriptors: $s(t)$ and its derivative $\dot{s}(t)$. That information is sufficient for the computation of continuous and discontinuous control actions of a sliding mode approach.

In a CSTR process, the results show that LSMC presents a smoother response than the PI, SMC, and LAMDA-PID at the reference changes. In terms of disturbance rejection, it

has been verified that LSMC is robust enough to reach the reference in a lower time than the conventional SMC without an abrupt control action (without damaging the actuator).

In the mixing tank with variable dynamics, LSMC achieves a better performance against the other tested controllers. Also, we observe that LSMC regulates the process in short time, being stable without degrading its performance as the PID does. Additionally, the controller reaches the reference in the presence of the disturbances, which is not achieved by the conventional SMC.

The LSMC outperforms the results of LAMDA-PID, improving the ISE, with a smoother response that reaches the reference in less time, and reducing considerably the oscillations. Also, LSMC is more robust when there are disturbances due to its design based on the SMC theory, which considers the Lyapunov stability concepts. It is observed that LSMC is a chattering-free scheme, solving a problem present in traditional SMC schemes, and avoids the use of the sign function that causes this phenomenon.

ZLSMC uses the criteria of restriction given for the MADs of LAMDA and the reliability obtained of the sliding surface to compute a more aggressive control action in presence of large errors and smooth control action when the error is close to zero. The Total Utility improves the performance of LSMC allowing to reach the reference quickly and smoothly (reducing the overshoot), with control actions that would not affect the actuators in a real system since one of its strengths is being robust and chattering-free.

Adaptive LAMDA has the capability to control systems without the need to know its exact mathematical model. The proposed method can be implemented on any system in

which its inverse model can be identified, and offers a great advantage over non-adaptive methods as LSMC, Rule-based LAMDA, and conventional controllers.

Adaptive LAMDA requires a learning stage with a higher computational cost than to the other methods due to the optimization algorithms implemented for the self-adjustment of parameters required for learning the inverse dynamics of the plant to be controlled. The Adaptive LAMDA is better with respect to ANFIS as shown in all the case studies presenting a less aggressive response, properly following the desired reference or trajectory which leads to eliminate the error in steady state. In the robot tracking trajectory, it is evident that Adaptive LAMDA is much better than ANFIS, the control action of our proposal is less abrupt and oscillatory which is an advantage, since the actuators of the robot may not respond to the oscillatory control action computed by ANFIS.

7.2 Future Work

As future work, it is proposed to improve the performance of LAMDA-RD when it is tested in strong non-Gaussian distribution of feature values, and address in detail the curse of dimensionality in datasets with a very high number of features. Also, it is necessary to combine the clustering algorithm with supervised learning features to implement a hybrid algorithm (semi-supervised learning) based on LAMDA, which can be applied in systems with labeled and unlabeled data.

For LAMDA-RD is required to improve the algorithm performance through the computation of the suitable threshold for each cluster in the merge process, and to formalize the parameter calibration of the algorithm.

In the context of control, LAMDA approaches are competitive and better than classic fuzzy, PID, SMC, and ANFIS which leads to propose the validation of LAMDA control in other more complex systems, with dynamics more challenging to model, as in the case of aerial robots where the loads are critical. Also, it is necessary to propose a parameter calibration method to facilitate the tuning of the scaling gains of the LSMC controller.

The Adaptive LAMDA is stable as experiments show, both in the learning and in the operation stage, but theoretically, it must be demonstrated its global stability properties.

Finally, it is proposed to extend ZLSMC to Adaptive LAMDA approaches, in which the centers of the classes for the restriction and reliability of the Z-numbers can be automatically computed in online learning mode, in order to avoid the heuristic calibration that is a time-consuming and complex process in some systems with uncertain dynamics.

8. REFERENCES

[1] M. Santos, Un Enfoque Aplicado del Control Inteligente, Rev. Iberoam. Automática e Informática Ind. RIAI. 8 (2011) 283–296. https://doi.org/10.1016/j.riai.2011.09.016.

[2] P. Antsaklis, Defining intelligent control, IEEE Control Syst. Mag. 4 (1994) 58–66. http://www.nd.edu/~pantsakl/2010upload/162-RTFIC.pdf.

[3] P. Antsaklis, ntelligen t L le ming Control, 15 (1995) 5–7.

[4] R. Galán, A. Jiménez, R. Sanz, F. Martía, Control Inteligente, Intel. Artif. 4 (2000) 43–48. https://doi.org/10.4114/ia.v4i10.661.

[5] O. Granichin, Z. Volkovich, D. Toledano-Kitai, Randomized Algorithms in Automatic Control and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. https://doi.org/10.1007/978-3-642-54786-7.

[6] K. Words--adaptive, Adaptive Control Based on Explicit Criterion Minimization *, 21 (1985) 385–399. https://doi.org/10.1016/0005.

[7] N. Noroozi, M. Roopaei, M.Z. Jahromi, Adaptive fuzzy sliding mode control scheme for uncertain systems, Commun. Nonlinear Sci. Numer. Simul. 14 (2009) 3978–3992. https://doi.org/10.1016/j.cnsns.2009.02.015.

[8] S. Soyguder, H. Alli, An expert system for the humidity and temperature control in HVAC systems using ANFIS and optimization with Fuzzy Modeling Approach, Energy Build. 41 (2009) 814–822. https://doi.org/10.1016/j.enbuild.2009.03.003.

[9] T. Waegeman, F. Wyffels, B. Schrauwen, Feedback Control by Online Learning an Inverse Model, IEEE Trans. Neural Networks Learn. Syst. 23 (2012) 1637–1648. https://doi.org/10.1109/TNNLS.2012.2208655.

[10] P.. Fleming, R.. Purshouse, Evolutionary algorithms in control systems engineering: a survey, Control Eng. Pract. 10 (2002) 1223–1241. https://doi.org/10.1016/S0967-0661(02)00081-3.

[11] J. Aguilar-Martín, R. López De Mantaras, The process of

classification and learning the meaning of linguistic descriptors of concepts, in: Approx. Reason. Decis. Anal., North-Holland Publishing Company, 1982: pp. 165–175.

[12] J. Waissman, R. Sarrate, T. Escobet, J. Aguilar, B. Dahhou, Wastewater treatment process supervision by means of a fuzzy automaton model, in: Proc. 2000 IEEE Int. Symp. Intell. Control., IEEE, 2002: pp. 163–168. https://doi.org/10.1109/ISIC.2000.882917.

[13] J.F. Botía Valderrama, D.J.L. Botía Valderrama, On LAMDA clustering method based on typicality degree and intuitionistic fuzzy sets, Expert Syst. Appl. 107 (2018) 196–221. https://doi.org/10.1016/j.eswa.2018.04.022.

[14] J.R. Jang, ANFIS : Adap tive-Ne twork-Based Fuzzy Inference System, 23 (1993).

[15] M. Sayed Mouchaweh, Semi-supervised classification method for dynamic applications, Fuzzy Sets Syst. 161 (2010) 544–563. https://doi.org/10.1016/j.fss.2009.11.002.

[16] J.A. Morente-Molinera, J. Mezei, C. Carlsson, E. Herrera-Viedma, Improving Supervised Learning Classification Methods Using Multigranular Linguistic Modeling and Fuzzy Entropy, IEEE Trans. Fuzzy Syst. 25 (2017) 1078–1089. https://doi.org/10.1109/TFUZZ.2016.2594275.

[17] M. Cerrada, J. Aguilar, J. Altamiranda, R.-V. Sánchez, A hybrid heuristic algorithm for evolving models in simultaneous scenarios of classification and clustering, Knowl. Inf. Syst. 61 (2019) 755–798. https://doi.org/10.1007/s10115-019-01336-3.

[18] X.Y. Qiu, P. Srinivasan, Y. Hu, Supervised learning models to predict firm performance with annual reports: An empirical study, J. Assoc. Inf. Sci. Technol. 65 (2014) 400–413. https://doi.org/10.1002/asi.22983.

[19] Y. Song, W. Cai, H. Huang, Y. Zhou, D.D. Feng, Y. Wang, M.J. Fulham, M. Chen, Large Margin Local Estimate With Applications to Medical Image Classification, IEEE Trans. Med. Imaging. 34 (2015) 1362–1377. https://doi.org/10.1109/TMI.2015.2393954.

[20] Q.P. He, Jin Wang, Large-Scale Semiconductor Process

Fault Detection Using a Fast Pattern Recognition-Based Method, IEEE Trans. Semicond. Manuf. 23 (2010) 194–200. https://doi.org/10.1109/TSM.2010.2041289.

[21] J.F. Botía, C. Isaza, T. Kempowsky, M. V. Le Lann, J. Aguilar-Martín, Automaton based on fuzzy clustering methods for monitoring industrial processes, Eng. Appl. Artif. Intell. 26 (2013) 1211–1220. https://doi.org/10.1016/j.engappai.2012.11.003.

[22] L.F. Zhu, J.S. Wang, H.Y. Wang, A Novel Clustering Validity Function of FCM Clustering Algorithm, IEEE Access. 7 (2019) 152289–152315. https://doi.org/10.1109/ACCESS.2019.2946599.

[23] D.G. Ferrari, L.N. de Castro, Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods, Inf. Sci. (Ny). 301 (2015) 181–194. https://doi.org/10.1016/j.ins.2014.12.044.

[24] P. Berkhin, A Survey of Clustering Data Mining Techniques, in: Group. Multidimens. Data, Springer-Verlag, Berlin/Heidelberg, 2012: pp. 25–71. https://doi.org/10.1007/3-540-28349-8_2.

[25] A. Amini, T.Y. Wah, H. Saboohi, On Density-Based Data Streams Clustering Algorithms: A Survey, J. Comput. Sci. Technol. 29 (2014) 116–141. https://doi.org/10.1007/s11390-013-1416-3.

[26] A. Ben Ayed, M. Ben Halima, A. Alimi, Survey on clustering methods: Towards fuzzy clustering for big data, in: 2014 6th Int. Conf. Soft Comput. Pattern Recognit., IEEE, 2014: pp. 331–336. https://doi.org/10.1109/SOCPAR.2014.7008028.

[27] H. Parvin, B. Minaei-Bidgoli, A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm, Pattern Anal. Appl. 18 (2014) 87–112. https://doi.org/10.1007/s10044-013-0364-4.

[28] A. Bagherinia, B. Minaei-Bidgoli, M. Hossinzadeh, H. Parvin, Elite fuzzy clustering ensemble based on clustering

diversity and quality measures, Appl. Intell. 49 (2019) 1724–1747. https://doi.org/10.1007/s10489-018-1332-x.

[29] H. Alizadeh, B. Minaei-Bidgoli, H. Parvin, Optimizing fuzzy cluster ensemble in string representation, Int. J. Pattern Recognit. Artif. Intell. 27 (2013). https://doi.org/10.1142/S0218001413500055.

[30] M.-S. Yang, C.-Y. Lai, C.-Y. Lin, A robust EM clustering algorithm for Gaussian mixture models, Pattern Recognit. 45 (2012) 3950–3961. https://doi.org/10.1016/j.patcog.2012.04.031.

[31] M.S. Yang, S.J. Chang-Chien, Y. Nataliani, Unsupervised fuzzy model-based Gaussian clustering, Inf. Sci. (Ny). 481 (2019) 1–23. https://doi.org/10.1016/j.ins.2018.12.059.

[32] C. Aggarwal, C.K. Reddy, Data clustering: algorithms and applications, CRC press, 2013.

[33] J. Sui, Z. Liu, A. Jung, L. Liu, X. Li, Dynamic clustering scheme for evolving data streams based on improved STRAP, IEEE Access. 6 (2018) 46157–46166. https://doi.org/10.1109/ACCESS.2018.2864553.

[34] J. Alcala-Fdez, R. Alcala, S. Gonzalez, Y. Nojima, S. Garcia, Evolutionary Fuzzy Rule-Based Methods for Monotonic Classification, IEEE Trans. Fuzzy Syst. 25 (2017) 1376–1390. https://doi.org/10.1109/TFUZZ.2017.2718491.

[35] M. Cerrada, J. Aguilar, Fuzzy Classifier System and Genetic Programming on System Identification Problems, Proc. 3rd Annu. Conf. Genet. Evol. Comput. (2001) 1245–1251. https://dl.acm.org/citation.cfm?id=2955459.

[36] M. Elkano, M. Galar, J. Sanz, H. Bustince, CHI-BD: A fuzzy rule-based classification system for Big Data classification problems, Fuzzy Sets Syst. 348 (2018) 75–101. https://doi.org/10.1016/j.fss.2017.07.003.

[37] C. Isaza, J. Aguilar-Martin, M.V. Le Lann, J. Aguilar, A. Rios-Bolivar, An Optimization Method for the Data Space Partition Obtained by Classification Techniques for the Monitoring of Dynamic Processes, Artif. Intell. Res. Dev. 146 (2006) 80–87.

http://www.booksonline.iospress.nl/Content/View.aspx?piid=2087.

[38] T. Kempowsky, Surveillance de procédés à base de méthodes de classification : conception d'un outil d'aide pour la détection et le diagnostic des défaillances, (2004). https://tel.archives-ouvertes.fr/tel-00010247.

[39] H.-X. Li, Y. Wang, G. Zhang, Probabilistic Fuzzy Classification for Stochastic Data, IEEE Trans. Fuzzy Syst. 25 (2017) 1391–1402. https://doi.org/10.1109/TFUZZ.2017.2687402.

[40] A. Mesiarová-Zemánková, K. Ahmad, Averaging operators in fuzzy classification systems, Fuzzy Sets Syst. 270 (2015) 53–73. https://doi.org/10.1016/j.fss.2014.06.010.

[41] J. Nayak, B. Naik, H.S. Behera, Fuzzy C-Means (FCM) Clustering Algorithm: A Decade Review from 2000 to 2014, in: H.S. Behera, D.P. Mohapatra (Eds.), Comput. Intell. Data Min. - Vol. 2, Springer India, New Delhi, 2015: pp. 133–149. https://doi.org/10.1007/978-81-322-2208-8_14.

[42] M. Gong, L. Su, M. Jia, W. Chen, Fuzzy Clustering With a Modified MRF Energy Function for Change Detection in Synthetic Aperture Radar Images, IEEE Trans. Fuzzy Syst. 22 (2014) 98–109. https://doi.org/10.1109/TFUZZ.2013.2249072.

[43] S.A. Sert, H. Bagci, A. Yazici, MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks, Appl. Soft Comput. 30 (2015) 151–165. https://doi.org/10.1016/j.asoc.2014.11.063.

[44] H. Izakian, W. Pedrycz, I. Jamal, Fuzzy clustering of time series data using dynamic time warping distance, Eng. Appl. Artif. Intell. 39 (2015) 235–244. https://doi.org/10.1016/j.engappai.2014.12.015.

[45] C. Bedoya, J. Waissman Villanova, C.V. Isaza Narvaez, Yager–Rybalov Triple Π Operator as a Means of Reducing the Number of Generated Clusters in Unsupervised Anuran Vocalization Recognition, in: 2014: pp. 382–391. https://doi.org/10.1007/978-3-319-13650-9_34.

[46] T. Kempowsky, A. Subias, J. Aguilar-Martin, Process

situation assessment: From a fuzzy partition to a finite state machine, Eng. Appl. Artif. Intell. 19 (2006) 461–477. https://doi.org/10.1016/j.engappai.2005.12.012.

[47] I. Arroyo, Evaluación de dos técnicas de reconocimiento de patrones para su implementación en el simulador de pilotaje automatico (PA-135, NM-79 chopper) de taller del STC metro de la CD. de méxico, (2013). http://jupiter.utm.mx/~tesis_dig/12105.pdf.

[48] J. Mora-Florez, V. Barrera-Nunez, G. Carrillo-Caicedo, Fault Location in Power Distribution Systems Using a Learning Algorithm for Multivariable Data Analysis, IEEE Trans. Power Deliv. 22 (2007) 1715–1721. https://doi.org/10.1109/TPWRD.2006.883021.

[49] B. Lamrini, M.-V. Le Lann, A. Benhammou, E.K. Lakhal, Detection of functional states by the 'LAMDA' classification technique: application to a coagulation process in drinking water treatment, Comptes Rendus Phys. 6 (2005) 1161–1168. https://doi.org/10.1016/j.crhy.2005.11.017.

[50] H.R. Hernandez, J.L. Camas, A. Medina, M. Perez, M. Veronique Le Lann, Fault Diagnosis by LAMDA methodology Applied to Drinking Water Plant, IEEE Lat. Am. Trans. 12 (2014) 985–990. https://doi.org/10.1109/TLA.2014.6893990.

[51] M. Ruiz, J. Colomer, M. Rubio, J. Meléndez, Combination of multivariate statistical process control and classification tool for situation assessment applied to a sequencing batch reacto wastewater treatment, in: VIII Int. Work. Intell. Stat. Qual. Control. Print. House Zakl. Poligr. Warszawa, 2004: pp. 1–9. http://upcommons.upc.edu/handle/2117/15353.

[52] J. Aguilar-Martin, C. Isaza, E. Diez-Lledo, M.V. LeLann, J.W. Vilanova, Process Monitoring Using Residuals and Fuzzy Classification with Learning Capabilities, in: Theor. Adv. Appl. Fuzzy Log. Soft Comput., Springer Berlin Heidelberg, Berlin, Heidelberg, 2007: pp. 275–284. https://doi.org/10.1007/978-3-540-72434-6_28.

[53] V. Krivanek, Application LAMDA algorithm for Fault Detection and Isolation, in: 14th Int. Conf. Mechatronika, IEEE, 2011: pp. 46–51.

https://doi.org/10.1109/MECHATRON.2011.5961069.

[54] J.G. Zambrano, E. Guzmán-Ramirez, O. Pogrebnyak, Search Algorithms for Engineering Optimization, InTech, 2013. https://doi.org/10.5772/45841.

[55] E. Guzman, J.G. Zambrano, A. Orantes, O. Pogrebnyak, A theoretical exposition to apply the lamda methodology to vector quantization, in: 2009 52nd IEEE Int. Midwest Symp. Circuits Syst., IEEE, 2009: pp. 743–746. https://doi.org/10.1109/MWSCAS.2009.5235988.

[56] J.M.T. Garfias, J.L. Flores, A.O. Molina, M.C.J.L.B. Avalos, A New Tool for Merging the Information Based on Clustering Methods, in: 2011 IEEE Electron. Robot. Automot. Mech. Conf., IEEE, 2011: pp. 155–160. https://doi.org/10.1109/CERMA.2011.86.

[57] J.C. Atine, A. Doncescu, J. Aguilar-Martin, A fuzzy clustering approach for supervision of biological processes by image processing, Proc. 4th Conf. Eur. Soc. Fuzzy Log. Technol. EUSFLAT. (2005) 1057–1063.

[58] A. Doncescu, J. Aguilar-Martin, J.-C. Atine, Image color segmentation using the fuzzy tree algorithm T-LAMDA, Fuzzy Sets Syst. 158 (2007) 230–238. https://doi.org/10.1016/j.fss.2006.10.007.

[59] J.F. Botía, A. Cardenas, G. Quintero, GVD and SPM optimization in optical fiber using a hybrid method, in: 2012 IEEE Colomb. Commun. Conf., IEEE, 2012: pp. 1–6. https://doi.org/10.1109/ColComCon.2012.6233668.

[60] F. Ruiz, C. Isaza, A. Agudelo, J. Agudelo, A new criterion to validate and improve the classification process of LAMDA algorithm applied to diesel engines, Eng. Appl. Artif. Intell. 60 (2017) 117–127. https://doi.org/10.1016/j.engappai.2017.02.005.

[61] A. Doncescu, S. Regis, N. Kabbaj, Reinforced Operators in Fuzzy Clustering Systems, in: F. Xhafa, L. Barolli, P.J. Papajorgji (Eds.), Springer New York, New York, NY, 2010: pp. 247–266. https://doi.org/10.1007/978-1-4419-1636-5_12.

[62] C. Bedoya, C. Uribe, C. Isaza, Unsupervised Feature

Selection Based on Fuzzy Clustering for Fault Detection of the Tennessee Eastman Process, in: Lect. Notes Comput. Sci., Springer, Berlin, Heidelberg, 2012: pp. 350–360. https://doi.org/10.1007/978-3-642-34654-5_36.

[63] T. Kempowsky, J. Aguilar, A. Subias, M.-V. Le Lann, Classification Tool Based on Interactivity Between Expertise and Self-Learning Techniques 1, IFAC Proc. Vol. 36 (2003) 675–680. https://doi.org/10.1016/S1474-6670(17)36570-9.

[64] M. Mizumoto, Pictorial representations of fuzzy connectives, Part I: Cases of t-norms, t-conorms and averaging operators, Fuzzy Sets Syst. 31 (1989) 217–242. https://doi.org/10.1016/0165-0114(89)90005-5.

[65] J. Mora-Florez, V. Barrera-Nuez, G. Carrillo-Caicedo, Fault Location in Power Distribution Systems Using a Learning Algorithm for Multivariable Data Analysis, IEEE Trans. Power Deliv. 22 (2007) 1715–1721. https://doi.org/10.1109/TPWRD.2006.883021.

[66] Q. Yang, S. Jagannathan, Reinforcement Learning Controller Design for Affine Nonlinear Discrete-Time Systems using Online Approximators, IEEE Trans. Syst. Man, Cybern. Part B. 42 (2012) 377–390. https://doi.org/10.1109/TSMCB.2011.2166384.

[67] D. Romeres, M. Zorzi, R. Camoriano, S. Traversaro, A. Chiuso, Derivative-Free Online Learning of Inverse Dynamics Models, IEEE Trans. Control Syst. Technol. (2019) 1–15. https://doi.org/10.1109/TCST.2019.2891222.

[68] J. Vieira, F.M. Dias, A. Mota, Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study, Eng. Appl. Artif. Intell. 17 (2004) 265–273. https://doi.org/10.1016/j.engappai.2004.03.001.

[69] J. Si, Y.T. Wang, On-line learning control by association and reinforcement, IEEE Trans. Neural Networks. 12 (2001) 264–276. https://doi.org/10.1109/72.914523.

[70] M. Hagan, H. Demuth, O. De Jesús, An introduction to the use of neural networks in control systems, Int. J. Robust

Nonlinear Control. 12 (2002) 959–985. https://doi.org/10.1002/rnc.727.

[71] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man. Cybern. SMC-15 (1985) 116–132. https://doi.org/10.1109/TSMC.1985.6313399.

[72] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man. Cybern. 23 (1993) 665–685. https://doi.org/10.1109/21.256541.

[73] M.A. Denaï, F. Palis, A. Zeghbib, Modeling and control of non-linear systems using soft computing techniques, Appl. Soft Comput. 7 (2007) 728–738. https://doi.org/10.1016/j.asoc.2005.12.005.

[74] J.S.R. Jang, Chuen-Tsai Sun, Neuro-fuzzy modeling and control, Proc. IEEE. 83 (1995) 378–406. https://doi.org/10.1109/5.364486.

[75] H. Arpaci, O.F. Özgüven, ANFIS & PIλ Dμ controller design and comparison for overhead cranes, Indian J. Eng. Mater. Sci. 18 (2011) 191–203.

[76] S.R. Khuntia, S. Panda, ANFIS approach for SSSC controller design for the improvement of transient stability performance, Math. Comput. Model. 57 (2013) 289–300. https://doi.org/10.1016/j.mcm.2011.06.052.

[77] M.A. Denai, F. Palis, A. Zeghbib, ANFIS based modelling and control of non-linear systems: a tutorial, in: 2004 IEEE Int. Conf. Syst. Man Cybern., IEEE, 2004: pp. 3433–3438. https://doi.org/10.1109/ICSMC.2004.1400873.

[78] A. Niasar, A. Vahedi, H. Moghbelli, ANFIS-Based Controller with Fuzzy Supervisory Learning for Speed Control of 4-Switch Inverter Brushless DC Motor Drive, in: 37th IEEE Power Electron. Spec. Conf., IEEE, 2006: pp. 1–5. https://doi.org/10.1109/PESC.2006.1711957.

[79] F. Baghli, Y. Lakhal, L. El Bakkali, O. Hamdoun, Design and simulation of Adaptive Neuro Fuzzy Inference System (ANFIS) controller for a robot manipulator, in: 2014 Second World Conf. Complex Syst., IEEE, 2014: pp. 298–303. https://doi.org/10.1109/ICoCS.2014.7060879.

[80] M. Shafiq, M.A. Shafiq, H.A. Yousef, Stability and Convergence Analysis of Direct Adaptive Inverse Control, Complexity. 2017 (2017) 1–12. https://doi.org/10.1155/2017/7834358.

[81] R.E. Precup, M.L. Tomescu, Stable fuzzy logic control of a general class of chaotic systems, Neural Comput. Appl. 26 (2015) 541–550. https://doi.org/10.1007/s00521-014-1644-7.

[82] R.-E. Precup, H. Hellendoorn, A survey on industrial applications of fuzzy control, Comput. Ind. 62 (2011) 213–226. https://doi.org/10.1016/j.compind.2010.10.001.

[83] X.F. Yuan, Y.N. Wang, L.H. Wu, Adaptive inverse control of excitation system with actuator uncertainty, Neural Process. Lett. 27 (2008) 125–136. https://doi.org/10.1007/s11063-007-9064-7.

[84] Jia Li, Yu Jinshou, Nonlinear hybrid adaptive inverse control using neural fuzzy system and its application to CSTR systems, in: Proc. 4th World Congr. Intell. Control Autom. (Cat. No.02EX527), IEEE, 2002: pp. 1896–1900. https://doi.org/10.1109/WCICA.2002.1021413.

[85] M. Turki, S. Bouzaida, A. Sakly, F. M'Sahli, Adaptive control of nonlinear system using neuro-fuzzy learning by PSO algorithm, in: 2012 16th IEEE Mediterr. Electrotech. Conf., IEEE, 2012: pp. 519–523. https://doi.org/10.1109/MELCON.2012.6196486.

[86] Y. Chen, G. Mei, G. Ma, S. Lin, J. Gao, Robust adaptive inverse dynamics control for uncertain robot manipulator, Int. J. Innov. Comput. Inf. Control. 10 (2014) 575–587.

[87] S. Ravi, M. Sudha, P.A. Balakrishnan, Design of Intelligent Self-Tuning GA ANFIS Temperature Controller for Plastic Extrusion System, Model. Simul. Eng. 2011 (2011) 1–8. https://doi.org/10.1155/2011/101437.

[88] Z. Du, X. Li, Q. Mao, A new online hybrid learning algorithm of adaptive neural fuzzy inference system for fault prediction, Int. J. Model. Identif. Control. 23 (2015) 68. https://doi.org/10.1504/IJMIC.2015.067716.

[89] C. Pramod, M. Tomar, G. Pillai, A Modified Extreme

Learning ANFIS for Higher Dimensional Regression Problems, in: Comput. Intell. Theor. Appl. Futur. Dir. I, Springer Singapore, 2019: pp. 279–292. https://doi.org/10.1007/978-981-13-1135-2_22.

[90] L.A. Zadeh, A Note on Z-numbers, Inf. Sci. (Ny). 181 (2011) 2923–2932. https://doi.org/10.1016/j.ins.2011.02.022.

[91] R.H. Abiyev, N. Akkaya, I. Gunsel, Control of omnidirectional robot using z-number-based fuzzy system, IEEE Trans. Syst. Man, Cybern. Syst. 49 (2019) 238–252. https://doi.org/10.1109/TSMC.2018.2834728.

[92] R.A. Aliev, W. Pedrycz, O.H. Huseynov, S.Z. Eyupoglu, Approximate Reasoning on a Basis of Z-Number Valued If–Then Rules, IEEE Trans. Fuzzy Syst. 25 (2017) 1589–1600. https://doi.org/10.1109/TFUZZ.2016.2612303.

[93] B. Kang, D. Wei, Y. Li, Y. Deng, A Method of Converting Z-number to Classical Fuzzy Number, J. Inf. Comput. Sci. 9 (2012) 703–709.

[94] R.A. Aliev, A.V. Alizadeh, O.H. Huseynov, The arithmetic of discrete Z-numbers, Inf. Sci. (Ny). 290 (2015) 134–155. https://doi.org/10.1016/j.ins.2014.08.024.

[95] R.A. Aliev, L.M. Zeinalova, Decision making under Z-information, Stud. Comput. Intell. 502 (2014) 233–252. https://doi.org/10.1007/978-3-642-39307-5_10.

[96] B. Kang, D. Wei, Y. Li, Y. Deng, Decision making using Z-numbers under uncertain environment, J. Comput. Inf. Syst. 8 (2012) 2807–2814.

[97] H. gang Peng, J. qiang Wang, Outranking Decision-Making Method with Z-Number Cognitive Information, Cognit. Comput. 10 (2018) 752–768. https://doi.org/10.1007/s12559-018-9556-y.

[98] Z. Tao, X. Liu, H. Chen, J. Liu, F. Guan, Linguistic Z-number fuzzy soft sets and its application on multiple attribute group decision making problems, Int. J. Intell. Syst. 35 (2020) 105–124. https://doi.org/10.1002/int.22202.

[99] W. Jiang, C. Xie, B. Wei, Y. Tang, Failure mode and effects

analysis based on Z-numbers, Intell. Autom. Soft Comput. 24 (2018) 1–8. https://doi.org/10.1080/10798587.2017.1327158.

[100] B. Kang, Y. Deng, R. Sadiq, Total utility of Z-number, Appl. Intell. 48 (2018) 703–729. https://doi.org/10.1007/s10489-017-1001-5.

[101] R.H. Abiyev, I. Günsel, N. Akkaya, Z-number Based Fuzzy System for Control of Omnidirectional Robot, in: Intell. Technol. Robot., Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-35249-3_60.

[102] R.H. Abiyev, Z Number Based Fuzzy Inference System for Dynamic Plant Control, Adv. Fuzzy Syst. 2016 (2016). https://doi.org/10.1155/2016/8950582.

[103] M. Abdelwahab, V. Parque, A.M.R. Fath Elbab, A.A. Abouelsoud, S. Sugano, Trajectory tracking of wheeled mobile robots using Z-Number based fuzzy logic, IEEE Access. 8 (2020) 18426–18441. https://doi.org/10.1109/ACCESS.2020.2968421.

[104] L.A. Zadeh, A Note on Z-numbers, Inf. Sci. (Ny). 181 (2011) 2923–2932. https://doi.org/10.1016/j.ins.2011.02.022.

[105] L. Morales, J. Aguilar, D. Chávez, C. Isaza, LAMDA-HAD, an Extension to the LAMDA Classifier in the Context of Supervised Learning, Int. J. Inf. Technol. Decis. Mak. 19 (2020) 283–316. https://doi.org/10.1142/S0219622019500457.

[106] L. Morales, C.A. Ouedraogo, J. Aguilar, C. Chassot, S. Medjiah, K. Drira, Experimental comparison of the diagnostic capabilities of classification and clustering algorithms for the QoS management in an autonomic IoT platform, Serv. Oriented Comput. Appl. 13 (2019) 199–219. https://doi.org/10.1007/s11761-019-00266-w.

[107] L. Morales, J. Aguilar, An Automatic Merge Technique to Improve the Clustering Quality Performed by LAMDA, IEEE Access. 8 (2020) 162917–162944. https://doi.org/10.1109/ACCESS.2020.3021675.

[108] L. Morales, H. Lozada, J. Aguilar, E. Camargo, Applicability

of LAMDA as classification model in the oil production, Artif. Intell. Rev. 53 (2020) 2207–2236. https://doi.org/10.1007/s10462-019-09731-6.

[109] M. Lichman, UCI Machine Learning Repository . Irvine, CA: University of California, School of Information and Computer Science., (2013). http://archive.ics.uci.edu/ml.

[110] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, IEEE Trans. Fuzzy Syst. 1 (1993) 98–110. https://doi.org/10.1109/91.227387.

[111] T. Niknam, M.H. Khooban, Fuzzy sliding mode control scheme for a class of non-linear uncertain chaotic systems, IET Sci. Meas. Technol. 7 (2013) 249–255. https://doi.org/10.1049/iet-smt.2013.0039.

[112] Y.-J. Niu, X.-Y. Wang, A novel adaptive fuzzy sliding-mode controller for uncertain chaotic systems, Nonlinear Dyn. 73 (2013) 1201–1209. https://doi.org/10.1007/s11071-012-0444-9.

[113] I. Lagrat, H. Ouakka, I. Boumhidi, B.P. Atlas, Fuzzy Sliding Mode PI Controller for Nonlinear Systems, WSEAS Trans. Signal Process. 2 (2006) 1137–1143.

[114] J. Yi, N. Yubazaki, Stabilization fuzzy control of inverted pendulum systems, Artif. Intell. Eng. 14 (2000) 153–163. https://doi.org/10.1016/S0954-1810(00)00007-8.

[115] B.M. Al-Hadithi, A. Jiménez, F. Matía, A new approach to fuzzy estimation of Takagi–Sugeno model and its applications to optimal control for nonlinear systems, Appl. Soft Comput. 12 (2012) 280–290. https://doi.org/10.1016/j.asoc.2011.08.044.

[116] C. Knospe, PID control, IEEE Control Syst. 26 (2006) 30–31. https://doi.org/10.1109/MCS.2006.1580151.

[117] G. Feng, A Survey on Analysis and Design of Model-Based Fuzzy Control Systems, IEEE Trans. Fuzzy Syst. 14 (2006) 676–697. https://doi.org/10.1109/TFUZZ.2006.883415.

[118] L. Teng, Y. Wang, W. Cai, H. Li, Robust model predictive control of discrete nonlinear systems with time delays and disturbances via T – S fuzzy approach, J. Process Control. 53 (2017) 70–79.

https://doi.org/10.1016/j.jprocont.2016.11.012.

[119] A. Bayas, I. Škrjanc, D. Sáez, Design of fuzzy robust control strategies for a distributed solar collector field, Appl. Soft Comput. 71 (2018) 1009–1019. https://doi.org/10.1016/j.asoc.2017.10.003.

[120] Y. Ren, X. Duan, H. Li, C.L. Philip Chen, Multi-variable fuzzy logic control for a class of distributed parameter systems, J. Process Control. 23 (2013) 351–358. https://doi.org/10.1016/j.jprocont.2012.12.004.

[121] G.-C. Hwang, S.-C. Lin, A stability approach to fuzzy control design for nonlinear systems, Fuzzy Sets Syst. 48 (1992) 279–287. https://doi.org/10.1016/0165-0114(92)90343-3.

[122] M. Roopaei, M. Zolghadri Jahromi, Chattering-free fuzzy sliding mode control in MIMO uncertain systems, Nonlinear Anal. Theory, Methods Appl. 71 (2009) 4430–4437. https://doi.org/10.1016/j.na.2009.02.132.

[123] M. Sugeno, T. Takagi, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man. Cybern. 15 (1985) 116–132.

[124] L. Morales, J. Aguilar, A. Rosales, J.A. Gutierrez de Mesa, D. Chavez, An Intelligent Controller based on LAMDA, in: 2019 IEEE 4th Colomb. Conf. Autom. Control, IEEE, 2019: pp. 1–6. https://doi.org/10.1109/CCAC.2019.8921299.

[125] J.-J. Slotine, W. Li, Applied NoLinear Optimal Control, Prentice Hall, New Jersey, 1991.

[126] H.T. Yau, C.L. Chen, Chattering-free fuzzy sliding-mode control strategy for uncertain chaotic systems, Chaos, Solitons and Fractals. 30 (2006) 709–718. https://doi.org/10.1016/j.chaos.2006.03.077.

[127] O. Camacho, C.A. Smith, Sliding mode control: an approach to regulate nonlinear chemical processes, ISA Trans. 39 (2000) 205–218. https://doi.org/10.1016/S0019-0578(99)00043-9.

[128] P. Van Overschee, B. De Moor, RAPID: The End of Heuristic PID Tuning, IFAC Proc. Vol. 33 (2000) 595–600. https://doi.org/10.1016/S1474-6670(17)38308-8.

[129] S.J. Yoo, J.B. Park, Y.H. Choi, Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates, Inf. Sci. (Ny). 177 (2007) 3074–3098. https://doi.org/10.1016/j.ins.2007.02.009.

[130] Y. Bao, H. Wang, J. Zhang, Adaptive inverse control of variable speed wind turbine, Nonlinear Dyn. 61 (2010) 819–827. https://doi.org/10.1007/s11071-010-9689-3.

[131] T. Waegeman, B. Schrauwen, Towards learning inverse kinematics with a neural network based tracking controller, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7064 LNCS (2011) 441–448. https://doi.org/10.1007/978-3-642-24965-5_50.

[132] K. Mehran, Takagi-Sugeno Fuzzy Modeling for Process Control, Sch. Electr. Electron. Comput. Eng. …. (2008). http://www.staff.ncl.ac.uk/damian.giaouris/pdf/IA Automation1/TS FL tutorial.pdf.

[133] M.A. Shoorehdeli, M. Teshnehlab, A.K. Sedigh, M.A. Khanesar, Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods, Appl. Soft Comput. 9 (2009) 833–850. https://doi.org/10.1016/j.asoc.2008.11.001.

[134] M.A. Shoorehdeli, M. Teshnehlab, A.K. Sedigh, Training ANFIS as an identifier with intelligent hybrid stable learning algorithm based on particle swarm optimization and extended Kalman filter, Fuzzy Sets Syst. 160 (2009) 922–948. https://doi.org/10.1016/j.fss.2008.09.011.

[135] C. Paleologu, J. Benesty, S. Ciochiňa, A robust variable forgetting factor recursive least-squares algorithm for system identification, IEEE Signal Process. Lett. 15 (2008) 597–600. https://doi.org/10.1109/LSP.2008.2001559.

[136] L. Morales, J. Aguilar, A. Rosales, D. Chávez, P. Leica, Modeling and control of nonlinear systems using an Adaptive LAMDA approach, Appl. Soft Comput. 95 (2020). https://doi.org/10.1016/j.asoc.2020.106571.

[137] B. Kang, D. Wei, Y. Li, Y. Deng, A method of converting Z-number to classical fuzzy number, J. Inf. Comput. Sci. 9

(2012) 703–709.

[138] L. Morales, J. Aguilar, O. Camacho, A. Rosales, An intelligent sliding mode controller based on LAMDA for a class of SISO uncertain systems, Inf. Sci. (Ny). 567 (2021) 75–99. https://doi.org/10.1016/j.ins.2021.03.012.

[139] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Appl. Intell. 48 (2018) 4743–4759. https://doi.org/10.1007/s10489-018-1238-7.

[140] C.J. Veenman, M.J.T. Reinders, E. Backer, A maximum variance cluster algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1273–1280. https://doi.org/10.1109/TPAMI.2002.1033218.

[141] I. Tsamardinos, E. Greasidou, G. Borboudakis, Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation, Mach. Learn. 107 (2018) 1895–1922. https://doi.org/10.1007/s10994-018-5714-4.

[142] A. Balakrishnama, S., & Ganapathiraju, Linear discriminant analysis-a brief tutorial, Inst. Signal Inf. Process. 18 (1998) 1–8. https://doi.org/http://www.isip.piconepress.com/publications/reports/1998/isip/lda/.

[143] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 993–1001. https://doi.org/10.1109/34.58871.

[144] F. Schölkopf, B., Smola, A. J., & Bach, Learning with kernels: support vector machines, regularization, optimization, and beyond., MIT press, 2002.

[145] I. Rish, An empirical study of the naive Bayes classifier, IJCAI 2001 Work. Empir. Methods Artif. Intell. New York IBM. 3 (2001) 41–46. https://doi.org/10.1.1.330.2788.

[146] P.H. Swain, H. Hauska, The decision tree classifier: Design and potential, IEEE Trans. Geosci. Electron. 15 (1977) 142–147. https://doi.org/10.1109/TGE.1977.6498972.

[147] V.F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, J.P. Rigol-Sanchez, An assessment of the effectiveness of a random forest classifier for land-cover classification, ISPRS J. Photogramm. Remote Sens. 67

(2012) 93–104. https://doi.org/10.1016/j.isprsjprs.2011.11.002.

[148] J.G. Rohra, B. Perumal, S.J. Narayanan, P. Thakur, R.B. Bhatt, User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks, in: K. Deep, J.C. Bansal, K.N. Das, A.K. Lal, H. Garg, A.K. Nagar, M. Pant (Eds.), Adv. Intell. Syst. Comput., Springer Singapore, Singapore, 2017: pp. 286–295. https://doi.org/10.1007/978-981-10-3322-3_27.

[149] X. Gu, P.P. Angelov, Autonomous data-driven clustering for live data stream, in: 2016 IEEE Int. Conf. Syst. Man, Cybern., IEEE, 2016: pp. 001128–001135. https://doi.org/10.1109/SMC.2016.7844394.

[150] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Appl. Intell. 48 (2018) 4743–4759. https://doi.org/10.1007/s10489-018-1238-7.

[151] A. Ultsch, Clustering wih SOM: U*C, in: Proc. 5th Work. Self-Organizing Maps, 2005: pp. 75–82.

[152] J.A. Silva, E.R. Faria, R.C. Barros, E.R. Hruschka, A.C.P.L.F. de Carvalho, J. Gama, Data stream clustering, ACM Comput. Surv. 46 (2013) 1–31. https://doi.org/10.1145/2522968.2522981.

[153] L. Morales, D. Pozo, J. Aguilar, A. Rosales, Adaptive LAMDA applied to identify and regulate a process with variable dead time, in: 2020 IEEE Int. Conf. Fuzzy Syst., IEEE, 2020: pp. 1–8. https://doi.org/10.1109/FUZZ48607.2020.9177687.

[154] L. Morales, J. Aguilar, A. Garces-Jimenez, J.A. Gutierrez De Mesa, J.M. Gomez-Pulido, Advanced Fuzzy-Logic-Based Context-Driven Control for HVAC Management Systems in Buildings, IEEE Access. 8 (2020) 16111–16126. https://doi.org/10.1109/ACCESS.2020.2966545.

[155] C. Smith, A. Corripio, Principles and Practice of Automatic Process Control, Third Ed., John Wiley & Sons, Inc., 2006.

[156] E. Iglesias, Y. García, M. Sanjuan, O. Camacho, C. Smith, Fuzzy surface-based sliding mode control, ISA Trans. 46

(2007) 73–83. https://doi.org/10.1016/j.isatra.2006.04.002.

[157] S. Soyguder, M. Karakose, H. Alli, Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system, Expert Syst. Appl. 36 (2009) 4566–4573. https://doi.org/10.1016/j.eswa.2008.05.031.

[158] B. Arguello-Serrano, M. Velez-Reyes, Nonlinear control of a heating, ventilating, and air conditioning system with thermal load estimation, IEEE Trans. Control Syst. Technol. 7 (1999) 56–63. https://doi.org/10.1109/87.736752.

[159] E. Bristol, On a new measure of interaction for multivariable process control, IEEE Trans. Automat. Contr. 11 (1966) 133–134. https://doi.org/10.1109/TAC.1966.1098266.

[160] J.K. Pothal, D.R. Parhi, Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system, Rob. Auton. Syst. 72 (2015) 48–58. https://doi.org/10.1016/j.robot.2015.04.007.

[161] F.G. Rossomando, C. Soria, R. Carelli, Autonomous mobile robots navigation using RBF neural compensator, Control Eng. Pract. 19 (2011) 215–222. https://doi.org/10.1016/j.conengprac.2010.11.011.

[162] F.G. Rossomando, C.M. Soria, Identification and control of nonlinear dynamics of a mobile robot in discrete time using an adaptive technique based on neural PID, Neural Comput. Appl. 26 (2015) 1179–1191. https://doi.org/10.1007/s00521-014-1805-8.

[163] Adept Technology Inc., Pioneer 3-DX, (2011) 2.

[164] E. Rohmer, S.P.N. Singh, M. Freese, V-REP: A versatile and scalable robot simulation framework, in: 2013 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2013: pp. 1321–1326. https://doi.org/10.1109/IROS.2013.6696520.

[165] L. Morales, M. Herrera, O. Camacho, P. Leica, J. Aguilar, LAMDA Control Approaches Applied to Trajectory Tracking for Mobile Robots, IEEE Access. 9 (2021) 37179–37195. https://doi.org/10.1109/ACCESS.2021.3062202.

[166] Y.H. Kim, S.C. Ahn, W.H. Kwon, Computational complexity of general fuzzy logic control and its simplification for a loop

controller, Fuzzy Sets Syst. 111 (2000) 215–224. https://doi.org/10.1016/S0165-0114(97)00409-0.

[167] Q. Zhao, M. Xu, P. Fränti, Sum-of-Squares Based Cluster Validity Index and Significance Analysis, in: Int. Conf. Adapt. Nat. Comput. Algorithms, 2009: pp. 313–322. https://doi.org/10.1007/978-3-642-04921-7_32.

[168] I.W. Tsang, J.T. Kwok, P.M. Cheung, Core vector machines: Fast SVM training on very large data sets, J. Mach. Learn. Res. 6 (2005) 363–392.

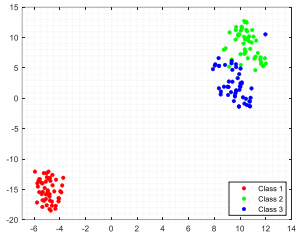
[169] K. Hassine, A. Erbad, R. Hamila, Important Complexity Reduction of Random Forest in Multi-Classification Problem, in: 2019 15th Int. Wirel. Commun. Mob. Comput. Conf., IEEE, 2019: pp. 226–231. https://doi.org/10.1109/IWCMC.2019.8766544.

[170] S. Firdaus, M. Uddin, A Survey on Clustering Algorithms and Complexity Analysis, Int. J. Comput. Sci. Issues. 12 (2015) 62.

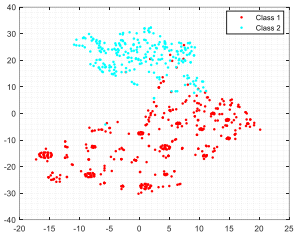
[171] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A.Y. Zomaya, S. Foufou, A. Bouras, A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis, IEEE Trans. Emerg. Top. Comput. 2 (2014) 267–279. https://doi.org/10.1109/TETC.2014.2330519.

APPENDIX A: Graphical representation of the datasets used in classification and clustering

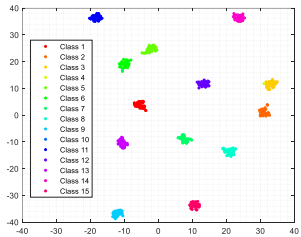
In this Appendix are shown the two-dimensional graphs of the datasets using the t-Distributed Stochastic Neighbor Embedding, method used for High-Dimensional Data reduction (tsne function in Matlab).



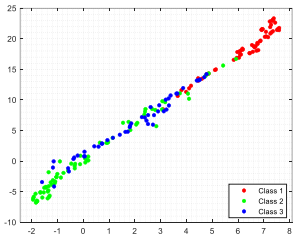
Iris Dataset



Breast Cancer Dataset



R15 Dataset



Wine Type Dataset

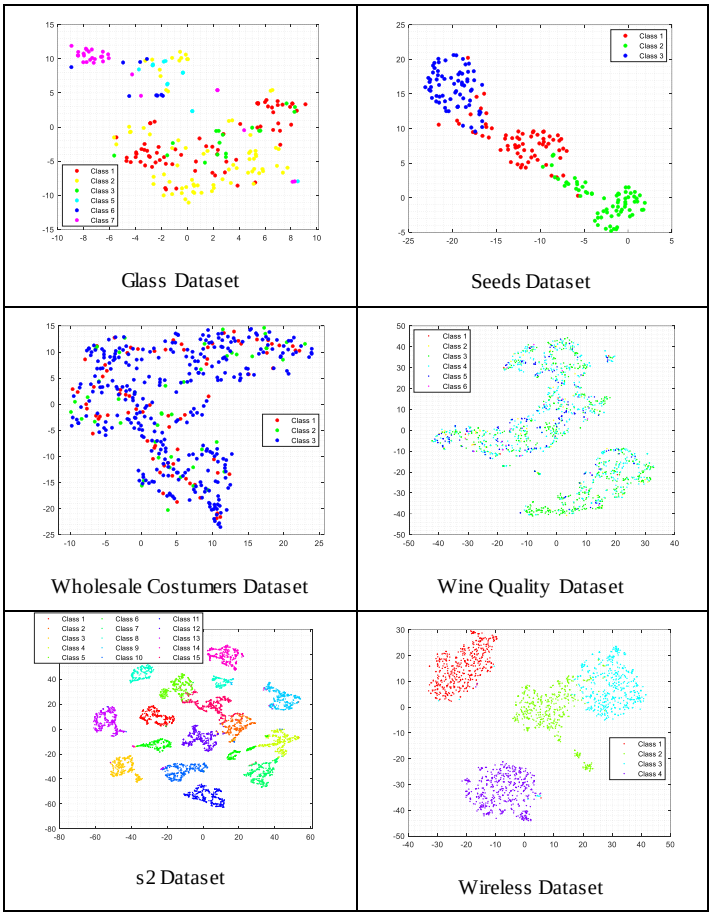
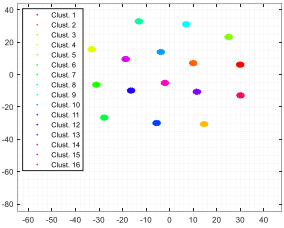
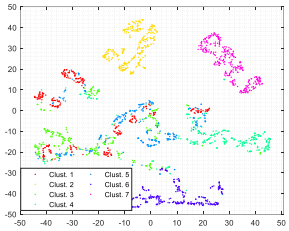


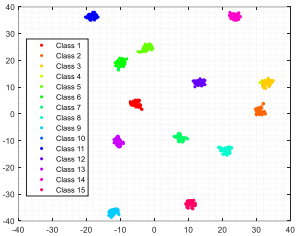
Figure A.1. Two-dimensional graphs of the classification datasets using the t-Distributed Stochastic Neighbor Embedding



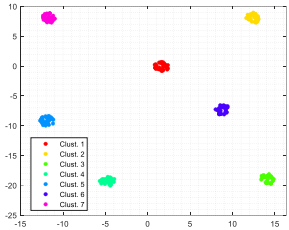
Dim 1024 Dataset



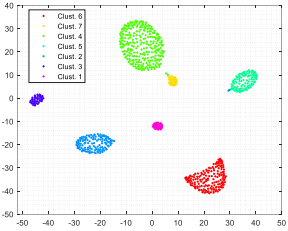
Segment Dataset



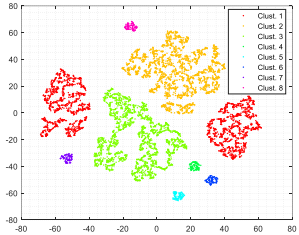
R15 Dataset



Hepta Dataset



Aggregation Dataset



Unbalance Dataset

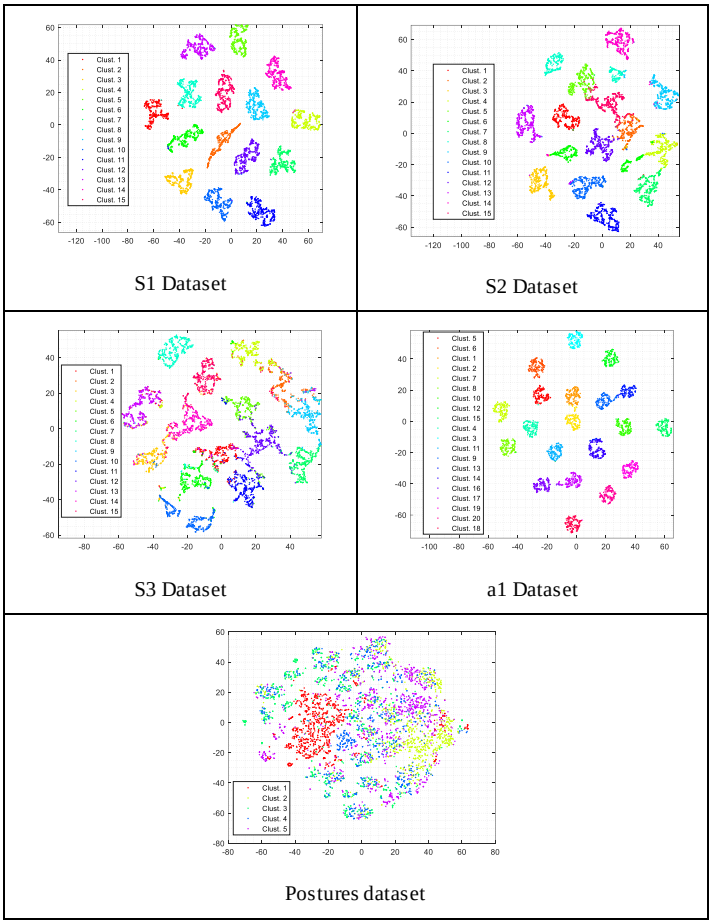
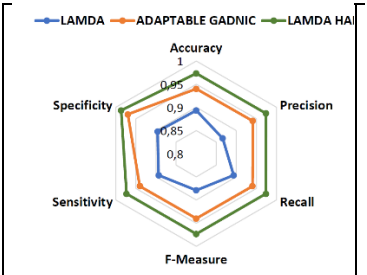


Figure A.2. Two-dimensional graphs of the clustering datasets using the t-Distributed Stochastic Neighbor Embedding

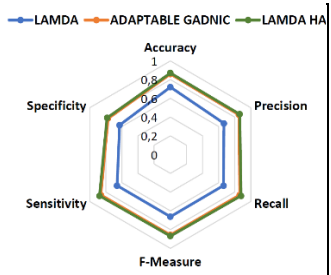
APPENDIX B: LAMDA-HAD EVALUATION

B.1. Comparison between LAMDA, LAMDA (Adaptable GAD_{NIC}) and LAMDA-HAD in classification benchmarks

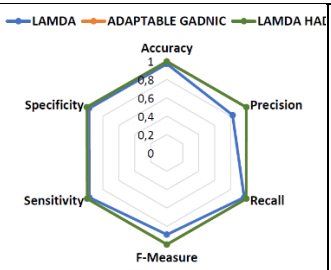
In this Appendix are shown the results of the tests of the algorithms LAMDA, LAMDA with adaptable $\mathrm{GAD}_{\mathrm{NIC}}$, and the complete LAMDA-HAD method, in order to observe the contribution of the two extensions in the performance of classification tasks during the test stage.



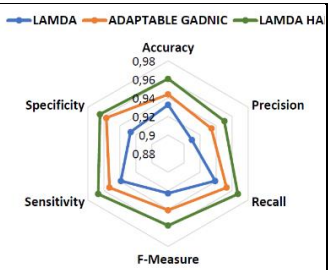
Iris Dataset



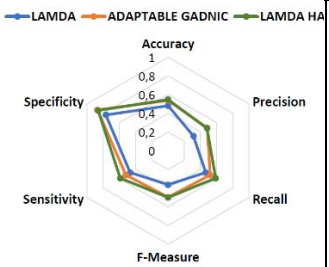
Breast Cancer Dataset



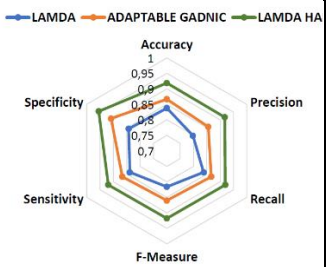
R15 Dataset



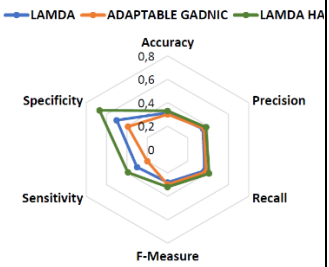
Wine Type Dataset



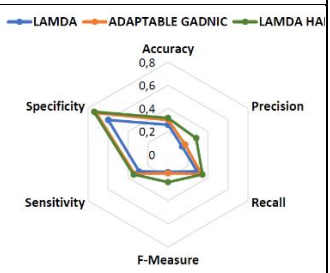
Glass Dataset



Seeds Dataset



Wholesale Costumers Dataset



Wine Quality Dataset

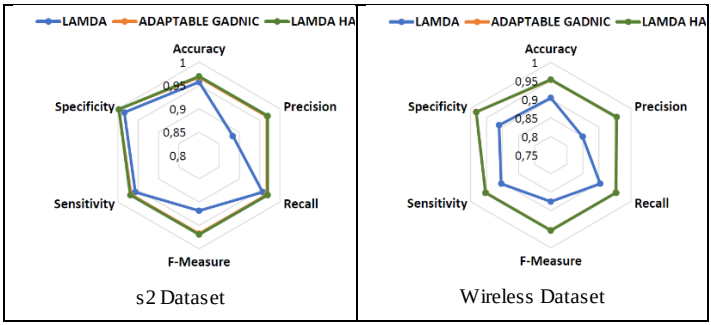


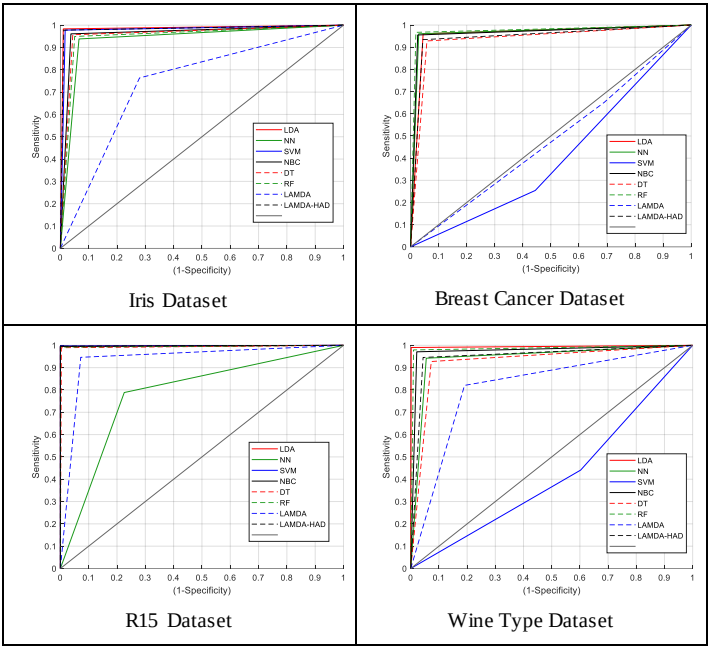
Figure B.1. Comparison between LAMDA, LAMDA (Adaptable GAD_{NIC}) and LAMDA-HAD in classification benchmarks

Figure A.1 shows that the two proposed extensions improve the performance of the original algorithm, reducing the number of misclassified individuals. The cases where the proposal still makes mistakes, is when there is a non-separability problem due to the definition of the data space, because some descriptors do not adequately characterize the individuals in the classes. The problem of maintaining a fixed NIC in all classes could be solved by establishing the GAD_{NIC}, for each class. The calculation of HAD reinforces the assignment of individuals to the classes, improving the measured performance metrics in all the benchmarks.

B.2. ROC CURVES OF THE TESTED CLASSIFIERS

In order to diagnose the performance of the algorithms, Figure B.2 shows the ROC curves of the classifiers for each of the datasets tested in the classification stage. The ROC curves can be obtained based on the calculation of the

micro/macro averages. However, in this work, in order not to confuse the reader with excess results, and as a complement to the values shown in Tables 5.2 and 5.3, the micro average results of the k fold cross-validation are presented to obtain the curves.



Iris Dataset | Breast Cancer Dataset

R15 Dataset | Wine Type Dataset

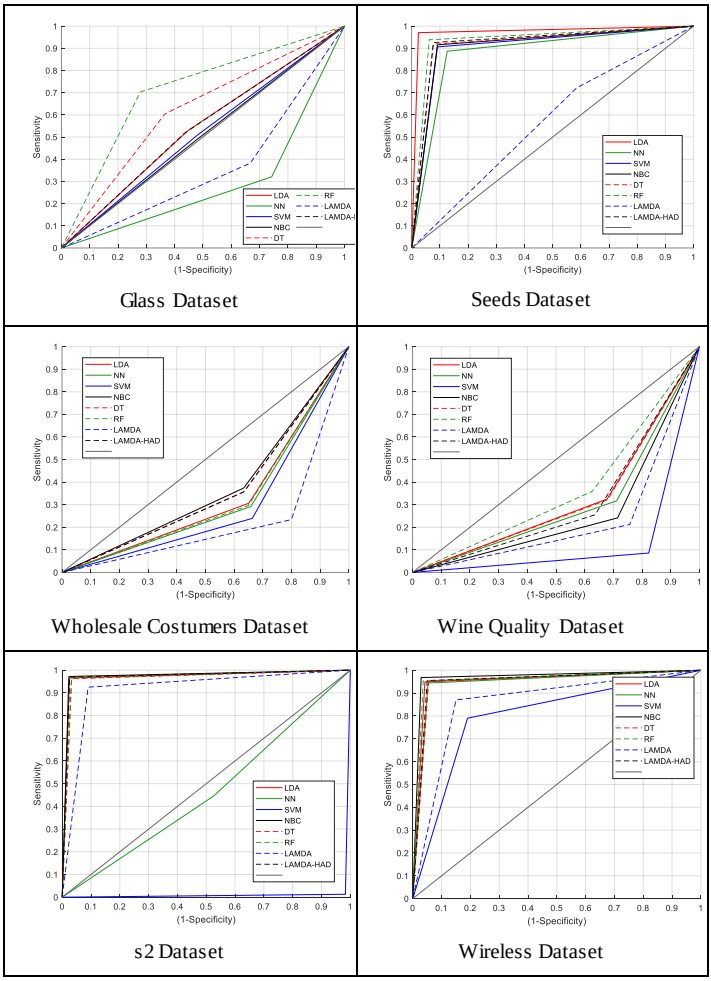


Figure B.2. Comparison of the micro-average ROC for the datasets with different classifiers:

The ROC curves in each of the datasets allow a comparative graphical analysis of the performance of the classifiers, where we can be observe that RF, LDA and LAMDA-HAD are the algorithms that present the best results in most data groups.This characteristic is also reflected in Table B.1, which shows the AUC values with their respective standard deviation.

Table B.1. Average AUC and Standard Deviation of LAMDA-HAD and other classification algorithms

		LDA	**NN**	**SVM**	**NBC**	**DT**	**RF**	**LMD**	**LM-HAD**
Iris	$\overline{\boldsymbol{v}}$	0,987	0,936	0,980	0,962	0,957	0,949	0,742	0,983
	$\boldsymbol{\sigma}$	0,012	0,001	0,005	0,003	0,016	0,008	0,019	0,005
Breast	$\overline{\boldsymbol{v}}$	0,958	0,966	0,405	0,965	0,935	0,974	0,481	0,945
	$\boldsymbol{\sigma}$	0,018	0,008	0,001	0,012	0,009	0,019	0,033	0,007
R15	$\overline{\boldsymbol{v}}$	0,998	0,781	0,999	0,998	0,993	0,996	0,937	0,998
	$\boldsymbol{\sigma}$	0,001	0,009	0,001	0,001	0,001	0,001	0,005	0,001
Wine-Type	$\overline{\boldsymbol{v}}$	0,994	0,944	0,419	0,975	0,927	0,985	0,815	0,951
	$\boldsymbol{\sigma}$	0,006	0,011	0,055	0,008	0,016	0,006	0,022	0,009
Glass	$\overline{\boldsymbol{v}}$	0,541	0,289	0,514	0,505	0,619	0,712	0,358	0,541
	$\boldsymbol{\sigma}$	0,051	0,040	0,025	0,030	0,060	0,011	0,037	0,051
Seeds	$\overline{\boldsymbol{v}}$	0,974	0,881	0,908	0,913	0,921	0,938	0,568	0,925
	$\boldsymbol{\sigma}$	0,008	0,004	0,007	0,006	0,007	0,004	0,007	0,008
Whol.	$\overline{\boldsymbol{v}}$	0,328	0,317	0,288	0,370	0,361	0,323	0,217	0,361
	$\boldsymbol{\sigma}$	0,019	0,018	0,017	0,025	0,057	0,042	0,045	0,025
Wine	$\overline{\boldsymbol{v}}$	0,324	0,303	0,131	0,263	0,321	0,366	0,228	0,310
	$\boldsymbol{\sigma}$	0,050	0,012	0,032	0,039	0,030	0,009	0,030	0,071
s2	$\overline{\boldsymbol{v}}$	0,970	0,460	0,015	0,975	0,965	0,967	0,918	0,972
	$\boldsymbol{\sigma}$	0,005	0,001	0,006	0,002	0,005	0,003	0,006	0,004
Wire-less	$\overline{\boldsymbol{v}}$	0,954	0,946	0,800	0,970	0,948	0,959	0,860	0,953
	$\boldsymbol{\sigma}$	0,006	0,005	0,002	0,003	0,004	0,001	0,004	0,002

APPENDIX C: LAMDA-RD parameter calibration

In this Appendix, it is presented the LAMDA-RD parameters calibration, which affect the quality and number of the formed clusters. These parameters are: d_{nb} (Definition 7) and D_t (Proposition 3).

To start with the parameter calibration methodology, some metrics used in clustering are described.

C.1. Metrics for clustering analysis

Silhouette coefficient (SC): it is a metric between [-1,1], -1 for incorrect clustering and 1 for highly dense clustering (dense and well separated), values around zero indicate overlapping clusters. This is composed of two values, $a_{SC}(x)$ is the mean distance between an individual and all other individuals in the same cluster, and $b_{SC}(x)$ is the mean distance between an individual and all other individuals in the nearest cluster. If the value is bigger, then the clustering is better. Considering N, the number of elements of the dataset, SC is computed as:

$$SC = \frac{1}{N}\sum_{x \in X}\left(\frac{b_{SC}(x) - a_{SC}(x)}{max(a_{SC}(x), b_{SC}(x))}\right) \qquad (C.1)$$

Sum-of-squares within clusters (SSW): it is an internal measure used to evaluate the cohesion of the clusters that the algorithm has generated. The smaller the value is, the better the clustering. It is defined by (C.2).

$$SSW(C,m) = \frac{1}{N}\sum_{k=1}^{m}\sum_{i\epsilon C_k}\left\|\bar{X}_k^i - \rho_k\right\| \qquad (C.2)$$

$\bar{X}_k^i$ is the $i-th$ individual in the cluster C_k, ρ_k is its centroid, and m is the number of clusters.

Sum-of-squares between clusters (SSB): it is a prototype-based separation measure used to evaluate the inter-cluster distance. If the value is bigger, then the clustering is better. It is defined by (C.3).

$$SSB(C,m) = \frac{1}{n_k}\sum_{k=1}^{m} n_k\left(\rho_k - \rho_g\right) \qquad (C.3)$$

where n_k is the number of elements in the cluster k, ρ_g is the mean value of the whole data set (global center).

WB-index (WB_{index})[167]: it is based on SSW and SSB. It emphasizes the effect of SSW multiplying it by the generated number of clusters m. This metric is an alternative to methods based on knee point detection because most indices show monotonicity with increasing number of clusters. Therefore, indices with a clear minimum or maximum value are preferred, being WB_{index} one of them. Being a relationship between SSW and SSB, it can be noted that lower its value, the better the quality of the formed clusters. In cases in which it is necessary to know the optimal number of groups, the WB-index are plotted for different number of partitions, and the model with the minimum value is chosen as the optimum. This index is defined in (C.4).

$$WB_{index} = \frac{m \times SSW}{SSB} \qquad (C.4)$$

Performance Coefficient (P_c): it is a metric that we propose which is a relationship between SC or $SILA$ and WB-index, in order to establish which of the tested algorithms presents the best performance. The value of P_C must be minimal and greater than zero, because WB must be small and SC must be positive and close to 1, to establish an adequate clustering.

$$P_C = \frac{WB_{index}}{SC} \qquad (C.5)$$

C.2. LAMDA-RD calibration

A guideline for the calibration is presented below, which shows how the variation of the parameters d_{nb} and D_t (necessary to be set by the user) affects the quality of the clusters for the case of R15. Figures B.1a shows the variations of P_C, depending on the parameters d_{nb} and D_t, and Figure B.1b shows its top view, in which the different areas are represented in colors. The yellow zone, e.g. ($D_t = 0.1$, $d_{nb} = 0.27$, $P_C = 15$) presents high P_C values, which as detailed in the experimental tests, this implies poor quality in the created clusters. Based on this, it is necessary to look for the zone with the minimum P_C, in this case, the dark blue zones (which shows the next values: $D_t = 0.3$, $d_{nb} = 0.03$, $P_C = 1.577$), that is, good quality clusters (the lowest P_C). However, the number of created clusters m must also be considered, which is represented in Figure B.2a as a function of the parameters d_{nb} and D_t, and Figure B.2b shows its top view. In this case, the yellow areas represent a high number of created clusters ($D_t = 0.9$, $d_{nb} = 0.015$, $m = 56$ clusters), while dark blue areas ($D_t = 0.1$, $d_{nb} = 0.27$, $m = 1$ cluster) are not useful because all data has been

grouped into a single cluster. Finally, we observe the green zone ($D_t = 0.3, \quad d_{nb} = 0.03, \quad m = 15$ clusters), which coincides with the values of d_{nb} and D_t with the minimum P_C (see Figure. B.1). So, the general idea of the method is to find a balance between P_C and m.

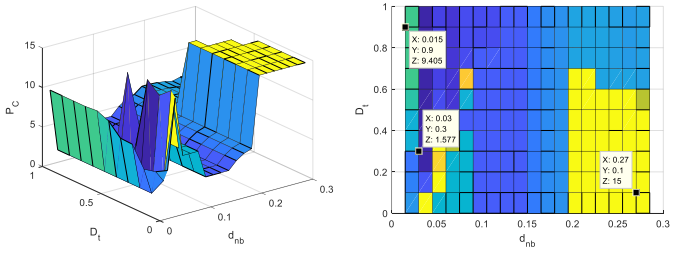
Based on the results of Figure C.1 and C.2, the following criteria can be established:

- Low values of D_t make the merging process between neighboring clusters with low or no density in the overlapped area (see Figure C.3a), which is not adequate since they produce a non-demanding or low exigency algorithm (as is shown in Figure C.2 for dark blue zones), performing the merge process with separate or dissimilar neighboring clusters, which leads to poor quality clusters, as is shown by high P_C in Figure C.1 for the equivalent zone (yellow area).

- High values of D_t produce a more demanding algorithm (as is shown clearly in Figure C.2 for yellow zones) since it requires a higher percentage of individuals in the overlapping area (see in Figure C.3), performing the merge process only when the neighboring clusters are very close, which improves the quality of them, as is shown by low P_C in Figure C.1 for the equivalent zone (dark blue area).

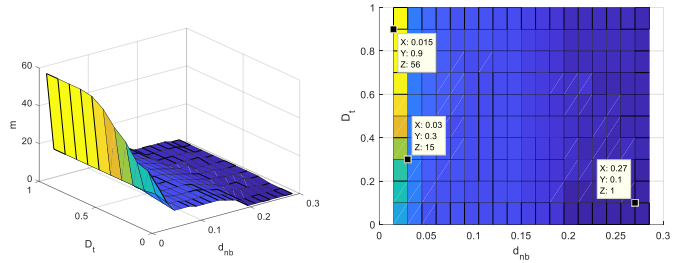
- Low values of distance between neighbors d_{nb} allow obtaining a more demanding algorithm (see Figures C.1 and C.2, the best P_C and a non-excessive number of clusters m is presented with a low d_{nb}), since the calculation of $K_{k,j}$ is stricter (strongly penalizing the dissimilarity between samples). High values of d_{nb}

produce a non-demanding or low exigency algorithm, by weakening the penalization for the dissimilarity between samples.



(a) (b)

Figure C.1. Obtained results for $R15$, in function of d_{nb} and D_t: (a) P_C. (b) top view of P_C



(a) (b)

Figure C.2. Obtained results for $R15$, as function of d_{nb} and D_t: (a) number of clusters "m". (b) top view of the number of clusters

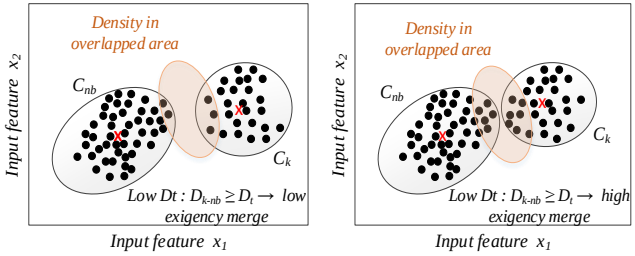


Figure C.3. Illustrative example of: (a) low value of D_t. (b) high value of D_t

Low values of distance between neighbors d_{nb} allow obtaining a more demanding algorithm (see Figures C.1 and C.2, the best P_C and a non-excessive number of clusters m is presented with a low d_{nb}), since the calculation of $K_{k,j}$ is stricter (strongly penalizing the dissimilarity between samples). High values of d_{nb} produce a non-demanding or low exigency algorithm, by weakening the penalization for the dissimilarity between samples.

Figure C.4 shows the recommended zone for the initial parameter calibration, looking for a balance zone in Figure C.1 and C.2 to obtain a better P_C, without creating an excessive number of clusters m. Based on this, it is possible to verify the quadrants of maximum and minimum exigency, and the balanced zone, which can be taken as a starting point to perform the search of the most appropriate D_t and d_{nb}.

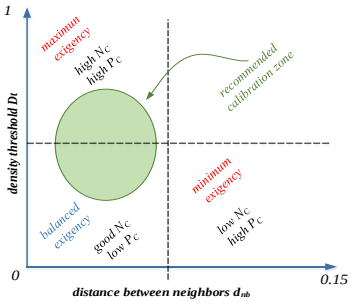


Figure C.4. Recommended calibration of D_t and d_{nb}

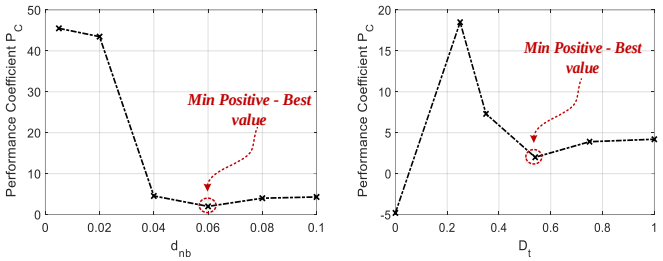
The critical cases occur in the yellow zones of Figure C.2, low D_t and high d_{nb}, which generate the minimum number of clusters. In the case of R15, the best results are obtained by calibrating d_{nb} to a small value, as is shown in Figure C.2, where we have a great variation of D_t. Now, contrasting the results with Figure C.1, the smaller P_C must be located on the graph.

From the experimentation, a generic behavior could be observed, concluding that the parameter calibration can start with a value of $D_t \approx 0.5$, and $d_{nb} \approx 0.1 \times D_t$, e.g. for R15 the best values are: $D_t = 0.3$, $d_{nb} = 0.03$ (shaded area of Figure C.4).

For example, for s1 dataset the formed clusters with different parameter values are shown below, in which the parameter D_t is initially set $D_t = 0.54$, and d_{nb} is changed until finding the minimum P_C looking for an adequate number and quality of clusters (in this case, 15 clusters). By increasing the value of d_{nb}, the algorithm creates fewer clusters, which are better constituted by covering the more dispersed individuals. On the other hand, by setting the value of d_{nb}, fixed, and

changing the values of D_t, we observe that the algorithm is less strict when it is small, which implies a decrease in the number of clusters (less strict).

The behavior of P_C for the case when D_t is fixed and d_{nb} is changed, is shown in Figure B.5a, while the behavior of P_C for the case when d_{nb} is fixed and D_t is changed, is shown in Figure C.5b; in both cases, the minimum P_C is a guide to calibrate these parameters.



(a) (b)

Figure C.5. Performance Coefficient for different values of d_{nb} and D_t (marked in red the best value)

APPENDIX D: Computational Complexity of LAMDA-HAD and LAMDA-RD

D.1. Memory Usage

The number of parameters required to perform the classification/clustering tasks is based on the number of descriptors and formed classes/clusters, l and m, respectively. According to The number of parameters ($\#parameters$) to be stored in memory is:

$$\#parameters_{HAD} = lm + 2n + 3 \qquad (D.1)$$

$$\#parameters_{RD} = lm + 2n + 3 \qquad (D.2)$$

In addition, if there are N samples, each with n descriptors, the total number of stored values is:

$$\#stored\_values_{HAD} = Nl \qquad (D.3)$$

$$\#stored\_values_{RD} = Nl \qquad (D.4)$$

It is assumed that each value is stored in 2 bytes of memory [166]. It can be concluded that its complexity linearly increases therefore the Big(O) function of LAMDA-HAD and LAMDA-RD is:

$$f(Nl) = O(Nl) \qquad (D.5)$$

D.2. Number of operations

In this subsection is evaluated the number of arithmetic operations (arithmetic complexity) used to solve a problem. Addition, subtraction, multiplication, division, power and root

are considered as basic operations. The number of operations in each step to assign one sample to a cluster is detailed in Table D.1 [107]. Note that the symbol -- indicates that the algorithm does not perform that operation.

Table D.1. Arithmetic complexity (number of operations) of LAMDA, LAMDA-HAD and LAMDA-RD

	Arithmetic Complexity		
	LAMDA	**LAMDA-HAD**	**LAMDA-RD**
For normalization	3	3	3
For $MAD_{k,j}$	$4ml$	$4ml$	--
For update $\rho_{k,j}$	4		4
For $CMAD$	--	--	$3ml$
For $d_{k,\bar{X}}$	--	--	$m(l-1)+2$
For $K_{k,\bar{X}}$	--	--	$3m$
For $RMAD_{k,j}$,	--	--	m
For $GAD_{k,j}$	$21(m+1)(l-1)$	$21(m+1)(l-1)$	$21(m+1)(l-1)$
For $MGAD_{k,p}$	--	n_k+1	--
For GAD_{NIC}	--	$m+1$	--
For AD_{GAD}	--	$5m^2$	--
For HAD	--	m	--
Class/Cluster identif.	1	2	1
For $t_{nb,j}$	--	--	$n(n_{nb}^2-n_{nb}+1)$
Count individuals in overlapping	--	--	$n_k+n_{nb}+1$
For D_{k-nb}	--	--	3
In the case of merge, to update $\rho_{new,j}$	--	--	$n_k+n_{nb}+1$

The arithmetic complexity of LAMDA in the classification context (C_{LC}), compared with LAMDA-HAD (C_{HAD}) is computed adding the cells in each column of the Table D.1 and multiplying for the number of data instances N.

$$C_{LC} = N(25ml - 21m + 21l - 13)$$
$$\rightarrow \quad f(Nml) = O(Nml) \qquad (D.6)$$
$$C_{HAD} = N(5m^2 + 25ml - 20m + 21l + n_k - 11)$$
$$\rightarrow f(Nm^2) = O(Nm^2) \qquad (D.7)$$

The comparative results in asymptotic notation (Big-O) of the algorithms tested in the classification context (training) are presented in Table D.2.

Table D.2. Arithmetic complexity in terms of Big-O of classification algorithms

Algorithm	Arithmetic Complexity big-O Notation
LDA	$O(Nl^2)$
SVM	$O(N^3)$
NBC	$O(Nl)$
DT	$O(Nl\log(n))$
RF	$O(MNl\log(n))$
LAMDA	$O(Nml)$
LAMDA-HAD	$O(Nm^2)$

**M is the number of trees, information taken from* [168,169]

In the other hand, the arithmetic complexity of LAMDA in the clustering context (C_{LU}), compared with LAMDA-RD (C_{RD}) is computed adding the cells in each column of the Table D.1.

$$C_{LU} = 25ml - 21m + 21l - 13$$
$$\rightarrow \quad f(ml) = O(ml) \qquad (D.8)$$

$$C_{RD} = 25ml - 18m + l(n_{nb}^2 - n_{nb} + 22) + 2(n_{nb} + n_k) - 7$$

$$\rightarrow f(ln_{nb}^2) = O(ln_{nb}^2), \quad if\ n_{nb} \approx N \Rightarrow \quad O(N^2 l) \qquad (D.9)$$

The comparative results in asymptotic notation (Big-O) of the algorithms tested in the clustering context are presented in Table D.3.

Table D.3. Arithmetic complexity in terms of Big-O of clustering algorithms

Algorithm	Arithmetic Complexity big-O Notation
KM	$O(Nkl)$
KMD	$O(N^2 lt)$
FCM	$O(N)$
ATH	$O(N^3)$
DBS	$O(Nl)$
LAMDA	$O(ml)$
LAMDA-RD	$O(N^2 l)$

**k is the number of clusters and t the number of iterations, information taken from* [170,171]

APPENDIX E: Demonstration of the stability of the learning algorithm

This section shows in detail the stability analysis of the Adaptive-LAMDA method, specifically in the part of learning during its application in the control loop. For this we have raised the Theorem 2, whose proof is detailed as follows.

Theorem 2. Consider the learning algorithm presented in Subsection 4.3.1, with the output error defined as:

$$E_d(k) = \frac{1}{2}[Out^d(k) - Out_L^d(k)]^2 = \frac{1}{2}e(k)^2 \quad (E.1)$$

the Lyapunov function defined as:

$$V(k) = E_d(k) \quad (E.2)$$

and:

$$0 < \eta < \frac{2}{3(N_\Upsilon + N_\Psi + N_\alpha)} \quad (E.3)$$

For $e_r(k) > 0$:

$$0 < \beta < \frac{2e(k)}{3\left(\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right)} \quad (E.4)$$

$$0 < tr\big(P(k+1)\big) < \frac{2e(k)}{3e_r(k)(\|a(k+1)\|_F)^2} \quad (E.5)$$

For $e_r(k) < 0$:

$$0 < \beta <$$

$$\frac{-2e(k)}{3\left(\left(\frac{\partial Out_l^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right)} \quad (E.6)$$

$$0 < tr\big(P(k+1)\big) < \frac{-2e(k)}{3e_r(k)(\|a(k+1)\|_F)^2} \qquad (E.7)$$

Then, the error $e(k) \longrightarrow 0$ in the training stage for a controllable system independent of the application, system order, number of inputs and classes.

Proof. To demonstrate the convergence of the algorithm, the Lyapunov's theory is used. Then, the change of $V(k)$ defined in (E.2) is computed as:

$$\begin{aligned}\Delta V(k) &= V(k+1) - V(k) = E_d(k+1) - E_d(k)\\ &= \frac{1}{2}(e(k+1)^2 - e(k)^2)\\ &= \frac{1}{2}\big(e(k+1) - e(k)\big)\big(e(k+1) + e(k)\big)\\ &= \frac{1}{2}\Delta e(k)\big(\Delta e(k) + 2e(k)\big) \qquad (E.8)\end{aligned}$$

Grouping all the terms of the antecedent in vector form $\Upsilon(k)$ for the centers and $\Psi(k)$ for the standard deviation of the classes:

$$\Upsilon(k) = [\rho_{1,1}(k), \dots, \rho_{1,j}(k), \dots, \rho_{2,1}(k), \dots, \rho_{2,j}(k), \dots, \rho_{k,1}(k), \dots, \rho_{k,j}(k), \dots, \rho_{m,n}(k)]^T \quad (E.9)$$

$$\Psi(k) = [\sigma_{1,1}(k), \dots, \sigma_{1,j}(k), \dots, \sigma_{2,1}(k), \dots, \sigma_{2,j}(k), \dots, \sigma_{k,1}(k), \dots, \sigma_{k,j}(k), \dots, \sigma_{m,n}(k)]^T \quad (E.10)$$

and the consequent parameters in the matrix $h(k)$, from (4.53):

$$h(k) = \begin{bmatrix} h_{11} & \dots & h_{1j} & \dots & h_{1n} & h_1 \\ h_{k1} & \dots & h_{kj} & \dots & h_{kn} & h_k \\ h_{m1} & \dots & h_{mj} & \dots & h_{mn} & h_m \end{bmatrix} \qquad (E.11)$$

The change in error $\Delta e(k)$ can be approximated by:

$$\Delta e(k) = \left(\frac{\partial e(k)}{\Upsilon(k)}\right)^T \Delta\Upsilon(k) + \left(\frac{\partial e(k)}{\partial\Psi(k)}\right)^T \Delta\Psi(k) + \left(\frac{\partial e(k)}{\partial\alpha(k)}\right)\Delta\alpha(k) + tr\left(\left(\frac{\partial e(k)}{\partial h(k)}\right)\Delta h(k)\right) \quad (E.12)$$

Updating the centers by the gradient descent method:

$$\Upsilon(k+1) = \Upsilon(k) + \eta\left(-\frac{\partial E_d(k)}{\partial\Upsilon(k)}\right) + \beta\big(\Upsilon(k) - \Upsilon(k-1)\big) \quad (E.13)$$

$$\frac{\partial E_d(k)}{\partial\Upsilon(k)} = \frac{\partial E_d(k)}{\partial e(k)}\frac{\partial e(k)}{\partial Out_L^d(k)}\frac{Out_L^d(k)}{\partial\Upsilon(k)} = -e(k)\frac{\partial Out_L^d(k)}{\partial\Upsilon(k)} \quad (E.14)$$

The change of $\Upsilon(k)$, replacing (E.14) in (E.13), is:

$$\Delta\Upsilon(k) = \eta e(k)\frac{\partial Out_L^d(k)}{\partial\Upsilon(k)} + \beta\Delta\Upsilon(k-1) \qquad (E.15)$$

Applying the same procedure of (E.13)-(E.15) for $\Psi(k)$ and $\alpha(k)$, it is obtained:

$$\Delta\Psi(k) = \eta e(k)\frac{\partial Out_L^d(k)}{\partial\Psi(k)} + \beta\Delta\Psi(k-1) \qquad (E.16)$$

$$\Delta\alpha(k) = \eta e(k)\frac{\partial Out_L^d(k)}{\partial\alpha(k)} + \beta\Delta\alpha(k-1) \qquad (E.17)$$

Rewriting (4.53) to compute the consequent parameters $h(k+1)$:

$$h(k+1) = h(k) + P(k+1)a(k+1)e_r(k) \qquad (E.18)$$

with:

$$e_r(k) = Out(k+1) - a^T(k+1)h(k) \qquad (E.19)$$

From (E.18), the change of $h(k)$ is computed by:

$$\Delta h(k) = P(k+1)a(k+1)e_r(k) \qquad (E.20)$$

Thus, the estimated output of LAMDA for the d –th sample from (4.53) is:

$$Out_L^d(k+1) = a^T(k+1)h(k) \qquad (E.21)$$

From (E.1) and (E.21), the derivative is:

$$\frac{\partial e(k)}{\partial h(k)} = \frac{\partial e(k)}{\partial Out_L^d(k)} \frac{\partial Out_L^d(k)}{\partial h(k)} = -a^T(k+1) \qquad (E.22)$$

Replacing (E.15)-(E.17) and (E.22) in (E.12), $\Delta e(k)$ is computed as:

$$\Delta e(k) = \left(-\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \left(\eta e(k) \frac{\partial Out_L^d(k)}{\partial \Upsilon(k)} + \beta \Delta \Upsilon(k-1)\right)$$
$$+ \left(-\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \left(\eta e(k) \frac{\partial Out_L^d(k)}{\partial \Psi(k)} + \beta \Delta \Psi(k-1)\right)$$
$$+ \left(-\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right) \left(\eta e(k) \frac{\partial Out_L^d(k)}{\partial \alpha(k)} + \beta \Delta \alpha(k-1)\right)$$
$$- a^T(k+1)P(k+1)a(k+1)e_r(k) \qquad (E.23)$$

$\Delta e(k) =$

$$-\eta e(k) \left(\left\| \frac{\partial Out_L^d(k)}{\partial \Upsilon(k)} \right\|_2 \right)^2 - \left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \beta \Delta \Upsilon(k-1)$$
$$-\eta e(k) \left(\left\| \frac{\partial Out_L^d(k)}{\partial \Psi(k)} \right\|_2 \right)^2 - \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \beta \Delta \Psi(k-1)$$

$$-\eta e(k)\left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)^2-\left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\beta\Delta\alpha(k-1)$$
$$-a^T(k+1)P(k+1)a(k+1)e_r(k) \quad (E.24)$$

Now, the following norms are replaced with the terms $N_\Upsilon, N_\Psi, N_\alpha$:

$$N_\Upsilon=\left(\|\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\|_2\right)^2, N_\Psi=\left(\|\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\|_2\right)^2,$$
$$N_\alpha=\left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)^2 \quad (E.25)$$

Replacing (E.25) in (E.24):

$$\Delta e(k)=-\eta e(k)N_\Upsilon-\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T\beta\Delta\Upsilon(k-1)$$
$$-\eta e(k)N_\Psi-\left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T\beta\Delta\Psi(k-1)$$
$$-\eta e(k)N_\alpha-\left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\beta\Delta\alpha(k-1)$$
$$-a^T(k+1)P(k+1)a(k+1)e_r(k) \quad (E.26)$$

$$\Delta e(k)=-e(k)\left[\eta N_\Upsilon+\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T\frac{\beta}{e(k)}\Delta\Upsilon(k-1)+\eta N_\Psi\right.$$
$$+\left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T\frac{\beta}{e(k)}\Delta\Psi(k-1)+\eta N_\alpha$$
$$+\left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\frac{\beta}{e(k)}\Delta\alpha(k-1)$$
$$\left.+a^T(k+1)P(k+1)a(k+1)\frac{e_r(k)}{e(k)}\right] (E.27)$$

Replacing (E.27) in $\Delta V(k)$ presented in (E.8):

$$\Delta V(k) = \frac{1}{2}e^2(k)\Bigg[\eta(N_\Upsilon + N_\Psi + N_\alpha) + \frac{\beta}{e(k)}\Bigg(\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\Bigg) + a^T(k+1)P(k+1)a(k+1)\frac{e_r(k)}{e(k)}\Bigg]$$

$$\Bigg[\eta(N_\Upsilon + N_\Psi + N_\alpha) + \frac{\beta}{e(k)}\Bigg(\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\Bigg) + a^T(k+1)P(k+1)a(k+1)\frac{e_r(k)}{e(k)} - 2\Bigg] \quad (E.28)$$

From (E.28), the following equalities are considered:

$$A(k) = \eta(N_\Upsilon + N_\Psi + N_\alpha) \qquad (E.29)$$

$$B(k) = \frac{\beta}{e(k)}\left(\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right) \qquad (E.30)$$

$$C(k) = a^T(k+1)P(k+1)a(k+1)\frac{e_r(k)}{e(k)} \qquad (E.31)$$

Replacing (E.29)-(E.31) in (E.28), to guarantee the convergence and stability, $\Delta V(k)$ must meet the condition:

$$\Delta V(k) < 0 \Rightarrow \frac{1}{2}e^2(k)[A(k) + B(k) + C(k)][A(k) + B(k) + C(k) - 2] < 0$$
$$\Rightarrow 0 < A(k) + B(k) + C(k) < 2 \qquad (E.32)$$

A is always positive, while the signs of $B(k)$ and $C(k)$ must be evaluated to meet the condition presented in (E.32), if $B(k) > 0$:

$$\frac{\beta}{e(k)}\left(\left(\frac{\partial Out_L^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_L^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right) > 0 \qquad (E.33)$$

Because $P(k+1)$ is Hermitanian semidefinite positive [133], then:

$$\frac{e_r(k)}{e(k)} > 0 \implies C(k) > 0 \qquad (E.34)$$

For stability, it is sufficient to consider the same weights for all the terms in (E.32):

$$0 < A(k) < \frac{2}{3};\ 0 < B(k) < \frac{2}{3}\ ;\ 0 < C(k) < \frac{2}{3} \qquad (E.35)$$

From (E.29), (E.33)-(E.35), it is obtained:

$$0 < \eta < \frac{2}{3(N_\Upsilon + N_\Psi + N_\alpha)} \qquad (E.36)$$

$$0 < \beta < \frac{2e(k)}{3\left(\left(\frac{\partial Out_l^d(k)}{\partial \Upsilon(k)}\right)^T \Delta\Upsilon(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \Psi(k)}\right)^T \Delta\Psi(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \alpha(k)}\right)\Delta\alpha(k-1)\right)} \qquad (E.37)$$

$$0 < a^T(k+1)P(k+1)a(k+1)\frac{e_r(k)}{e(k)} < \frac{2}{3} \qquad (E.38)$$

Considering the property: $XYX^T = tr(YX^TX)$, and applying it in (E.38):

$$0 < a^T(k+1)P(k+1)a(k+1) =$$

$$tr\big(P(k+1)a(k+1)a^T(k+1)\big) < \frac{2e(k)}{3e_r(k)} \qquad (E.39)$$

Applying the property $W, Z \in H_0^+(n)$, then

$0 \le tr\,WZ \le tr\,W\ tr\,Z$, in (E.39):

$$0 < tr\big(P(k+1)\big)tr\big(a(k+1)a^T(k+1)\big) < \frac{2e(k)}{3e_r(k)}$$

$$0 < tr\big(P(k+1)\big)(\|a(k+1)\|_F)^2 < \frac{2e(k)}{3e_r(k)}$$

$$0 < tr\big(P(k+1)\big) < \frac{2e(k)}{3e_r(k)(\|a(k+1)\|_F)^2} \qquad (E.40)$$

Now, if $B(k) < 0$:

$$0 < -B(k) < \frac{2}{3} \qquad (E.41)$$

$$0 < \beta < \frac{-2e(k)}{3\left(\left(\frac{\partial Out_l^d(k)}{\partial Y(k)}\right)^T \Delta Y(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \Psi(k)}\right)^T \Delta \Psi(k-1) + \left(\frac{\partial Out_l^d(k)}{\partial \alpha(k)}\right) \Delta \alpha(k-1)\right)} \qquad (E.42)$$

From (E.35), if $C(k) < 0$:

$$\frac{e_r(k)}{e(k)} < 0 \qquad (E.43)$$

$$0 < -C(k) < \frac{2}{3} \qquad (E.44)$$

Considering the same procedure from (E.39)-(E.40), in (E.44), it is obtained:

$$0 < -a^T(k+1)P(k+1)a(k+1)\frac{e_r(k)}{e(k)} < \frac{2}{3}$$

$$0 > -a^T(k+1)P(k+1)a(k+1) > \frac{2}{3}\frac{e(k)}{e_r(k)}$$

$$0 < a^T(k+1)P(k+1)a(k+1) < -\frac{2}{3}\frac{e(k)}{e_r(k)}$$

$$0 < tr\big(P(k+1)\big) < \frac{-2e(k)}{3e_r(k)(\|a(k+1)\|_F)^2} \qquad (E.45)$$

ORDEN DE EMPASTADO