

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY  
CAMPUS MONTERREY**

**ESCUELA DE INGENIERÍA Y TECNOLOGÍAS DE INFORMACIÓN**



**COMBINING ARTIFICIAL INTELLIGENCE AND ROBUST  
TECHNIQUES WITH MRAC IN FAULT TOLERANT CONTROL**

**TESIS**

**PRESENTADA COMO REQUISITO PARCIAL PARA OBTENER EL GRADO  
ACADEMICO DE:**

**DOCTOR EN CIENCIAS DE INGENIERÍA CON  
ESPECIALIDAD EN MECATRÓNICA**

**POR:**

**ADRIANA VARGAS MARTÍNEZ**

**MONTERREY, N. L.**

**DICIEMBRE DE 2011**

© Copyright by Adriana Vargas Martínez, 2011  
All Rights Reserved

# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

## CAMPUS MONTERREY

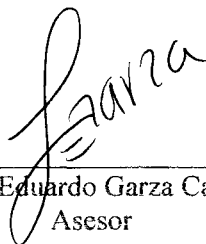
### ESCUELA DE INGENIERÍA Y TECNOLOGÍAS DE INFORMACIÓN

Los miembros del Comité de Tesis recomendamos que este documento de disertación presentado por la M. en C. Adriana Vargas Martínez, sea aceptado como requisito parcial para obtener el grado académico de:

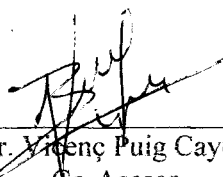
Doctor en Ciencias de Ingeniería con

Especialidad en Mecatrónica

Comité de Tesis:



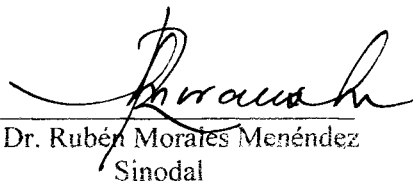
Dr. Luis Eduardo Garza Castañón  
Asesor



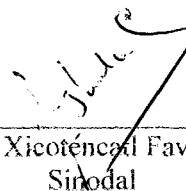
Dr. Videnc Puig Cayuela  
Co-Asesor



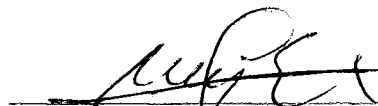
Dr. Efraín Alcorta García  
Co-Asesor



Dr. Rubén Morales Menéndez  
Sinodal



Dr. Antonio Ramón Xicoténcatl Favela Contreras  
Sinodal



Dr. Alex Elías Zúñiga  
Director del Programa Doctoral en  
Ciencias de Ingeniería

Diciembre, 2011

**COMBINING ARTIFICIAL INTELLIGENCE AND ROBUST  
TECHNIQUES WITH MRAC IN FAULT TOLERANT CONTROL**

BY:

Adriana Vargas Martínez

THESIS

Presented to the Doctorate Program in Engineering

This thesis is a partial requirement for the degree of

Doctor of Philosophy in

**Engineering Sciences**

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY

December 2011



## **ACKNOWLEDGEMENTS**

I am heartily thankful to my principal supervisor, Dr. Luis Eduardo Garza Castañón, whose guidance, encouragement and support allowed me to finish one of the most important achievements in my life. It has been an honor to be his Doctorate student. I truly appreciate all his contributions and time dedication to this project.

It is a pleasure also to thank to Dr. Vicenç Puig, Dr. Efraín Alcorta, Dr. Rubén Morales-Menéndez and Dr. Antonio Favela. This thesis would not have been possible without their support.

I would like to thank all my professors during my academic life because each one of them helps me to acquire the knowledge to develop this project.

I am grateful to the Tecnológico de Monterrey Campus Monterrey, because in this Institution I was able to realize all my academic dreams and this Institution has been my second home for the last eleven years.

In addition, I would like to thank my family, friends and colleagues for giving me their support in this long and challenging journey. I am thankful to my parents for their never-ending support and encouragement. My loving thank to my brother and family in law for their moral support. And most of all, I am grateful for my loving, encouraging, supportive and very patient husband César for his never-ending and faithful support. Thank you so much.

Adriana Vargas

To my father,

There are no words to express how much I admire you and love you.

## Index

<b>SUMMARY.....</b>	<b>1</b>
<b>1 INTRODUCTION.....</b>	<b>6</b>
1.1 INTRODUCTION TO FAULT TOLERANT CONTROL .....	6
1.2 CLASSIFICATION OF THE FAULT TOLERANT CONTROL METHODS.....	8
1.3 FAULT TOLERANT CONTROL SCHEMES .....	10
1.4 JUSTIFICATION.....	11
1.5 OBJECTIVE.....	12
1.6 HYPOTHESIS .....	12
<b>2 BACKGROUND THEORY .....</b>	<b>15</b>
2.1 MODEL REFERENCE ADAPTIVE CONTROL .....	15
2.1.1 MRAC based on the MIT rule.....	15
2.1.2 MRAC based on the Lyapunov stability theory .....	17
2.2 ARTIFICIAL NEURAL NETWORKS .....	19
2.3 GENETIC ALGORITHMS .....	22
2.4 SLIDING MODE CONTROL .....	23
2.5 LINEAR PARAMETER VARYING SYSTEMS .....	24
2.6 $H_\infty$ LOOP SHAPING CONTROLLER.....	25
2.7 LITERATURE REVIEW.....	27
2.8 FAULTS TYPES.....	33
<b>3 INDUSTRIAL HEAT EXCHANGER.....</b>	<b>35</b>
3.1 MODEL REFERENCE ADAPTIVE CONTROLLER (MRAC) .....	38
3.1.1 Experiments and Results .....	41
3.2 MODEL REFERENCE ADAPTIVE CONTROLLER PLUS PID CONTROLLER (MRAC-PID) .....	44
3.2.1 Experiments and Results .....	45
3.3 MODEL REFERENCE ADAPTIVE CONTROLLER PLUS AN ARTIFICIAL NEURAL NETWORK CONTROLLER (MRAC-ANN).....	48
3.3.1 Experiments and Results .....	49
3.4 MODEL REFERENCE ADAPTIVE CONTROLLER PLUS ARTIFICIAL NEURAL NETWORK AND PID CONTROLLERS (MRAC-ANN-PID).....	51
3.4.1 Experiments and Results .....	51
3.5 MODEL REFERENCE ADAPTIVE CONTROLLER PLUS AN $H_\infty$ CONTROLLER (MRAC- $H_\infty$ ) .....	54
3.5.1 Experiments and Results .....	58
3.6 MODEL REFERENCE ADAPTIVE CONTROL PLUS SLIDING MODE CONTROL (MRAC-SMC).....	61
3.6.1 Experiments and Results .....	61
3.7 MEAN SQUARE ERROR FOR THE EXPERIMENTS IN THE INDUSTRIAL HEAT EXCHANGER .....	64
3.8 MANIPULATED VARIABLE ANALYSIS FOR THE INDUSTRIAL HEAT EXCHANGER .....	65
<b>4 COUPLED-TANK SYSTEM.....</b>	<b>70</b>
4.1 MRAC-4 OPERATING POINTS LPV CONTROLLER .....	72
4.1.1 Experiments and Results .....	76
4.2 MRAC-ARTIFICIAL NEURAL NETWORK 4 OPERATING POINTS LPV CONTROLLER.....	79
4.2.1 Experiments and Results .....	80
4.3 MRAC- $H_\infty$ 4 OPERATING POINTS LPV CONTROLLER.....	83
4.3.1 Experiments and Results .....	84
4.3.2 Comparison between MRAC-4OP-LPV, MRAC-ANN4OP-LPV and the MRAC- $H_\infty$ 4OP-LPV based on MIT and based on Lyapunov theory .....	86
4.4 MRAC-LPV CONTROLLER .....	91
4.4.1 Experiments and Results .....	93
4.5 $H_\infty$ GAIN SCHEDULING -MRAC LPV CONTROLLER .....	97
4.5.1 Experiments and Results .....	105

4.5.2	<i>Comparison between the MRAC-LPV and the MRAC-<math>H_\infty</math>GS-LPV based on MIT and based on Lyapunov theory</i>	107
4.6	EXPERIMENTS USING THE MRAC MIT AND MRAC LYAPUNOV BASED DESIGN WITH THE NONLINEAR MODEL OF THE SYSTEM	109
4.7	EXPERIMENTS AND RESULTS USING THE COUPLED-TANK SYSTEM AS TESTBED IMPLEMENTING MULTIPLICATIVE FAULTS	116
4.7.1	<i>Multiplicative Faults applied in the LPV System</i>	116
4.7.2	<i>Multiplicative Faults applied in the Nonlinear System</i>	120
4.8	MANIPULATED VARIABLE ANALYSIS FOR THE COUPLED TANK SYSTEM	125
4.9	COMPARISON WITH SIMILAR APPROACHES	128
<b>5</b>	<b>STABILITY ANALYSIS</b>	<b>134</b>
5.1	LPV STABILITY ANALYSIS	134
5.1.1	<i>LPV Plant Open Loop Stability Analysis</i>	136
5.1.2	<i>Maximizing the Quadratic Stability Region of the LVP Plant</i>	138
5.2	STABILITY ASSUMPTIONS FOR THE MRAC CONTROLLER	141
5.2.1	<i>Lyapunov Stability Theorem for the Design of the MRAC Controller</i>	141
5.3	STABILITY ASSUMPTIONS FOR $H_\infty$ GAIN SCHEDULING CONTROLLER	145
<b>6</b>	<b>CONCLUSIONS</b>	<b>147</b>
6.1	INDUSTRIAL HEAT EXCHANGER	147
6.1.1	<i>Comparison between MRAC-ANN-PID, MRAC-<math>H_\infty</math>, MRAC-SMC, MRAC-ANN, MRAC-PID and MRAC</i>	147
6.2	COUPLED-TANK SYSTEM	147
6.2.1	<i>Comparison between MRAC-ANN4OP-LPV, MRAC-4OP-LPV and MRAC-<math>H_\infty</math>4OP-LPV based on the MIT rule and on Lyapunov theory</i>	147
6.2.2	<i>Comparison between MRAC-<math>H_\infty</math>GS-LPV and MRAC-LPV based on the MIT rule and on Lyapunov theory</i>	148
6.2.3	<i>Comparison between the MRAC controllers using the nonlinear model of the system</i>	149
<b>7</b>	<b>FUTURE WORK</b>	<b>152</b>
<b>8</b>	<b>REFERENCES</b>	<b>154</b>
	<b>APPENDIX A</b>	<b>164</b>
	<b>APPENDIX B</b>	<b>181</b>
	<b>APPENDIX C</b>	<b>193</b>
	<b>PUBLICATIONS</b>	<b>217</b>

## Index of Figures

Figure 1. Architecture for Fault Tolerant Autonomous Control Systems proposed by (Blanke, 1997). .....	7
Figure 2. Architecture for Fault-Adaptive Tolerant Control Technology proposed by (Karsai et al, 2003). ....	7
Figure 3. FTC classification approaches .....	13
Figure 4. Model Reference Adaptive Controller (MRAC) general scheme (Astrom and Wittenmark, 1995). ..15	
Figure 5. Model Reference Adaptive Controller based on MIT rule. ....	16
Figure 6. Model Reference Adaptive Controller based on Lyapunov theory. ....	19
Figure 7. Basic Artificial Neuron. ....	19
Figure 8. General Multi-layer Artificial Neural Network. ....	20
Figure 9. Levenberg-Maquard Backpropagation Algorithm Steps (Priddy and Keller, 2005). ....	21
Figure 10. Genetic Algorithm Steps. ....	23
Figure 11. General $H_{\infty}$ Controller Configuration (Zames, 1981). ....	25
Figure 12. Shaped plant and controller (Skogestad and Postlethwaite, 2005). ....	26
Figure 13. $H_{\infty}$ robust stabilization problem (Skogestad and Postlethwaite, 2005). ....	26
Figure 14. Multi-ANN faulty models FTC scheme (Gomaa, 2004). ....	28
Figure 15. Artificial Neural Network FTC scheme proposed by (Wang & Wang, 1999). ....	29
Figure 16. Representation of Additive and Multiplicative Faults (Mahmoud et al., 2003). ....	33
Figure 17. Representation of Actuator and Sensor Faults. ....	33
Figure 18. Industrial Heat Exchanger used in the experiments. ....	35
Figure 19. PRBS test for model estimation. ....	36
Figure 20. PRBS test for model validation. ....	36
Figure 21. MRAC Results testing different sizes of $\gamma$ . ....	40
Figure 22. Fault Tolerant MRAC Controller Structure. ....	41
Figure 23. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC controller. ....	42
Figure 24. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC controller. ....	43
Figure 25. Fault Tolerant MRAC-PID Controller structure. ....	45
Figure 26. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-PID controller. ....	46
Figure 27. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-PID controller. ....	47
Figure 28. Fault Tolerant MRAC-ANN Controller Structure. ....	48
Figure 29. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-ANN controller. ....	49
Figure 30. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-ANN controller. ....	50
Figure 31. Fault Tolerant MRAC-ANN-PID Controller Structure. ....	51
Figure 32. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-ANN-PID controller. ....	52
Figure 33. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-ANN controller. ....	53
Figure 34. $H_{\infty}$ Controller Singular Value plot for the steam process. ....	56
Figure 35. $H_{\infty}$ Controller Singular Value plot for the water process. ....	57
Figure 36. Fault Tolerant MRAC- $H_{\infty}$ Controller Structure. ....	58
Figure 37. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC- $H_{\infty}$ controller. ....	59
Figure 38. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC- $H_{\infty}$ controller. ....	60
Figure 39. Fault Tolerant MRAC-SMC Controller Structure. ....	61
Figure 40. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-SMC controller. ....	62
Figure 41. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-SMC controller. ....	63
Figure 42. Comparison between the manipulated variable and the system output of the MRAC scheme, applying an additive sensor fault of 10% at 5000 seconds. ....	65
Figure 43. Comparison between the manipulated variable and the system output of the MRAC scheme, applying an additive actuator fault of 10% at 5000 seconds. ....	66
Figure 44. Comparison between the manipulated variable and the system output of the MRAC-ANN-PID scheme, applying an additive sensor fault of 10% at 5000 seconds. ....	67
Figure 45. Comparison between the manipulated variable and the system output of the MRAC-ANN-PID scheme, applying an additive actuator fault of 10% at 5000 seconds. ....	68
Figure 46. Coupled-tank system designed by (Apkarian, 1999). ....	70
Figure 47. MRAC-4 Operating Points LPV Controller based on MIT rule. ....	73
Figure 48. MRAC-4 Operating Points LPV Controller based on Lyapunov theory. ....	74
Figure 49. MRAC Results testing different sizes of $\gamma$ for sensor faults. ....	75
Figure 50. MRAC Results testing different sizes of $\gamma$ for actuator faults. ....	75

Figure 51. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-sensor fault of 23.3% at the first operating point. ....	76
Figure 52. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-actuator fault of 1% at the first operating point. ....	77
Figure 53. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-sensor fault of 10% at the third operating point. ....	77
Figure 54. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-actuator fault of 1% at the third operating point. ....	78
Figure 55. MRAC-Artificial Neural Network 4 Operating Points LPV Controller based on MIT rule. ....	79
Figure 56. MRAC-Artificial Neural Network 4 Operating Points LPV Controller based on Lyapunov theory. ....	80
Figure 57. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-sensor fault of 23.3% at the first operating point. ....	80
Figure 58. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-actuator fault of 1% at the first operating point. ....	81
Figure 59. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-sensor fault of 10% at the third operating point. ....	81
Figure 60. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-actuator fault of 1% at the third operating point. ....	82
Figure 61. MRAC- $H_\infty$ 4 Operating Points LPV Controller based on MIT rule. ....	83
Figure 62. MRAC- $H_\infty$ 4 Operating Points LPV Controller based on Lyapunov theory. ....	84
Figure 63. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-sensor fault of 23.3% at the first operating point. ....	84
Figure 64. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-actuator fault of 1% at the first operating point. ....	85
Figure 65. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-sensor fault of 10% at the third operating point. ....	85
Figure 66. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-actuator fault of 1% at the third operating point. ....	86
Figure 67. MRAC + ANN + $H_\infty$ 4 Operating Points LPV Controllers based on MIT rule. ....	87
Figure 68. MRAC + ANN + $H_\infty$ 4 Operating Points LPV Controllers based on Lyapunov theory. ....	87
Figure 69. MRAC-LPV system based on MIT rule. ....	92
Figure 70. MRAC-LPV system based on Lyapunov theory. ....	92
Figure 71. MRAC-LPV Results testing different sizes of $\gamma$ . ....	93
Figure 72. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of magnitude 3.3% and an abrupt-actuator fault of magnitude 20% for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ . ....	94
Figure 73. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of magnitude 160% and an abrupt-actuator fault of magnitude 20% for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ . ....	94
Figure 74. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 3.3% and a gradual-actuator fault of 20% for the operating points $\varphi_1=0.6$ and $\varphi_2=0.6$ . ....	95
Figure 75. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 160% and a gradual-actuator fault of 20% for the operating points $\varphi_1=0.6$ and $\varphi_2=0.6$ . ....	95
Figure 76. Bode Diagram of the 4 Plants Multiplicative Uncertainties. ....	99
Figure 77. Bode Diagram of all Multiplicative Uncertainties. ....	99
Figure 78. Bode Diagram of the 4 Plants Additive Uncertainties. ....	100
Figure 79. Bode Diagram of all Additive Uncertainties. ....	101
Figure 80. Loop Shaping Structure. ....	102
Figure 81. $H_\infty$ Gain Scheduling – MRAC LPV Controller based on MIT rule. ....	104
Figure 82. $H_\infty$ Gain Scheduling – MRAC LPV Controller based on Lyapunov theory. ....	105
Figure 83. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of 3.3% and an abrupt-actuator fault of 20% for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ . ....	105

Figure 84. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of 160% and an abrupt-actuator fault of 20% for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ .	106
Figure 85. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 3.3% and a gradual-actuator fault of 20% for the operating points $\varphi_1=0.6$ and $\varphi_2=0.6$ .	106
Figure 86. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 160% and a gradual-actuator fault of 20% for the operating points $\varphi_1=0.6$ and $\varphi_2=0.6$ .	107
Figure 87. Nonlinear Process MRAC-LPV Controller based on MIT rule.	109
Figure 88. Nonlinear Process MRAC-LPV Controller based on Lyapunov theory.	110
Figure 89. Nonlinear Process MRAC- $H_\infty$ GS-LPV Controller based on MIT rule.	110
Figure 90. Nonlinear Process MRAC- $H_\infty$ GS-LPV Controller based on Lyapunov theory.	111
Figure 91. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with an abrupt-sensor fault of 3.3% and an abrupt-actuator fault of 20% for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ .	111
Figure 92. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with an abrupt-sensor fault of 166% and an abrupt-actuator fault of magnitude 20% for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ .	112
Figure 93. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with a gradual-sensor fault of 3.3% and a gradual-actuator fault of 20% for the operating points $\varphi_1=0.6$ and $\varphi_2=0.6$ .	113
Figure 94. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with a gradual-sensor fault of 166% and a gradual-actuator fault of 20% for the operating points $\varphi_1=0.6$ and $\varphi_2=0.6$ .	114
Figure 95. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 100%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the LPV system.	117
Figure 96. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the LPV system.	118
Figure 97. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 50%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the LPV system.	119
Figure 98. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the LPV system.	120
Figure 99. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 100%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the nonlinear system.	121
Figure 100. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the nonlinear system.	122
Figure 101. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 50%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the nonlinear system.	123
Figure 102. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time	

5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points $\varphi_1=0.3$ and $\varphi_2=0.5$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	124
Figure 103. Comparison between the manipulated variable and the system output of the MRAC-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) and an additive actuator fault (15000 seconds) of 10% in the operating points $\varphi_1=0.35$ and $\varphi_2=0.35$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	125
Figure 104. Comparison between the manipulated variable and the system output of the MRAC-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 5% and an additive actuator fault (15000 seconds) of 15% in the operating points $\varphi_1=0.35$ and $\varphi_2=0.35$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	125
Figure 105. Comparison between the manipulated variable and the system output of the MRAC-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 20% and an additive actuator fault (15000 seconds) of 5% in the operating points $\varphi_1=0.35$ and $\varphi_2=0.35$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	126
Figure 106. Comparison between the manipulated variable and the system output of the MRAC- $H_\infty$ GS-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) and an additive actuator fault (15000 seconds) of 10% in the operating points $\varphi_1=0.35$ and $\varphi_2=0.35$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	126
Figure 107. Comparison between the manipulated variable and the system output of the MRAC- $H_\infty$ GS-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 5% and an additive actuator fault (15000 seconds) of 15% in the operating points $\varphi_1=0.35$ and $\varphi_2=0.35$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	127
Figure 108. Comparison between the manipulated variable and the system output of the MRAC- $H_\infty$ GS-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 20% and an additive actuator fault (15000 seconds) of 5% in the operating points $\varphi_1=0.35$ and $\varphi_2=0.35$ and a change in the operating point at time 10000 seconds for the nonlinear system. ....	127
Figure 109. $H_\infty$ Gain Scheduling Controller representation. ....	145
Figure 110. FDI + MRAC FTC Scheme. ....	152
Figure 111. DPCA and Contribution Plots methodologies (Tudón et al. 2010). ....	153



## Index of Tables

Table 1. Genetic Algorithm Parameters .....	22
Table 2. Industrial Heat Exchanger Sensors/Transmitters Description. ....	35
Table 3. MSE of different sizes of $\gamma$ .....	39
Table 4. PID Controller Parameters Properties (O'Dwyer, 2009).....	44
Table 5. Matlab® Optimization Toolbox Parameters.....	44
Table 6. Obtained best PID Controller parameters using GA Optimization.....	44
Table 7. Mean Square Error for the Abrupt and Gradual Sensor Faults. ....	64
Table 8. Mean Square Error for the Abrupt and Gradual Actuator Faults.....	64
Table 9. Variable Definition.....	70
Table 10. Model Reference Adaptive Controller of the 4 Operating Points based on MIT rule. ....	72
Table 11. MSE of different sizes of $\gamma$ .....	74
Table 12. Results of experiments of the MRAC-4OP-LPV, MRAC-ANN4OP-LPV and MRAC- $H_\infty$ 4OP-LPV methodologies based on MIT rule. ....	88
Table 13. Results of experiments of the MRAC-4OP-LPV, MRAC-ANN4OP-LPV and MRAC- $H_\infty$ 4OP-LPV methodologies based on Lyapunov theory. ....	89
Table 14. MSE Results of the comparison between the MRAC-4OP-LPV, MRAC-ANN4OP-LPV and MRAC- $H_\infty$ 4OP-LPV based on MIT rule and based on Lyapunov theory. ....	90
Table 15. MSE of the MRAC-LPV results of different sizes of $\gamma$ .....	93
Table 16. Different Plants at the four and nominal operating points. ....	98
Table 17. Multiplicative Uncertainty of each Plant.....	98
Table 18. Additive Uncertainty of each Plant. ....	100
Table 19. Results of experiments of the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule.....	108
Table 20. Results of experiments of the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on Lyapunov theory.....	108
Table 21. MSE comparison between the MRAC-LPV and MRAC- $H_\infty$ GS-LPV MIT and Lyapunov based design. ....	108
Table 22. MSE Results of the comparison between the Nonlinear MRAC-LPV and the Nonlinear MRAC- $H_\infty$ GS-LPV MIT and Lyapunov based design. ....	115
Table 23. Results of experiments using multiplicative sensor (5000 seconds) and actuator (15000 seconds) faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for LPV systems.....	116
Table 24. Results of experiments using multiplicative sensor (5000 seconds) and actuator (15000 seconds) faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for nonlinear systems. ....	121
Table 24. Summary of comparison with similar approaches.....	130
Table 25. Resulting plants at the corners of the parameter box.....	136

**Table of Abbreviations**

FTC	Fault Tolerant Control
LPV	Linear Parameter Varying
LTI	Linear Time Invariant
AI	Artificial Intelligence
ANN	Artificial Neural Network
FL	Fuzzy Logic
GA	Genetic Algorithm
AMM	Admissible Model Matching
LMI	Linear Matrix Inequality
PDLM	Parameter-Dependent Lyapunov Matrix
SOP	Static Output Feedback
MRAC	Model Reference Adaptive Control
PID	Proportional Integral Derivative
SMC	Sliding Mode Control
MIT	Massachusetts Institute of Technology
$H_\infty$ GS	$H_\infty$ Gain Scheduling
FDI	Fault Detection and Isolation
FACT	Fault-Adaptive Control Technology
AFTC	Active Fault Tolerant Control
PCA	Principal Component Analysis
DPCA	Dynamic Principal Component Analysis
CA	Component Analysis
YJBK	Youla-Jabr-Bongiorno-Kucera
RLS	Recursive Least Square
PNN	Pseudolinear Neural Network
LFR	Lineal Fractional Representation
MPC	Model Predictive Control
SISO	Single Input Single Output
TDof	Two Degree of Freedom
PEM-FC	Polymer Electrolyt Membrane Fuel Cell
FSV <sub>1</sub>	Solenoid valve in the water inlet
TT <sub>1</sub>	Temperature transmitter of the water inlet
FV <sub>1</sub>	Pneumatic control valve in the water inlet
FT <sub>1</sub>	Flow transmitter in the water inlet
TT <sub>2</sub>	Temperature transmitter of the water outlet
FV <sub>2</sub>	Pneumatic control valve in the steam inlet
FT <sub>2</sub>	Flow transmitter of the steam inlet
FSV <sub>2</sub>	Solenoid valve in the steam inlet
PRBS	Pseudo Random Binary Sequence
MSE	Mean Square Error
CSTR	Continuous Stirred Tank Reactor
QAAC	Quaternion Based Adaptive Attitude Control
RBF	Radial Basis Function
VS-MRAC	Variable Structure Model Reference Adaptive Control
SVD	Singular Value Decomposition

**Table of Symbols**

$G_p$	Process Model
$G_r$	Reference Model
$y_{process}$	Process Output
$y_p$	Process Output
$y_{reference}$	Process Reference
$y_r$	Process Reference
$U$	Process Input
$u_c$	Controller Input
$E$	Error
$\theta_1$	Adaptive Feedforward Gain
$\theta_2$	Adaptive Feedback Gain
$\Gamma$	Speed of learning
$V$	Lyapunov function
$\Theta$	Parameters Varying over Time
$K$	$H_\infty$ controller
$P$	Generalized Plant
$W$	Exogenous Signals
$Z$	Error Signals
$V$	Measured Variables
$W_1$	Pre Compensators
$W_2$	Post Compensators
$G_s$	Shaped Plant
$\Delta_M, \Delta_N$	Uncertainties
$E$	Stability Margin
$GAM$	Precision Parameter
$G_{steam}$	Steam Model
$G_{water}$	Water Model
$F_{steam}$	Steam Flow
$F_{water}$	Water Flow
$L$	Open Loop
$T$	Complementary Sensitivity Function
$S$	Sensitivity Function
$G_{steam\ faults}$	Steam Faults Model
$G_{water\ faults}$	Water Faults Model
$h_1, h_2$	water level of tank 1 and tank 2
$A_1, A_2$	cross-section area of tank 1 and tank 2
$a_1, a_2$	cross-section area of the outflow orifice of tank 1 and tank 2
$U$	pump voltage
$k_p$	pump gain
$G$	gravitational constant
$\alpha_4, \alpha_3, \alpha_2, \alpha_1$	approximation constant
$\phi_1, \phi_2$	Parameter Dependent Operating Ranges
$G_{LPV\ model}$	LPV Model Plant
$p(t)$	Depend Parameter
$S(p)$	System Matix
$W_m$	Multiplicative Uncertainty
$W_a$	Additive Uncertainty
$uf$	System Output with Actuator Fault
$\alpha$	Degradation Percentage of the Actuator
$A(t), E(t)$	Parameter-Dependent Model

$V_s$	Set of Corners of the Parameter Box
$Q, M_i$	Symmetric Matrices
$\omega$	Corners
$u_i$	Interval Center
$\delta_i$	Interval Radius

# SUMMARY

The investigation of this thesis presents different approaches for Fault Tolerant Control based on Model Reference Adaptive Control, Artificial Neural Networks, PID controller optimized by a Genetic Algorithm, Nonlinear, Robust and Linear Parameter Varying (LPV) control for Linear Time Invariant (LTI), LPV and nonlinear systems. All of the above techniques are integrated in different controller's structures to prove their ability to accommodate a fault.

Modern systems and their challenging operating conditions in certain processes increase the possibility of system failures causing damages in equipment and/or their operators. In these environments, the use of automation control (i.e. adaptive and robust control) and intelligent systems is fundamental to minimize the impact of faults. Therefore, Fault Tolerant Control (FTC) methods have been proposed to ensure the continuous operations of system even in fault situation and to prevent more serious effects.

Until now, most of the FTC methods that have been developed are based on classical control theory (Yu et al., 2005; Zhang et al., 2007; Fradkov et al., 2008; Yang et al., 2008). The use of Artificial Intelligence (AI) in FTC has emerged recently (Stengel, 1991; Bastani & Chen, 1998; Patton et al., 1999; Korbiicz et al., 2004). Classical Artificial Intelligence (AI) approaches such as Artificial Neural Networks (ANN), Fuzzy Logic (FL), ANN-FL and Genetic Algorithms (GA) may offer some advantages over traditional methods (Schroder et al., 1998; Yu et al., 2005; Dong et al., 2006; Alves et al., 2009; Beainy et al., 2009; Kurihara, 2009; Li, 2009; Nieto et al., 2009; Panagi & Polycarpou, 2009) in the control community such as state observers, statistical analysis, parameter estimation, parity relations, residual generation, etc. The reasons are that AI approaches can reproduce the behavior of nonlinear dynamical systems with models extracted from data. Also, there are many learning processes that improve the FTC performance. This is a very important issue in FTC applications on automated processes, where information is easily available, or processes where accurate mathematical models are hard to obtain.

In the last years, FTC and control schemes based on LPV systems have been developed. In Bosche et al. (2009) a Fault Tolerant Control structure for vehicle dynamics is developed employing an LPV model with actuator failures. The methodology described in Bosche et al. (2009) paper is based on the resolution of Linear Matrix Inequalities (LMIs) using the DC-stability concept and a Parameter-Dependent Lyapunov Matrix (PDLM). In Montes de Oca et al. (2009), an Admissible Model Matching (AMM) FTC method based on LPV fault representation was presented; in this approach the faults were considered as scheduling variables in the LPV fault representation allowing the controller adaptation on-line. For instance, in Rodrigues et al. (2007) a FTC methodology for polytopic LPV systems was presented. The most important contribution of Rodrigues et al. (2007) work was the development of a Static Output Feedback (SOF) that maintains the system performance using an adequate controller reconfiguration when a fault appears.

On the other hand, advanced techniques from Robust Control such as  $H_\infty$ , have also been applied to FTC with encouraging results. For example, in Dong et al. (2009), an active FTC scheme for a class of linear time-delay systems, using a  $H_\infty$  controller in generalized internal mode architecture in combination with an adaptive observer-based fault estimator was presented. In Xiadong et al. (2008) a dynamic output feedback FTC approach that uses a  $H_\infty$  index for actuator continuous gain faults was proposed. And, in Liang & Duan (2004) a  $H_\infty$  FTC approach was used against sensor failures for uncertain descriptor system (systems which

capture the dynamical behavior of natural phenomena).

To improve the capabilities of the FTC systems mentioned above, different types of controller based on Adaptive Control, Artificial Neural Networks, Robust, Nonlinear and LPV Control for LTI, LPV and Nonlinear systems are proposed in this thesis. These controllers are first tested in an Industrial Heat Exchanger and then tested in a Coupled-Tank LPV System. Different types of faults are simulated in the implemented schemes: First, additive abrupt faults and gradual faults were introduced. In the abrupt fault case, the whole magnitude of the fault is developed in one moment of time and is simulated with a step function. On the other hand, gradual faults are developed during a period of time and are implemented with a ramp function. Second, multiplicative faults were tested. All types of faults, additive and multiplicative, can be implemented in sensors (feedback), in which the properties of the process are not affected, but the sensor readings are mistaken. And it also can be implemented in actuators (process entry) causing changes in the behavior of the process or interruption.

The controllers developed to test the Industrial Heat Exchanger are a Model Reference Adaptive Controller (MRAC), an MRAC with a PID controller whose parameters were optimized using a GA (MRAC-PID), an MRAC with an ANN (MRAC-ANN), an MRAC with a PID and an ANN (MRAC-ANN-PID), an MRAC with a Sliding Mode Controller (MRAC-SMC) and finally, an MRAC with an  $H_\infty$  control (MRAC- $H_\infty$ ). These MRAC controllers were design using the MIT rule. The controller with the best response against the faults is the MRAC-ANN-PID controller because was robust against the tested sensor and the actuator were imperceptible with almost a 0% error between the reference model and the process model.

For the Coupled-Tank LPV system, an MRAC (MRAC-4OP-LPV), an MRAC with an ANN (MRAC-ANN4OP-LPV) and an MRAC with an  $H_\infty$  controller (MRAC- $H_\infty$ 4OP-LPV) were designed for 4 operating points of the LPV system. For the sensor faults, the controller with the best results was the MRAC-NN4OP-LPV because it was fault tolerant against the tested sensor faults no matter the value of the operating point. This method resulted the best scheme because is a combination of two type of controllers, one is a Model Reference Adaptive Controller (MRAC) and the other one is an Artificial Neural Network designed to follow the ideal trajectory (non-faulty trajectory). For the actuator faults, the MRAC- $H_\infty$ 4OP-LPV was the best scheme because it was fault tolerant to the applied faults and also could accommodate the faults faster than the MRAC-4OP-LPV scheme.

In addition, for the Coupled-Tank system, an MRAC (MRAC-LPV) controller and an MRAC with an  $H_\infty$  Gain Scheduling controller (MRAC- $H_\infty$ GS-LPV) that work for all the operating points of the LPV system were developed. Both controllers were tested using the LPV system of the plant and also were tested using the nonlinear model of the system. In general, for additive and multiplicative faults, the MRAC- $H_\infty$ GS-LPV showed better results because is a combination of two type of LPV controllers, one is a Model Reference Adaptive Controller (MRAC) and the other one is a  $H_\infty$  Gain Scheduling Controller, both controllers were designed for an LPV system giving them the possibility of controlling any desired operating point between the operation range of the dependent variables ( $\varphi_1$  and  $\varphi_2$ ). In addition, the manipulated variable was plotted and it can be observe on this figure how the system compensates the fault.

The main contributions of this research are the development of the MRAC with an Artificial Neural Network and a PID controller optimized by a Genetic Algorithm (MRAC-ANN-PID) and the development of an MRAC with an  $H_\infty$  Gain Scheduling Controller that works for all the operating points of an LPV system (MRAC- $H_\infty$ GS-LPV). The MRAC-ANN-PID controller as mentioned above resulted to be robust against sensor and the actuator faults were imperceptible with a very low error between the reference model and the process. The PID parameters of this controller  $K_p$ ,  $K_i$  and  $K_d$  were optimized in order to follow the desired trajectory (no faulty system) and the ANN was trained also to follow the desired system trajectory no matter the fault size. The MRAC-ANN-PID controller is different from the controllers that already exist in the literature first because none of them had the controller structure of the MRAC-ANN-PID, second because most of them do not use any Artificial Intelligence methods such as ANN or GA. And third, in the literature, the ANN is used to represent or estimate the plant not as a controller which is the case of this research. On the other hand, for the MRAC- $H_\infty$ GS-LPV controller the main contribution was the development of a passive structure of FTC able to deal with abrupt and gradual faults in actuators and sensors of nonlinear processes represented by LPV models. This controller can accommodate the tested faults for any operating point between the operating ranges. The MRAC and the  $H_\infty$  Gain Scheduling controller were specially designed to switch from one operating point to another in less than a second. The MRAC controller was chosen as a FTC because guarantees asymptotic output tracking, it has a direct physical interpretation and it is easy to implement. The  $H_\infty$  Gain Scheduling Controller was also chosen because it increases the robust performance and stability of the closed loop system. In the existing literature, the  $H_\infty$  technique has been combined with other schemes to control systems but to the best of our knowledge there are no reports concerning the combination of an MRAC with an  $H_\infty$  Gain Scheduling controller.



# **CHAPTER 1**

## **INTRODUCTION**

# 1 Introduction

## 1.1 Introduction to Fault Tolerant Control

An increasing demand on products quality, system reliability, and plant availability has allowed that engineers and scientists give more attention to the design of methods and systems that can handle certain types of faults. In addition, the global crisis creates more competition between industries and plant shutdowns are not an option because they cause production losses and consequently lack of presence in the markets; primary services such as power grids, water supplies, transportation systems, and communication and commodities production cannot be interrupted without putting at risk human health and social stability.

On the other hand, modern systems and challenging operating conditions increase the possibility of system failures which can cause loss of human lives and equipments; also, some dangerous environments in places such as nuclear or chemical plants, set restrictive limits to human work. In all these environments, the use of automation and intelligent systems is fundamental to minimize the impact of faults.

The most important benefit of the Fault Tolerant Control (FTC) approach is that the plant continues operating in spite of a fault, no matter if the process has certain degradation in its performance. This strategy prevents that a fault develops into a more serious failure. In summary, the main advantages of implementing an FTC system are (Blanke et al., 1997):

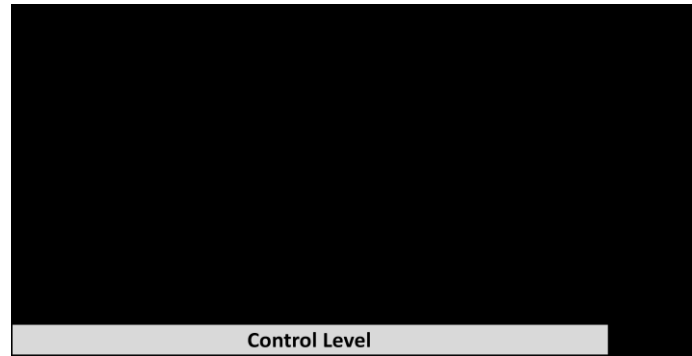
- Plant availability and system reliability in spite of the presence of a fault.
- Prevention to develop a single fault into a system failure.
- The use of information redundancy to detect faults instead of adding more hardware.
- The use of reconfiguration in the system components to accommodate a fault.
- FTC admits degraded performance due to a fault but maintains the system availability.
- FTC is not very expensive because most of the time no new hardware will be needed.

Some areas where FTC is being used more often are: aerospace systems, flight control, automotive engine systems and industrial processes. All of these systems have a complex structure and require a close supervision; FTC utilizes plant redundancy to create an intelligent system that can supervise the behavior of the plant components making these kinds of systems more reliable.

Since few years ago, emerging FTC techniques have been proposing new controller designs capable to tolerate system malfunctions and maintain stability and desirable performance properties. In order to achieve its objectives, two main tasks have to be considered on an active FTC system: fault detection and diagnosis and controller reconfiguration. The main purpose of fault detection and diagnosis is to detect, isolate and identify the fault, determining which faults affect the availability and safety of the plant. The controller reconfiguration task accommodates the fault and re-calculates the controller parameters in order to reduce the fault effects.

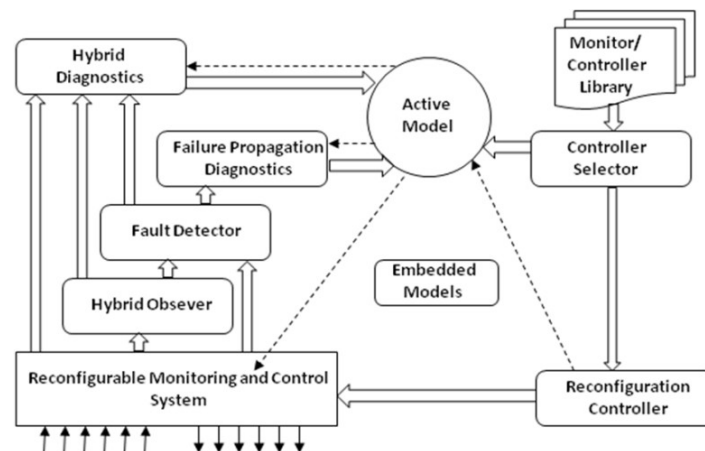
Although several schemes of FTC have been proposed, most of them are closely related to a general architecture. Blanke et al., (1997) introduces an approach for the design of an FTC system, shown in Figure 1, which included three operational levels: single sensor validation, fault detection and isolation using analytical

redundancy, and an autonomous supervision and reconfiguration system. The single sensor validation level involves the control loop with actuators, sensors, the controller and the signal conditioning and filtering. The second level (FDI) is composed of detectors and effectors that will perform the remedial actions. And finally, the supervision level deals with state-event logic in order to describe the logical state of controlled objects.



**Figure 1. Architecture for Fault Tolerant Autonomous Control Systems proposed by (Blanke, 1997).**

A slightly different architecture is presented in (Karsai et al., 2003). They introduce a scheme of Fault-Adaptive Control Technology (FACT), centered on model-based approaches for fault detection, fault isolation and estimation, and controller selection and reconfiguration for hybrid systems (see Figure 2). Hybrid models derived from hybrid bond graphs are used to model the continuous and discrete system dynamics. The supervisory controller, modeled as a generalized finite state automaton, generates the discrete events that cause reconfigurations in the continuous energy-based bond graph models of the plant. Fault detection involves a comparison between expected behaviors of the system, generated from the hybrid models, with actual system behavior.



**Figure 2. Architecture for Fault-Adaptive Tolerant Control Technology proposed by (Karsai et al., 2003).**

## 1.2 Classification of the Fault Tolerant Control Methods

Some authors have proposed different classifications for the FTC methods (Blanke et al., 2003; Eterno et al., 1985; Farrel et al., 1993; Lunze & Richter, 2006; Patton, 1997; Stengel, 1991). The classification shown in Figure 3 includes all the methods explained by these authors. We can also find a recent and very complete survey of FTC methods and applications in (Zhang & Jiang, 2008).

Regarding the design methods, fault tolerant control can be classified into two main approaches: active or passive. In Active Fault Tolerant Control (AFTC), if a fault occurs, the control system will be reconfigured using some properties of the original system in order to maintain an acceptable performance, stability and robustness. In some cases, degraded system operations have to be accepted (Blanke et al., 2001; Patton, 1997; Mahmoud et al., 2003). In Passive Fault Tolerant Control (PFTC), the system has a specific fixed controller to counteract the effect and to be robust against certain faults (Eterno et al., 1985).

To implement the AFTC approach two tasks are needed: fault detection and isolation and controller reconfiguration or accommodation. FDI means early detection, diagnosis, isolation, identification, classification and explanation of single and multiple faults; and can be accomplished by using the following three methodologies (Venkatasubramanian et al., 2003a, 2003b, 2003c):

- *Quantitative Model-Based* approaches require knowledge of the process model and dynamics in a mathematical structural form. Also, the process parameters, which are unknown, are calculated applying parameter estimation methods to measure inputs and outputs signals of the process. This approach uses analytical redundancy that can be obtained by implementing Kalman filters, observers and parity space.
- *Qualitative Model-Based* are based on the essential comprehension of the process physics and chemical properties. The model understanding is represented with quality functions placed in different parts of the process. This methodology can be divided in abstraction hierarchies and causal models. Abstraction hierarchies are based on decomposition and the model can establish inferences of the overall system behavior from the subsystem law behavior. This can be done using functional or structural approaches. Causal models take the causal system structure to represent the process relationships and are classified in diagraphs, fault trees and qualitative physics.
- *Process History-Based* approaches use a considerable amount of the process historical data and transform this data into a priori knowledge in order to understand the system dynamics. This data transformation is done using qualitative or quantitative methods. The quantitative methods are divided in expert systems (solve problems using expertise domain) and trend modeling (represent only significant events to understand the process). Quantitative methods can be statistical (use PCA, DPCA, CA) and non statistical (Artificial Neural Networks) to recognize and classify the problem.

After the detection and isolation of the fault, a controller reconfiguration or accommodation is needed. In controller accommodation, when a fault appears, the variables that are measured and manipulated by the controller continue unaffected, but the dynamic structure and parameters of the controller change

(Blanke et al., 2003). The fault will be accommodated only if the control objective with a control law that involves the parameters and structure of the faulty system has a solution (Blanke et al., 2001). In order to achieve fault accommodation, two approaches can be used: adaptive control and switched control. Adaptive control means to modify the controller control law to handle the situation where the system's parameters are changing over time. It does not need a priori information about the parameters limits. The goal is to minimize the error between the actual behavior of the system and the desirable behavior. On the other hand, switched control is determined by a bank of controllers designed for specific purposes (normal operation or fault) that switch from one to another in order to control a specific situation (Lunze & Richter, 2006).

Meanwhile, controller reconfiguration is related with changing the structure of the controller, the manipulated and the measured variables when a fault occurs (Steffen, 2005). This is achieved by using the following techniques:

- *Controller Redesign*. The controller changes when a fault occurs in order to continue achieving its objective (Blanke et al., 2003). This can be done by using several approaches: pseudo inverse methods (modified pseudo inverse method, admissible pseudo inverse method), model following (adaptive model following, perfect model following, eigen structure assignment) and optimization (linear quadratic design, model predictive control) (Caglayan et al., 1988; Gao & Antsaklis, 1991; Jiang, 1994; Lunze & Richter, 2006; Staroswiecki, 2005).
- *Fault Hiding Methods*. The controller continues unchanged when a fault is placed, because a reconfiguration system hides the fault from the controller. This method can be realized using virtual actuators or virtual sensors. (Lunze & Richter, 2006; Steffen, 2005).
- *Projection Based Methods*. A controller is designed a priori for every specific fault situation and replaces the nominal controller if that specific fault occurs. This can be done by a bank of controllers and a bank of observers (Mahmoud et al., 2003).
- *Learning Control*. This methodology uses artificial intelligence like ANN, fuzzy logic, genetic algorithms, expert systems and hybrid systems which can learn to detect, identify and accommodate the fault (Polycarpou & Vemuri, 1995; Stengel, 1991; Karsai et al, 2003).
- *Physical Redundancy*. This is an expensive approach because it uses hardware redundancy (multiple sensor or actuators) and decision logic to correct a fault because it switches the faulty component to a new one. An example of this is the voting scheme method (Isermann et al., 2002; Mahmoud et al., 2003).

On the other hand, passive FTC is based on robust control. In this technique, an established controller with constant parameters is designed to correct a specific fault to guarantee stability and performance (Lunze & Richter, 2006). There is no need for online fault information. The control objectives of robust control are: stability, tracking, disturbance rejection, sensor noise rejection, rejection of actuator saturation and robustness (Skogestad & Postlethwaite, 2005). Robust control involves the following methodologies:

- *H<sub>∞</sub> controller*. This type of controller deals with the minimization of the H<sub>∞</sub>-norm in order to optimize the worst case of performance specifications. In Fault Tolerant Control, it can be used as an index to represent the attenuation of the disturbances performances in a closed loop system (Yang & Ye, 2006) or can be used for the design of robust and stable dynamical compensators (Jaimoukha et al., 2006; Liang & Duan, 2004).
- *Linear Matrix Inequalities (LMIs)*. In this case, convex optimization problems are solved with precise matrices constraints. In Fault Tolerant Control, they are used to achieve robustness against actuator and sensor faults. (Zhang et al., 2007).
- *Simultaneous Stabilization*. In this approach multiple plants must achieve stability using the same controller in the presence of faults. (Blondel, 1994).
- *Youla-Jabr-Bongiorno-Kucera (YJBK) parameterization*. This methodology is implemented in Fault Tolerant Control to parameterize stabilizing controllers in order to guarantee system stability. YJBK in summary is a representation of the feedback controllers that stabilize a given system (Neimann & Stoustrup, 2005).

### 1.3 Fault Tolerant Control Schemes

Although, most of the FTC methods that have been developed are based on classical control theory (Yu et al., 2005; Zhang et al., 2007; Fradkov et al., 2008; Yang et al., 2008), the use of Artificial Intelligence (AI) in FTC has emerged recently (Stengel, 1991; Bastani & Chen, 1998; Patton et al., 1999; Korbicz et al., 2004). Classical Artificial Intelligence (AI) approaches such as Artificial Neural Networks (ANN), Fuzzy Logic (FL), ANN-FL and Genetic Algorithms (GA) may offer advantages over traditional methods (Schroder et al., 1998; Yu et al., 2005; Dong et al., 2006; Alves et al., 2009; Beainy et al., 2009; Kurihara, 2009; Li, 2009; Nieto et al., 2009; Panagi & Polycarpou, 2009) in the control community such as state observers, statistical analysis, parameter estimation, parity relations, residual generation, etc. The reasons are that AI approaches can reproduce the behavior of nonlinear dynamical systems with models extracted from data. Also, there are many learning processes that improve the FTC performance. This is a very important issue in FTC applications on automated processes, where information is easily available, or processes where accurate mathematical models are hard to obtain.

ANN have been applied to FTC because they are helpful to identify, detect and accommodate system faults. The application of ANN to FTC can be divided in three groups. The first group includes ANN used as fault detectors by estimating changes in process models dynamics (Polycarpou & Helmicki, 1997; Patton et al., 1999; Polycarpou, 2001; Gomma, 2004). The second group includes ANN used as controllers (Wang & Wang, 1999; Pashikar et al., 2006), and the third group integrates ANN which performs both functions: fault detection, and control (Perhinschi et al., 2007; Yen & De Lima, 2005; Patan & Korbicz, 2009; Yu et al., 2009). In addition, Genetic Algorithms have been applied to fault tolerant control as a strategy to optimize and supervise the controlled system in order to accommodate system failures (Schroder et al., 1998; Sugawara et al., 2003). On the other hand, advanced techniques from Robust Control such as H<sub>∞</sub>, and Non-linear

Control as Sliding Mode Control, have also been applied to FTC with encouraging results. For example, in (Liang & Duan, 2004), an  $H_\infty$  FTC approach was used against sensor failures for uncertain descriptor system (systems which capture the dynamical behavior of natural phenomena). In (Xiadong et al., 2008), a dynamic output feedback FTC approach that uses an  $H_\infty$  index for actuator continuous gain faults was proposed. In (Dong et al. 2009), an active FTC scheme for a class of linear time-delay systems was presented, using an  $H_\infty$  controller in generalized internal mode architecture in combination with an adaptive observer-based fault estimator. Besides, in (Weidong & Shaocheng, 2007), a Fuzzy Adaptive Sliding Mode FTC for SISO nonlinear systems was developed. This methodology uses a corrective control law (SMC) when a fault occurs while the fuzzy system learns the unknown system and fault function dynamics and determines the FTC law. In (Yen & Ho, 2000), a FTC system that includes a Sliding Mode control and an Artificial Neural Network was presented. This system is able to control the desired trajectories tracking problems when a fault is present.

In the last years, FTC schemes based on LPV systems and control methods have been developed. For instance, in (Rodrigues et al., 2007) a FTC methodology for polytopic LPV systems was presented. The most important contribution of this work was the development of a Static Output Feedback (SOF) that maintains the system performance using an adequate controller reconfiguration when a fault appears. In (Bosche et al., 2009), a Fault Tolerant Control structure for vehicle dynamics is developed employing an LPV model with actuator failures. The methodology described in this research is based on the resolution of Linear Matrix Inequalities (LMIs) using the DC-stability concept and a Parameter-Dependent Lyapunov Matrix (PDLM). In (Montes de Oca et al., 2009), an Admissible Model Matching (AMM) FTC method based on LPV fault representation was presented. In this approach, the faults were considered as scheduling variables in the LPV fault representation allowing the controller adaptation on-line. Although several applications have used LPV systems theory to develop FTC schemes (Rodrigues et al., 2007; Bosche et al., 2009; Luzar et al., 2009) and also MRAC-based approaches for FTC have been explored (Abdullah & Zribi, 2009; Cho et al., 1990; Ahmed, 2000; Thanapalan et al., 2006; Yu, 2004; Miyasato, 2006), none of them integrates the methodologies proposed in this thesis (i.e. MRAC, LPVs and  $H_\infty$ ).

These investigation present different approaches for FTC based on Model Reference Adaptive Control, Artificial Neural Networks, PID controller optimized by a Genetic Algorithm, Nonlinear, Robust and LPV control. The investigation is organized as follows: Chapter 2 describes the background theory; in Chapter 3 the experiments and schemes implemented in the Industrial Heat Exchanger are developed; Chapter 4 the experiments and schemes implemented in the Coupled-Tank System are presented, in Chapter 5 the Stability Analysis is demonstrated, Chapter 6 addresses the conclusions and Chapter 7 proposed the future work.

## 1.4 Justification

Nowadays the complexity of modern production systems and processes has created the necessity to incorporate Fault Tolerant Control systems, because FTC can ensure the operability of the systems even if a fault or faults appear. For this reason, this thesis presents different combinations of FTC structures based on a

Model Reference Adaptive Controller combined with Artificial Neural Networks, PID controller optimized by a Genetic Algorithm, Robust and Linear Parameter Varying (LPV) control for Linear Time Invariant (LTI) and LPV systems.

The above mentioned structures were chosen first, because the adaptive control avoids explicit modeling, decision making and control redesign (Thanapalan et al., 2006). Therefore, the MRAC controller guarantees asymptotic output tracking, has an inherent capacity to accommodate perturbations and faults, has direct physical interpretation and it is easy to implement (Sang & Tao, 2009). But, the use of only this type of controller has certain limitations in the FTC systems (see Section 5.1 and 5.2 of Chapter 5). For this reason, it is normal to find in the literature combinations of the MRAC controller with other structures in order to guarantee the system performance in the presence of a fault (Cho et al., 1990; Miyasato, 2008), to isolate and determinate faults (Thanapalan, 2006), to reduce the unknown model dynamics, the disturbances and parameter variations (Hongjie & Bo, 2008), to have a nicer transient behavior, disturbance rejection capability (Hsu, 1990), etc.

From the above discussion, this investigation presents the incorporation of Artificial Neural Networks to an MRAC controller in order to help the system to maintain the desired system trajectory when a fault appears. On the other hand, the use of a Genetic Algorithm in the design of the PID controller helps to choose the optimal PID parameters. In addition, the incorporation to LPV controllers such the  $H_\infty$  Gain Scheduling Controller to the MRAC controller adds controller robustness because the LPV controller is able to control any operating point between the systems operating range. Also, the use of Robust Control techniques increases robust performance and stability to the closed loop system.

## **1.5 Objective**

The objective of this thesis is to present different approaches for Fault Tolerant Control based on Model Reference Adaptive Control, Artificial Neural Networks, PID controller optimized by a Genetic Algorithm, Robust and Linear Parameter Varying (LPV) control for Linear Time Invariant (LTI), LPV and nonlinear systems. All of the above techniques are integrated in different controller's structures to prove their ability to accommodate a fault.

## **1.6 Hypothesis**

The different Fault Tolerant Control approaches based on Model Reference Adaptive Control in combination with Artificial Neural Networks, PID controller optimized by a Genetic Algorithm, Robust and Linear Parameter Varying (LPV) control for Linear Time Invariant (LTI) and LPV systems are developed in this investigation because these approaches will improve the capabilities of the Fault Tolerant Control systems such as robustness, fault accommodation, stability, etc.



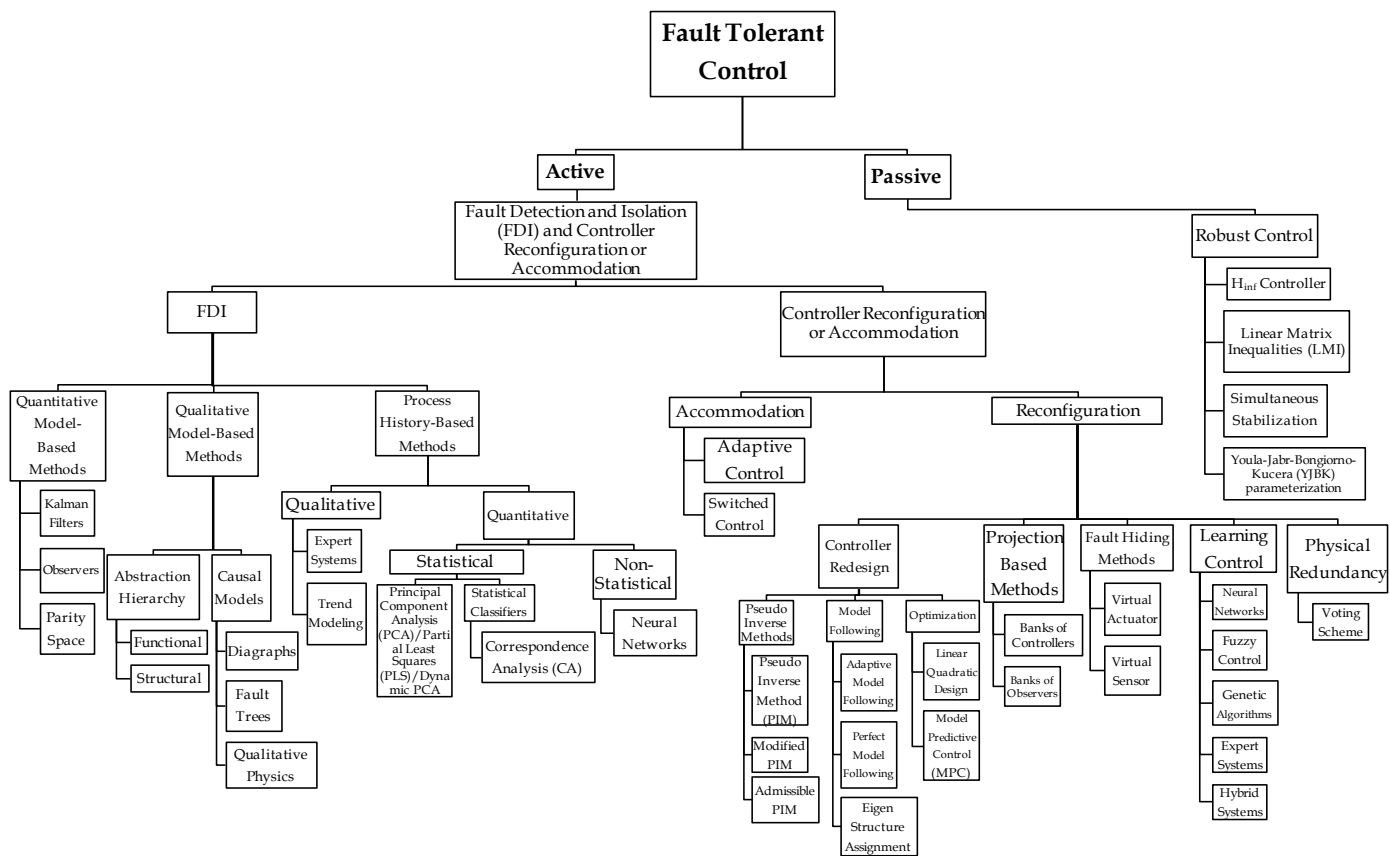


Figure 3. FTC classification approaches

# **CHAPTER 2**

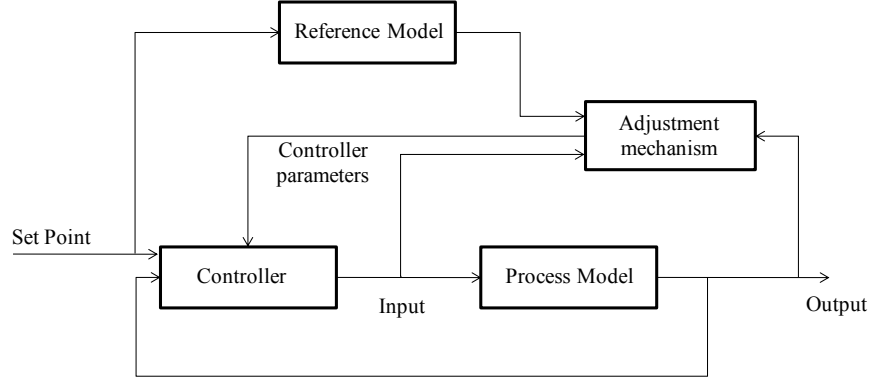
## **BACKGROUND**

### **THEORY**

## 2 Background Theory

### 2.1 Model Reference Adaptive Control

The MRAC, shown in Figure 4, implements a closed-loop controller that involves the parameters that should be optimized, in order to modify the system response to achieve the desired final value. The adaptation mechanism adjusts the controller parameters to match the process output with the reference model output. The reference model is specified as the ideal model behavior that the system is expected to follow.



**Figure 4. Model Reference Adaptive Controller (MRAC) general scheme (Astrom & Wittenmark, 1995).**

#### 2.1.1 MRAC based on the MIT rule

The mathematical procedure to design the MRAC system based on MIT rule (Figure 5) is the following. First, equation (1) should be transformed in order to include the Process Model and the Reference Model with their respective inputs (Astrom & Wittenmark, 1995):

$$e = y_{process} - y_{reference} = G_p * u - G_r * u_c \quad (1)$$

where  $e$ ,  $y_{process}$ ,  $y_{reference}$ ,  $u$  and  $u_c$  represent the error, process output, reference output, process input and controller input, respectively. To reduce the error, an adaptation mechanism that aims at reducing a cost function is used, in the form of:

$$J(\theta) = 1/2 e^2(\theta) \quad (2)$$

where  $\theta$  is the adaptive parameter inside the controller. For a second order system, the implemented MRAC scheme has two adaptation parameters: adaptive feedforward gain ( $\theta_1$ ) and adaptive feedback gain ( $\theta_2$ ). These parameters will be updated to follow the reference model. Then, the input is rewritten in terms of the adaptive feedforward and adaptive feedback gains as follows:

$$u = \theta_1 u_c - \theta_2 y_{process} \quad (3)$$

Such the process output can be expressed as:

$$y_{process} = G_p * u = \left( \frac{b_r}{s^2 + a_{1r}s + a_{0r}} \right) (\theta_1 u_c - \theta_2 y_{process}) = \left( \frac{b_r \theta_1}{s^2 + a_{1r}s + a_{0r} + b_r \theta_2} \right) u_c \quad (4)$$

Using equation (4), the error can be redefined as:

$$e = \left( \frac{b_r \theta_1}{s^2 + a_{1r}s + a_{0r} + b_r \theta_2} \right) u_c - (G_r * u_c) \quad (5)$$

Therefore, equation (2) can be minimized if the parameters  $\theta$  change in the negative direction of the gradient  $J$ , this is called the gradient descent method and is represented by:

$$d\theta/dt = -\gamma \partial J / \partial \theta = -\gamma \partial e / \partial \theta e \quad (6)$$

where  $\gamma$  is the parameter to adjust the speed of learning. The above equation is known as the MIT rule and determines how the parameter  $\theta$  will be updated to reduce the error (Whitaker, 1958). The error partial derivatives with respect to the adaptive feedforward ( $\theta_1$ ) and adaptive feedback ( $\theta_2$ ) gains are specified as:

$$\frac{\partial e}{\partial \theta_1} = \left( \frac{b_r}{s^2 + a_{1r}s + a_{0r} + b_r \theta_2} \right) u_c \quad (7)$$

$$\frac{\partial e}{\partial \theta_2} = \left( \frac{b_r^2 \theta_1}{(s^2 + a_{1r}s + a_{0r} + b_r \theta_2)^2} \right) u_c = - \left( \frac{b_r \theta_1}{s^2 + a_{1r}s + a_{0r} + b_r \theta_2} \right) y_{process} \quad (8)$$

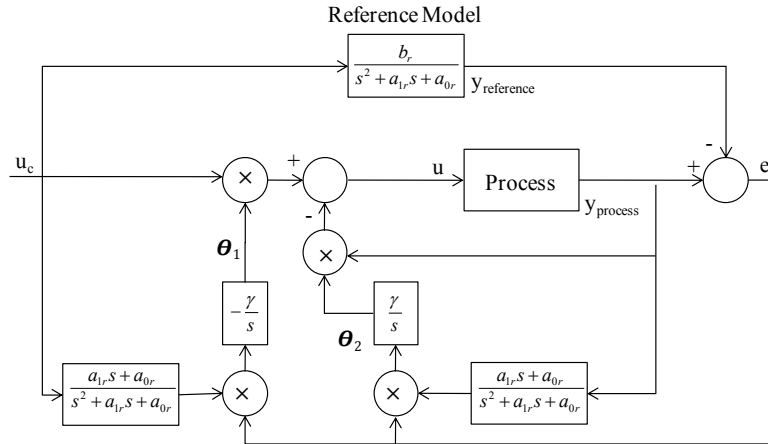
Consequently, the Process characteristic equation can be transformed into equation (9), because the MRAC system aim is to approximate the Process Model with the Reference Model.

$$s^2 + a_{1r}s + a_{0r} + b_r \theta_2 \approx s^2 + a_{1r}s + a_{0r} \quad (9)$$

Finally, with equation (9) defined, the error partial derivatives are transformed; and employing the MIT rule, the update rules for the adaptive feedforward ( $\theta_1$ ) and adaptive feedback ( $\theta_2$ ) gains are written. This can be viewed in equation (10) and (11).

$$\frac{\partial e}{\partial \theta_1} = \left( \frac{a_{1r}s + a_{0r}}{s^2 + a_{1r}s + a_{0r}} \right) u_c \rightarrow \frac{d\theta_1}{dt} = -\gamma \frac{\partial e}{\partial \theta_1} e = -\gamma \left( \frac{a_{1r}s + a_{0r}}{s^2 + a_{1r}s + a_{0r}} u_c \right) e \quad (10)$$

$$\frac{\partial e}{\partial \theta_2} = - \left( \frac{a_{1r}s + a_{0r}}{s^2 + a_{1r}s + a_{0r}} \right) y_{process} \rightarrow \frac{d\theta_2}{dt} = -\gamma \frac{\partial e}{\partial \theta_2} e = \gamma \left( \frac{a_{1r}s + a_{0r}}{s^2 + a_{1r}s + a_{0r}} y_{process} \right) e \quad (11)$$



**Figure 5. Model Reference Adaptive Controller based on MIT rule.**

### 2.1.2 MRAC based on the Lyapunov stability theory

The Lyapunov theory in the design of an MRAC controller was introduced because the MIT rule does not guarantee the stability of the closed-loop system. To design an MRAC controller using Lyapunov theory (see Figure 6), the first step is to derive a differential equation for the error that contains the adaptation parameters. Then, a Lyapunov function and an adaptation mechanism need to be established to reduce the error to zero. The Lyapunov derivative function  $dV/dt$  is usually negative semidefinite. Therefore, to determine the parameter convergence is necessary to establish persistently excitation and uniform observability on the system and the reference signal (Astrom & Wittenmark, 1995).

The Lyapunov stability theorem establishes the following: If there exists a function  $V: \mathbb{R}^n \rightarrow \mathbb{R}$  being positive definite and its derivative:

$$dV/dt = \partial V^T / \partial x \, dx/dt = \partial V^T / \partial x \, f(x) = -W(x) \quad (12)$$

is negative semidefinite, then the solution  $x(t)=0$  to

$$dx/dt = f(x) \quad f(0) = 0 \quad (13)$$

is stable. If  $dV/dt$  is negative definite the solution will be asymptotically stable.  $V$  denotes the Lyapunov function for the system. If:

$$dV/dt < 0 \text{ and } V(x) \rightarrow \infty \text{ when } \|x\| \rightarrow \infty \quad (14)$$

the solution is globally asymptotically stable. Therefore, the following procedure was realized:

Process model:

$$\ddot{y}_{process} + a_1 \dot{y}_{process} + a_0 y_{process} = bu \quad (15)$$

Reference model:

$$\ddot{y}_{reference} + a_{1r} \dot{y}_{reference} + a_{0r} y_{reference} = b_r u_c \quad (16)$$

Control law:

$$u = \theta_1 u_c - \theta_2 y_{process} \quad (17)$$

Error:

$$e = y_{process} - y_{reference} \quad (18)$$

Then, the error dynamics is represented by:

$$(19)$$

$$\dot{e} = \dot{y}_{process} - \dot{y}_{reference} = \frac{1}{a_1} [bu - \ddot{y}_{process} - a_0 y_{process}] - \frac{1}{a_{1r}} [b_r u_c - \ddot{y}_{reference} - a_{0r} y_{reference}]$$

To simplify the mathematical notation  $y_{reference} = y_r$  and  $y_{process} = y_p$ .

$$\dot{e} = \dot{y}_p - \dot{y}_r = \frac{1}{a_1} [bu - \ddot{y}_p - a_0 y_p] - \frac{1}{a_{1r}} [b_r u_c - \ddot{y}_r - a_{0r} y_r] \quad (20)$$

Substituting  $y_r = y_p - e$  and  $\ddot{y}_r = \ddot{y}_p - \ddot{e}$  from equation 20, equation 21 is obtained:

$$\dot{e} = \frac{1}{a_1} bu - \frac{1}{a_1} \ddot{y}_p - \frac{a_0}{a_1} y_p - \frac{1}{a_{1r}} b_r u_c + \frac{1}{a_{1r}} \ddot{y}_p - \frac{1}{a_{1r}} \ddot{e} + \frac{a_{0r}}{a_{1r}} y_p - \frac{a_{0r}}{a_{1r}} e \quad (21)$$

Replacing  $u = \theta_1 u_c - \theta_2 y_p$  in the above equation and placing the error terms in the left side of the equation, equation 22 is obtained:

$$\frac{1}{a_{1r}}\ddot{e} + \dot{e} + \frac{a_{0r}}{a_{1r}}e = \frac{1}{a_1}b\theta_1u_c - \frac{1}{a_{1r}}b_ru_c - \frac{1}{a_1}b\theta_2y_p - \frac{a_0}{a_1}y_p + \frac{a_{0r}}{a_{1r}}y_p - \frac{1}{a_1}\ddot{y}_p + \frac{1}{a_{1r}}\ddot{y}_p \quad (22)$$

The Reference Model is equal to the Process Model if no fault occurs ( $a_1=a_{1r}$ ,  $a_0=a_{0r}$ , and  $b=b_r$ ), then:

$$\frac{1}{a_{1r}}\ddot{e} + \dot{e} + \frac{a_{0r}}{a_{1r}}e = \frac{1}{a_{1r}}b_r\theta_1u_c - \frac{1}{a_{1r}}b_ru_c - \frac{1}{a_{1r}}b_r\theta_2y_p - \frac{a_{0r}}{a_{1r}}y_p + \frac{a_{0r}}{a_{1r}}y_p - \frac{1}{a_{1r}}\ddot{y}_p + \frac{1}{a_{1r}}\ddot{y}_p \quad (23)$$

$$\frac{1}{a_{1r}}\ddot{e} + \dot{e} + \frac{a_{0r}}{a_{1r}}e = \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p \quad (24)$$

$$\frac{1}{a_{1r}}\frac{d^2e}{dt^2} + \frac{de}{dt} + \frac{a_{0r}}{a_{1r}}e = \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p \quad (25)$$

$$\frac{de}{dt} = -\frac{1}{a_{1r}}\frac{d^2e}{dt^2} - \frac{a_{0r}}{a_{1r}}e + \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p \quad (26)$$

The proposed Lyapunov function is quadratic in tracking error and controller parameter estimation error since it is expected that the adaptation mechanism will drive both types of errors to zero. From the equation error dynamics (see equation 26) the proposed Lyapunov function is:

$$V(e, \theta_1, \theta_2) = \frac{1}{2}(a_{1r}e^2 + \frac{1}{\gamma b_r}(b_r\theta_1 - b_r)^2 + \frac{1}{\gamma b_r}(b_r\theta_2)^2) \quad (27)$$

where  $b_r$ ,  $\gamma$  and  $a_{1r}>0$ . Equation 27 will be zero when the error is zero and the controller parameters are equal to the desired values. The above Lyapunov function is valid if the derivative of this function is negative. Thus, the derivative of equation 27 is:

$$\dot{V} = a_{1r}e\frac{de}{dt} + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (28)$$

Substituting equation 26 in the above equation, and rearranging the similar terms, equation 29 is obtained.

$$\dot{V} = a_{1r}e\left(-\frac{1}{a_{1r}}\frac{d^2e}{dt^2} - \frac{a_{0r}}{a_{1r}}e + \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p\right) + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (29)$$

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 + (b_r\theta_1 - b_r)u_c e - (b_r\theta_2)y_p e + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (30)$$

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 + (b_r\theta_1 - b_r)u_c e + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} - (b_r\theta_2)y_p e + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (31)$$

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 + \frac{1}{\gamma}(b_r\theta_1 - b_r)\left(\frac{d\theta_1}{dt} + \gamma u_c e\right) + \frac{1}{\gamma}(b_r\theta_2)\left(\frac{d\theta_2}{dt} - \gamma y_p e\right) \quad (32)$$

Therefore, the adaptation parameters are selected to be updated as:

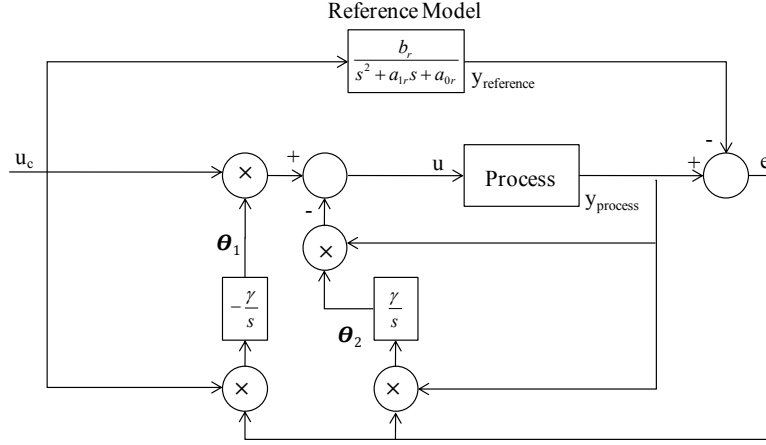
$$\frac{d\theta_1}{dt} = -\gamma u_c e \quad (33)$$

$$\frac{d\theta_2}{dt} = \gamma y_p e \quad (34)$$

Then

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 \quad (35)$$

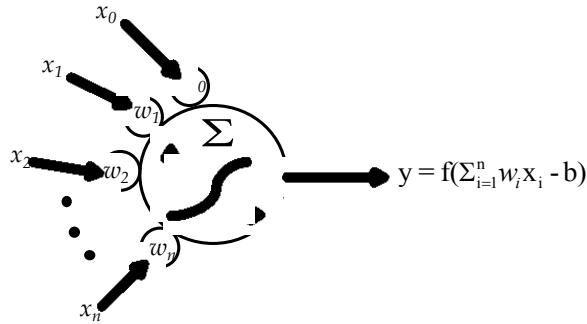
It can be seen that equation 35 is negative semidefinite which implies  $V(t) \leq V(0)$ . This ensures that  $e$ ,  $\theta_1$  and  $\theta_2$  are bounded. Since  $a_{1r} > 0$ ,  $a_{0r} > 0$  and  $u_c$  is bounded then  $y_r$  is bounded and therefore  $y_p = e + y_r$  is bounded as well. From the boundedness and convergence set theorem it can be concluded that the error  $e$  will go to zero (Astrom & Wittenmark, 1995).



**Figure 6. Model Reference Adaptive Controller based on Lyapunov theory.**

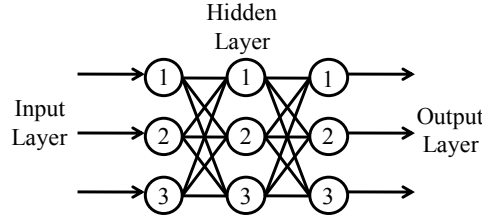
## 2.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are mathematical models that try to mimic the biological nervous system. An artificial neuron have multiple input signals  $x_1, x_2, \dots, x_n$  entering the neuron using connection links with specific weights  $w_1, w_2, \dots, w_n$  or  $\sum_{i=1}^n w_i x_i$  named the net input, and also have a firing threshold  $b$ , an activation function  $f$  and an output of the neuron that is represented by  $y = f(\sum_{i=1}^n w_i x_i - b)$ . The firing threshold  $b$  or bias can be represented as another weight by placing an extra input node  $x_0$  that takes a value of 1 and has a  $w_0 = -b$  (Nguyen et al., 2002). This can be represented in Figure 7.



**Figure 7. Basic Artificial Neuron.**

An ANN with more than one input layer of neurons, a middle layer called the hidden layer and an output layer is named a Multi-layer Artificial Neural Network (Figure 8).



**Figure 8. General Multi-layer Artificial Neural Network.**

An ANN can have a feedback or a feed-forward structure. In the feedback structure, the information can move back and forward. In the feed-forward structure, the information moves only forward from the input nodes through the outputs nodes with no cycles in the network (Ruan, 1997).

The ANN need to be trained from examples, in a process called supervised learning. Once a successfully training is done, the Artificial Neural Network is ready if and only if the network reproduces the desired outputs from the given inputs. The most common methodology for this kind of learning is the backpropagation algorithm, where the weights of the Artificial Neural Network are determined by using iteration until the output of the network is the same as the desired output (Rumerhart et al., 1986). In addition, unsupervised learning uses a mechanism for changing values of the weights according to the input values, this mechanism is named self-organization. An example of this algorithm is the Hebbian learning algorithm (Ruan, 1997).

To create and train the Artificial Neural Network controller, the original process inputs were introduced as well as the desired outputs. The created ANN is a two-layer feed forward neural network with  $n$  number of sigmoid hidden neurons and a linear output neuron. To train the network the Levenberg-Maquard backpropagation algorithm was used. This training algorithm is a combination of Gauss-Newton and gradient descent methods which integrates the benefits of the global and local convergence properties from the gradient descent and Gauss-Newton methods, respectively (Ye, 2004). The Levenberg-Marquardt method approaches the Hessian matrix in the form of the product of a Jacobian matrix by its transpose, the same form as the quasi-Newton Methods (Priddy & Keller, 2005; Hagan & Menhaj, 1994).

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial x_1} & \frac{\partial e_1}{\partial x_2} & \dots & \frac{\partial e_1}{\partial x_N} \\ \frac{\partial e_2}{\partial x_1} & \frac{\partial e_2}{\partial x_2} & \dots & \frac{\partial e_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_n}{\partial x_1} & \frac{\partial e_n}{\partial x_2} & \dots & \frac{\partial e_n}{\partial x_N} \end{bmatrix} \quad (36)$$

$$H \approx J^T J \quad (37)$$

Then, the gradient can be estimated as the product of the transpose Jacobian matrix by a vector which contains the minimized errors.

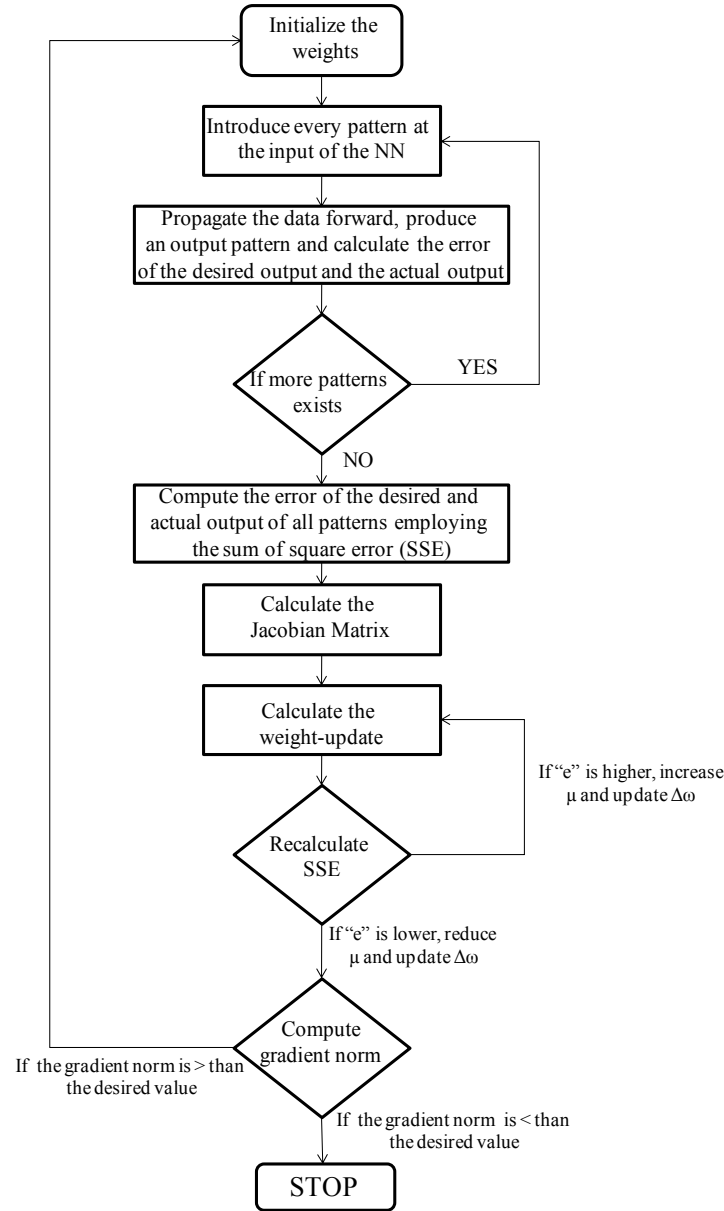
$$\nabla = J^T e \quad (38)$$

The combination of the above equations creates a weight-update formula, where  $\mu$  is the control parameter and  $I$  is an identity matrix.



$$\Delta\omega = -(J^T J + \mu I)^{-1} J^T e \quad (39)$$

It is important to mention that if  $\mu$  is a large number; the  $\Delta\omega$  equation will be similar to the gradient descent method. On the other hand, if  $\mu$  is zero, the equation will be similar to the Newton method. Figure 9 demonstrates the steps of the Levenberg-Maquard backpropagation algorithm (Priddy and Keller, 2005).



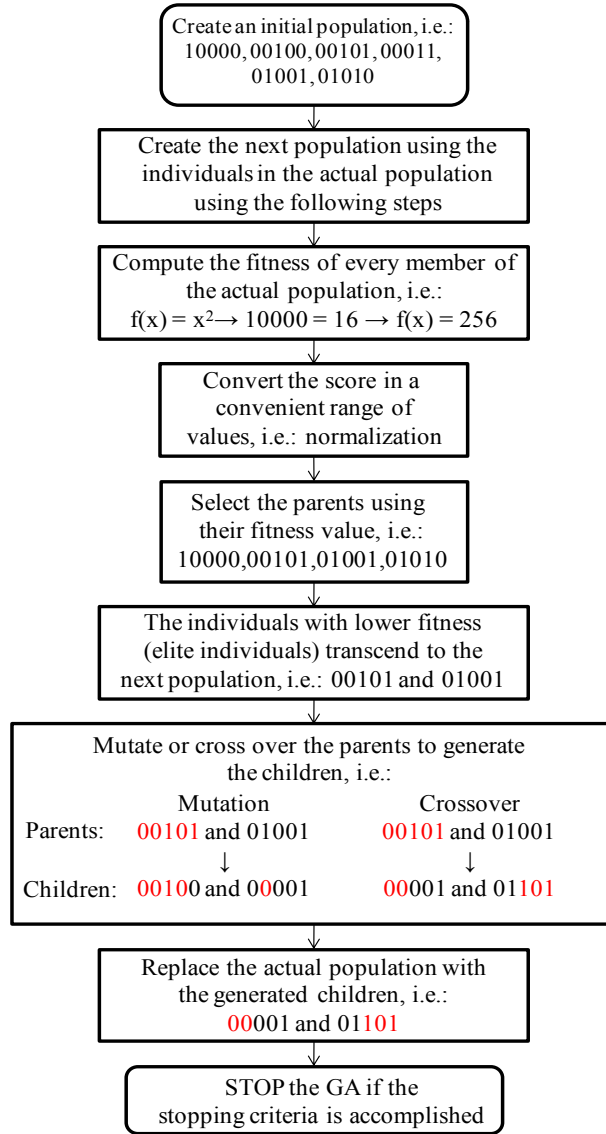
**Figure 9. Levenberg-Maquard Backpropagation Algorithm Steps (Priddy & Keller, 2005).**

## 2.3 Genetic Algorithms

Genetic Algorithms (GA) are searching and optimization algorithms motivated by natural selection evolution and natural genetics (Goldberg, 1989). Basically, the GA algorithm initially generates a random set of solutions (initial population of chromosomes), which are going to improve from iteration to iteration, by changing its features (mutation) and combining with other solutions (crossover). The simplest GA follows the next steps: Generate a random initial population of chromosomes, calculate the fitness of every chromosome in the population, apply selection, crossover and mutation and replace the actual population with the new population until the required solution is achieved. The main advantages of GA are: powerful computational effect, robustness, fault tolerance, fast convergence to a global optimal, capability of searching in complex landscape where the fitness function is discontinuous, can be combined with traditional optimization techniques (Tabu search) and have the ability to solve problem without needing human experts (Goldberg, 1989; Mitchell, 1996; Ruan 1997). The Genetic Algorithm is represented in the flow diagram (Figure 10). The parameters used in the Genetic Algorithm are showed in Table 1:

**Table 1. Genetic Algorithm Parameters**

GA Parameter	Unit
Population Size	20
Generations	100 Epoch
Selection Function	Tournament
Crossover Function	Scattered
Crossover Fraction	0.7
Mutation Function	Adapt Feasible
Stall Generation	50
Stall Time Limit	20
Function Tolerance	1e-06
Constraint Tolerance	1e-06
Time Limit	$\infty$
Maximum Iterations Number	100
Maximum Evaluation Number	100000
Variable Tolerance	1e-06
Mesh Size Tolerance	1e-06



**Figure 10. Genetic Algorithm Steps.**

## 2.4 Sliding Mode Control

The sliding mode controller is a technique in which the states of the systems reach a sliding surface (denote by “s”) and are maintained there by a shifting law design in order to stabilize the system using a state feedback control law (Khalil, 2002). In order to develop the procedure to design this controller, first the original transfer function is decomposing into a cascade system, and the following equations are obtained:

$$\dot{x}_1 = -\alpha_1 x_1 - \alpha_2 x_2 + u, \quad \dot{x}_2 = x_1 \quad \text{and} \quad y = \alpha_3 x_2 \quad (40)$$

where  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  are constants,  $x_2$  is equal to the system output,  $\dot{x}_2$  is the derivative of  $x_2$  and  $\dot{x}_1$  is the derivative of  $x_1$ . With the above variables defined, the following equations can be established:

$$\dot{x}_1 = f(x) + g(x)u \quad \text{and} \quad \dot{x}_2 = x_1 \quad (41)$$

where:  $f(x)$  and  $g(x)$  are nonlinear functions,  $g(x) > 0$ ,  $f(x)$  and  $g(x)$  need not to be continuous and  $x_2$  is stable if:

$$\dot{x}_2 = -ax_2, \quad a > 0 \quad (42)$$

On the other hand, if:

$$s = x_1 + ax_2 \rightarrow \dot{x}_2 = x_1 = -ax_2 + s \quad (43)$$

Then, the time derivate of  $s$  is:

$$\dot{s} = \dot{x}_1 + a\dot{x}_2 = f(x) + g(x)u + ax_1 \quad (44)$$

Therefore, the Lyapunov candidate function is:

$$V = \frac{1}{2} s^2 \quad (45)$$

where:

$$\dot{V} = \dot{s}s = [f(x) + g(x)u + ax_1] \quad (46)$$

$\dot{V}$  is negative definite if

$$f(x) + g(x)u + ax_1 \begin{cases} < 0 \text{ for } s > 0 \\ = 0 \text{ for } s = 0 \\ > 0 \text{ for } s < 0 \end{cases} \quad (47)$$

The stability is ensured if

$$u \begin{cases} < \beta(x) \text{ for } s > 0 \\ = \beta(x) \text{ for } s = 0 \\ > \beta(x) \text{ for } s < 0 \end{cases} \quad \text{and} \quad \beta(x) = -\frac{f(x) + ax_1}{g(x)} \quad (48)$$

Finally, the control law that will be used is:

$$u = \beta(x) - K \text{sign}(s) \quad (49)$$

where  $K > 0$ .

## 2.5 Linear Parameter Varying Systems

The Linear Parameter Varying (LPV) systems are systems with linear structure with a set of varying parameters over time. This type of systems can be represented either in input-output or state space form and either in continuous or discrete-time. The discrete-time representation of an LPV system is the following (Apkarian et al., 1995):

$$x(k+1) = A(k)x(k) + B(k)u(k) \quad (50)$$

$$y(k) = C(k)x(k) + D(k)u(k) \quad (51)$$

In equations (50) and (51), the matrix dependence over  $k$  could be anyone. On the other hand the continuous representation of an LPV system is shown next:

$$\dot{x} = A(\theta(t))x + B(\theta(t))u \quad (52)$$

$$y = C(\theta(t))x + D(\theta(t))u \quad (53)$$

where  $x$  represent the states,  $y$  is the measurement or output vector,  $u$  is the input vector and  $\theta$  represent the parameters varying over time.

An LPV system can be obtained through different methodologies: If the physical representation of the nonlinear system is available, the LPV model can be obtained through the Jacobian Linearization method, the State Transformation Method and the Substitution Function method (Shamma & Cloutier, 1993). The main objective of utilizing these methodologies is to hide the nonlinearity of the system in the varying parameters. On the other hand, if the model should be obtained from experimental data, the LPV system can be obtained using the Least Squares or the Recursive Least Square algorithm in different system operating points (Bamieh & Giarré, 2001).

In addition, an LPV controller could be represented as follows:

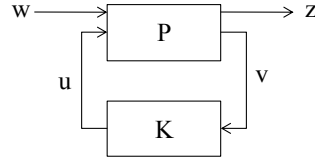
$$\dot{x}_c = A_c(\theta(t))x_c + B_c(\theta(t))y \quad (54)$$

$$u = C_c(\theta(t))x_c + D_c(\theta(t))y \quad (55)$$

where  $x_c$ ,  $y$  and  $u$  represent the states, the control output and the control input, respectively. This type of controller will adjust the dynamic variations of the plant in order to maintain stability and an adequate performance along of the trajectories of the parameter  $\Theta$ .

## 2.6 $H_\infty$ Loop Shaping Controller

The  $H_\infty$  control theory is a robust technique implemented in (Zames, 1981) to achieve robust performance and stabilization in a given system. This control theory uses the  $H_\infty$  norm which is the frequency response magnitude to maximum singular value of the interested transfer function (i.e. peak gain or worst case disturbances). The standard configuration problem for an  $H_\infty$  controller is shown in the next figure (Skogestad & Postlethwaite, 2005):



**Figure 11. General  $H_\infty$  Controller Configuration (Zames, 1981).**

where  $K$  is the  $H_\infty$  controller,  $P$  is a generalized plant,  $u$  are the control variables,  $w$  the exogenous signals,  $z$  the error signals which have to be minimized to achieve the control objectives and  $v$  the measured variables. In terms of state space, the above is rewritten as (Skogestad & Postlethwaite, 2005):

$$\begin{bmatrix} z \\ v \end{bmatrix} = P(s) \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad (56)$$

$$u = K(s)v \quad (57)$$

$$P = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \quad (58)$$

In which the linear fractional transformation of  $w$  to  $z$  is given by:

$$z = F_l(P, K)w \rightarrow F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} \quad (59)$$

The  $H_\infty$  control is then formulated as the minimization of  $H_\infty$  norm of  $F_l(P, K)$ .

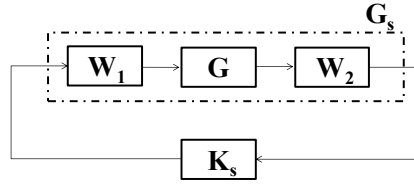
$$\| F_l(P, K) \|_\infty = \sup \bar{\sigma} (F_l(P, K)(j\omega)) \quad (60)$$

where  $\bar{\sigma}$  means the upper singular value.

The  $H_\infty$  loop shaping method proposed in (McFarlane & Glover, 1989) consists of a two stage process. In the first stage, the plant is augmented by pre and post compensators ( $W_1$  and  $W_2$ , respectively) in order to obtain the desired shape of the singular values in an open loop system. Second, the shaped plant is stabilized with  $H_\infty$  optimization. The shaped plant is denoted by:

$$G_s = W_2 G W_1 \quad (61)$$

The above configuration is observed in Figure 12:



**Figure 12. Shaped plant and controller (Skogestad and Postlethwaite, 2005).**

The controller  $K_s$  is obtained solving the robust stabilization problem for the shaped plant  $G_s$  with left coprime factorization.

$$G_s = M^{-1}N \quad (62)$$

Therefore, the perturbed plant model  $G_p$  can be represented as:

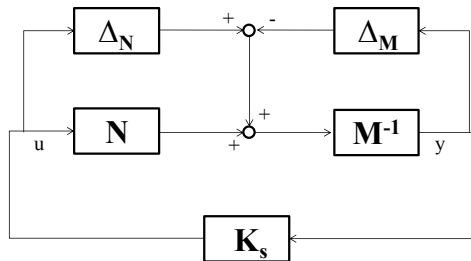
$$G_p = (M + \Delta_M)^{-1}(N + \Delta_N) \quad (63)$$

where  $\Delta_M$  and  $\Delta_N$  represent the uncertainty of the nominal plant  $G$ . Then, the goal of robust stabilization is to stabilize  $G$  and the family of perturbed plants  $G_p$ .

$$G_p = \{(M + \Delta_M)^{-1}(N + \Delta_N) : \| [\Delta_M \ \Delta_N] \|_\infty < \varepsilon\} \quad (64)$$

where  $\varepsilon > 0$  is the stability margin.

The above control design problem is represented graphically in Figure 13:



**Figure 13.  $H_\infty$  robust stabilization problem (Skogestad & Postlethwaite, 2005).**

The control problem presented in the above figure satisfies the stability property if and only if the nominal feedback system is stable and:

$$H_{\infty}norm \triangleq \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_{\infty} < \varepsilon \quad (65)$$

where  $(I - GK)^{-1}$  is the sensitivity function.

Finally, the feedback controller for the plant  $G$  of Figure 12 will be:

$$K = W_1 K_s W_2 \quad (66)$$

For each model of the desired system that need to be controlled, the  $H_{\infty}$  controller is designed by using the loop shaping method and the following steps are realized: First, the worst case system faults ( $G_{faults}$ ) are simulated and identified in the form of a Laplace function. Second, the Laplace functions are compared against the non-fault system function. Third, a loop shaping control synthesis is performed to calculate an optimal  $H_{\infty}$  controller for the Laplace fault-functions. This controller shapes the sigma plot of the Laplace fault-function and obtains the desired loop shaping with a precision parameter called “ $GAM$ ” (e.g. if  $GAM$  should be  $\geq 1$  with  $GAM = 1$  being a perfect match). Fourth, the stable-minimum-phase loop-shaping is calculated squaring down a pre-filter  $W$  (Le & Safonov, 1992):

$$W = C(sI - A)^{-1}B + D \quad (67)$$

and the shaped plant  $G_s$  is square in state space formulation:

$$G_s = G_{faults}W = C(sI - A)^{-1}B + D \quad (68)$$

Then, the desired shape  $G_s$  is accomplished with high precision in the frequency range by the shaped plant. After the above procedure, the Normalized-Coprime-Factor control synthesis (Glover & McFarlane, 1989) is used to calculate the ideal loop-shaping controller ( $K_s$ ):

$$K_s = C(sI - A)^{-1}B + D \quad (69)$$

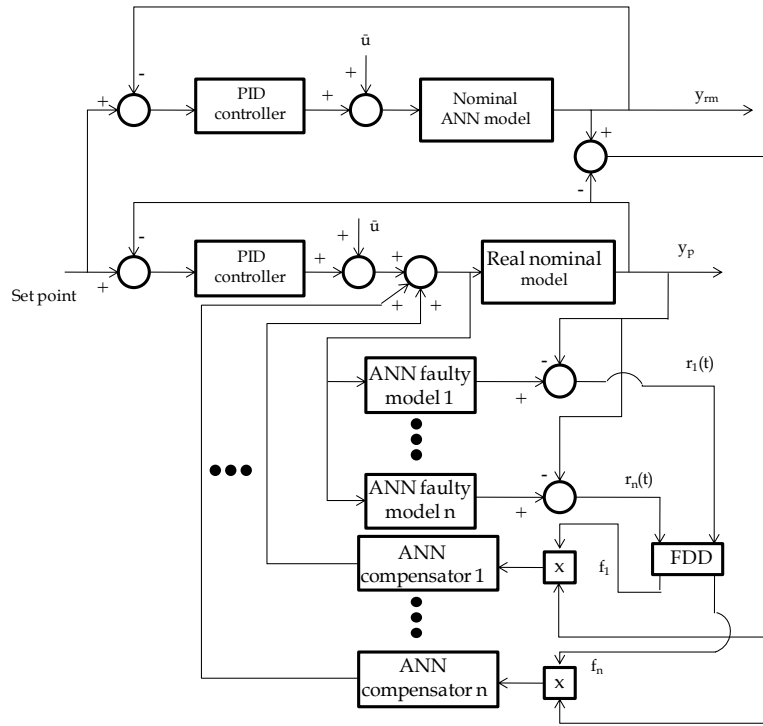
Finally, with the equation (69) the  $H_{\infty}$  controller is computed using:

$$H_{\infty} = W * K_s \quad (70)$$

## 2.7 Literature Review

First, for the case of Artificial Neural Networks, they have been applied in Fault Tolerant Control because they are helpful to identify, detect and accommodate system faults. Polycarpou & Helmicki (1995) proposed a construction of automated fault detection and accommodation architecture that uses on-line approximators and adaptive-learning schemes. The online approximator is an ANN model that monitors changes in the system dynamics due to a failure. Patton et al. (1999) use a scheme of an Artificial Neural Network to detect and isolate a fault in two steps: residual generation and decision making. In the first step, a residual vector characterizes the fault and then the second step process the residual vector information in order to locate the fault and the time of occurrence. Once the residual is trained, qualitative knowledge of the plant can be added. This combination of qualitative and quantitative approached is helpful to decrease the number of false alarms in the fault decision making step. Polycarpou (2001) proposed a methodology for fault accommodation of a multivariable nonlinear dynamical system using a learning approach that monitors and

approximates any abnormal behavior using ANNs and adaptive non linear estimation. When a fault occurs, the Artificial Neural Network is used to estimate the nonlinear fault function supplying a framework for fault identification and accommodation. The ANN at the beginning of the monitoring stage is capable of learning the modeling errors in order to improve the system robustness. Gomaa (2004) recommended a fault tolerant control approach based on multi-ANN system faulty models. The nominal plant is nonlinear and is vulnerable to faults. A feedforward neural network is trained as the nominal model. Two PID controllers are used, one for the nominal plant and the other for the Artificial Neural Network imitating the nominal plant (reference model). Both PID controllers were tuned using genetic algorithms. If there exist a difference between the nominal plant ( $y_p$ ) and the reference model ( $y_{rm}$ ) a nonzero residual is generated. Then, depending on the magnitude of the residual, an ANN faulty model and its respective compensation path are selected to counteract the fault and improve the system operating conditions. This can be observed in Figure 14.

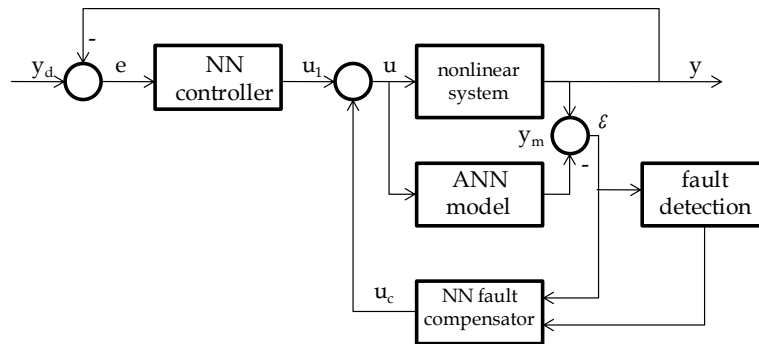


**Figure 14. Multi-ANN faulty models FTC scheme (Gomaa, 2004).**

Wang & Wang (1999) proposed an Artificial Neural Network-FTC where the nominal system is controlled by a Pseudolinear Neural Network (PNN) based one-step-ahead controller that uses a modified gradient approach. When the system is with no fault, a PNN model connected in parallel to the plant model can be trained and used for the design of the control algorithm. The PNN model is helpful for the detection of a fault. In addition, when a fault is present, a residual signal is generated and an extra Artificial Neural Network based fault compensation loop is imported in order to provide the closed loop stability. This last Artificial Neural Network is a two layer perceptron network and its weights are updated using the modified gradient approach. This FTC system is shown in Figure 15. Pashilkar et al. (2006) proposed a neural controller that improves the fault tolerant potential of a fighter aircraft during landing. The faults are caused



by severe winds or stuck control surfaces and can be divided in single faults (aileron or elevator stuck) or double faults (aileron and elevator stuck). This Artificial Neural Network controller employs a feedback error learning method with a dynamic radial basis function Artificial Neural Network. The Artificial Neural Network uses on-line training and not a-priori training. This kind of controller helped to improve the capability of handling large faults and also helps to achieve the desired requirements. Perhinschi et al. (2007) presented a methodology for detection, identification and accommodation of sensor and actuator failures inside fault tolerant control laws. The fault detection and identification uses neural estimators. The accommodating control law design for the actuator fault is done using non-linear dynamic inversion with Artificial Neural Network augmentation. Whereas the accommodation of sensor faults is accomplished by changing the failed sensor output for neural estimates calculated in the detection and identification process. This approach can handle sensor and actuator faults successfully. It uses membership functions to describe the mathematical model of process. Yen & DeLima (2005) presented an Artificial Neural Network trained on-line with a global dual heuristic programming architecture. This approach has also a supervision structure made from decision logic. This supervision level is very efficient to identify the controller faults in early stages and can supply new values to improve the convergence utilizing dynamic model bank information. Patan and Korbicz (2009) proposed a FTC system for a boiler unit in which a recurrent neural network was used to build an on-line fault approximator. Then, this approximator was employed in the fault detection and accommodation of the system. Yu et al. (2009) introduced a hybrid dynamic ANN based on fault diagnosis and fault tolerance method. This methodology is an incorporation of feedforward and recurrent ANN in order to form a dynamical identification model for the system. Finally, Kamalasadan and Ghandakly (2011) proposed a fighter aircraft pitch-rate command-tracking controller based on a neural network parallel controller. The scheme consists of a radial basis function neural network in parallel with a model reference adaptive controller and the controller is able to control the changes in the aircraft system.



**Figure 15. Artificial Neural Network FTC scheme proposed by (Wang & Wang, 1999).**

On the other hand, for Genetic Algorithms (GA), Schroder et al. (1998), proposed a fault tolerant control technique for an active magnetic bearing. In this approach a nonlinear model of a turbo machine rotor from the Rolls-Royce lifted up by an active magnetic bearing was presented. This model is capable of modeling different configurations of magnetic bearings. A multi-objective genetic algorithm was used to generate and adequate PID controller for the active magnetic bearing with different bearing configurations.

Also the fault accommodation was done using a centralized fault compensation scheme. Sugawara et al. (2003) showed a fault tolerant control approach using multi-layer neural networks with a genetic algorithm. The propose of this approach was to develop a self-recovery technique implemented for large scale Artificial Neural Networks programmed in a single ship to accommodate faults without the need of a host computer. This FTC scheme uses hardware redundancy and weight retraining using a genetic algorithm in order to reconfigure the Artificial Neural Network to accommodate the fault. The objective of the genetic algorithm is to reduce the error between the actual output and the desired output.

And finally, in the last years some LPV systems and control methods have been developed. For example, in Shamma & Cloutier (1993), a gain-scheduled design for a missile longitudinal autopilot was presented. The missile dynamics are represented in an LPV form using state transformation. A robust controller employing  $\mu$  synthesis was design to achieve the angle of attack control. In addition, an inner/outer loop structure was designed being the angle of attack the inner loop and the normal acceleration control the outer loop. Apkarian et al. (1995) presented a gain scheduled along parameter trajectories controller with guaranteed  $H_\infty$  performance for a class of LPV systems. The controller synthesis problem is formulated using LMIs that can be efficiently solved. The system state space matrices depend on a time varying parameters vector in which the parameters can be measured in real time and are fed to the controller in order to achieve the desired performance and the robustness of the closed-loop system. Li et al. (1999) proposed a gain-scheduled controller for a class of LPV systems in which the controller synthesis was formulated using LMIs. In addition, the performance specifications of this controller were in terms of  $L_2$  gain, general quadratic constraints, peak gain, generalized  $H_2$  performance and input and output constraints. In Bamieh & Giarré (2001), a discrete-time LPV model for gain scheduling control and an identification method for nonlinear system are presented. The identification of the LPV model is done through a linear regression as Least Mean Squares or Recursive Least Squares. Fodor & Tóth (2004) developed a robust control structure for a speed sensorless vector control of an Induction Motor. This control structure uses mixed sensitivity LPV  $H_\infty$  control theory resulting in a more accurate control of the Induction Motor. Groot et al. (2005) proposed an LPV control technique to create position-dependent controllers (feedback controllers) that are able to accommodate themselves in order to maintain an optimal closed-loop performance. In Kwiatkowski & Werner (2005) a methodology for reducing the scheduled parameters in an LPV controller was presented. This reduction was done using Principal Components Analysis (PCA) to typical scheduling trajectories. The proposed method in this thesis creates a trade-off between the number of reduced parameters in the LPV controller and the requested accuracy of the model. The LPV controller presented in this methodology uses the mixed sensitivity approach and compares its performance with a fixed-gain controller created using a robust  $H_2$  approach. Giarré et al. (2006) developed the identification of a nonlinear plant parameterizing its dynamics as an LPV model where the control assignment is the output regulation of different system set points. After obtaining the LPV model, the model is used to design a gain scheduled controller linear feedback controller (off-line) and a nonlinear correction controller (on-line). This last controller is the consequence of horizon optimization based on invariant set theory. Hecker (2006) presented a two vehicle steering controllers to enhance the yaw dynamics of a mid-size passenger car using robust  $H_\infty$  synthesis techniques. Both controllers satisfied the

desired mixed-sensitivity performance specifications. The first controller is designed with  $\mu$  synthesis guarantying robust performance if the parameters are uncertain but constant. On the other hand, the second controller is based on LPV control techniques and guarantees robust performance even though the parameters variation. Salcedo & Martínez (2006) proposed a methodology for the identification of LPV systems based on the previous linear identification of different operating points and adjusting them using a Least Square Levenberg & Maquardt algorithm allowing the adaptation of non-linear models. The LPV models are defined via Lineal Fractional Representation (LFR) of the variable parameters over time. Once the LPV model is obtained, the next step is to design an LPV linear controller. In Wang & Weiss (2006), a self-scheduled control for a doubly-fed induction generator was presented. The controller was designed employing the LMI methodology for LPV systems using linear interpolation. In addition, a controller reduction for the LPV controller was realized based on the truncation of fast modes. Lee & Park (2007) developed a robust dynamic feedback Model Predictive Controller (MPC) for LPV systems. The control law of this controller is calculated using LMIs at each sampling time solving a convex optimization problem. In addition, a parameter dependent Lyapunov function is developed in order to obtain a less conservative condition for the system stability. Robert et al. (2007) designed a polytopic methodology for LPV systems to obtain a  $H_\infty$  sampling period dependent controller in order to deal with the adaptation of a real-time controller's sampling period. In Rodrigues et al. (2007), a FTC methodology for polytopic LPV systems was presented. The most important contribution of this work was the development of a Static Output Feedback (SOF) that maintains the system performance using an adequate controller reconfiguration when a fault appears. In addition, the controllers used in this methodology were arranged through LMIs to maintain the system closed-loop stability. Gilbert et al. (2008) developed a fixed-order controller for SISO gain scheduled with guaranteed stability and  $H_\infty$  performance covering the entire scheduled parameter range. This approach uses polynomials as modeling objective. Also, it uses flexible LMIs conditions to allow the polynomial dependence of the open-loop system and the controller transfer functions in the scheduled parameters. And finally, it uses the LMI conditions decoupling between the Lyapunov and the controller variables for the design of the parameter dependent Lyapunov function and the fixed-order controller. Liang & Marquez (2008) proposed a global gain scheduling synchronization method for identical synchronization of quadratic chaotic systems. The quadratic chaotic system contains nonlinearities in the quadratic terms of the systems states such that it can be transformed into an LPV system. The implementation of the gain scheduling technique allows achieving the global synchronization for the quadratic chaotic system. In Poussot-Vassal et al. (2008), a new semi-active suspension control strategy was designed using Linear Parameter Varying (LPV) theory ensuring internal stability of the system. The controller is based on LMIs polytopic LPV/ $H_\infty$  control synthesis. Xie & Eisaka (2008) developed a two-degree-of freedom (TDOF) control for LPV systems. To design the TDOF controller, a coprime factorization of the LPV system was realized. Then, the TDOF controller approach for LTI systems is extended to LPV systems and the controller design problem can be established in terms of LMIs related to  $L_2$  gain performance in order to reject disturbances and to have a good tracking performance. In Bosche et al. (2009), a Fault Tolerant Control (FTC) structure for vehicle dynamics is developed employing an LPV model with actuator failures. The methodology described in this research is based on the resolution of Linear Matrix

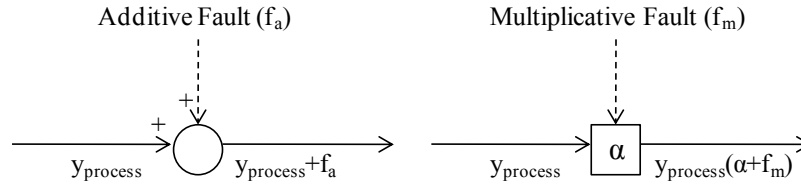
Inequalities (LMIs) using the D-stability concept and a Parameter-Dependent Lyapunov Matrix (PDLM). De Lira et al. (2009) presented a fault diagnosis problem in a Polymer Electrolyte Membrane Fuel Cell (PEM-FC) system using a model-based approach. The PEM-FC system is represented through an LPV model. The fault diagnosis method is based on the generation of structured residuals and the fault sensitivity analysis evaluation. In addition, the variation of the system dynamics is contemplated using an LPV observer when generating residuals and the fault isolation method is based on the relative fault sensitivity concept. Ginter & Pieper (2009) developed a  $H_\infty$  LPV controller based on a physical model and an iterative simulation based testing. To develop the controller, four operation regions of the system were established. Therefore, the controller synthesis results in a convex optimization problem with LMIs having available numerous well developed numerical algorithms. In Henry et al. (2009) a robust fault detection and isolation filters for LPV systems using a LFR were developed. The objective of the FDI filter is to minimize the influence of the unknown residuals inputs in the  $H_\infty$  case and to maximize the fault sensitivity in the  $H_2$  case. The parameters are optimized using LMI. Luzar et al. (2009) designed a Matlab block-set that can be employed for FTC of LPV systems. In this proposal, an observer estimates an unknown state of the system. Then, a fault identification system is executed and finally the control strategy is carried out using LMIs pole placement. In Montes de Oca et al. (2009), an Admissible Model Matching (AMM) FTC method based on LPV fault representation was presented. In this approach, the faults were considered as scheduling variables in the LPV fault representation allowing the controller adaptation on-line. In addition, a FDI scheme detect, isolate and estimate the faults, when the FDI is not able to estimate the magnitude of the fault a passive FTC method based on a single controller can manage the admissible faults. The AMM methodology is based on a set of admissible behaviors characterized through LMI regions. The LMIs are able to locate the closed-loop poles inside the unit circle region. Also, the fault accommodation can be represented using several LMIs. In Chen et al., (2010) an LPV Pole-placement approach as an FTC problem, this scheme involves pole-placement within suitable LMI regions in order to guarantee stability and performance of a multi-fault LPV estimation used within an FTC structure. In Montes de Oca et al. (2010) a Fault Tolerant Control design using LPV admissible model matching with  $H_2/H_\infty$  performance was presented. In this scheme the reconfiguration of the controller is done on-line based on using LPV gain-scheduling techniques allowing changes in the system parameters due to changes in the operating points and faults. This scheme is an active FTC strategy, in which the quadratic  $H_2/H_\infty$  performance helps to select the best controller. Paton and Klinkhieo (2010) presented an LPV fault estimation and FTC of a two-link manipulator. This scheme combines the use of LPV fault estimation and compensation to achieve active FTC performance requirements in which the system is characterized by sets of LMIs and can be obtained using efficient interior-point algorithms. In addition, a polytopic estimator is synthesized to generate actuator fault used in an FTC scheme to schedule the nominal system state feedback gain, therefore, the system performance can be maintain over a wide range of operating points within a proposed polytopic model. Montes de Oca & Puig (2010) presented a FTC scheme that uses a virtual sensor for LPV systems. In this scheme the control loop is reconfigure such that the nominal controller could be still used after the presence of a fault without the need of a retuning because the plant with the sensor fault is modified adding a virtual sensor block that hides the fault and allows the controller to see the plant as

a non-faulty plant. The virtual sensor is designed using polytopic LPV technique and LMIs and the LPV state feedback controller is designed to achieve quadratic  $H_2/H_\infty$  performance using a polytopic representation of the system that contribute to solve a finite number of algebraic LMIs.

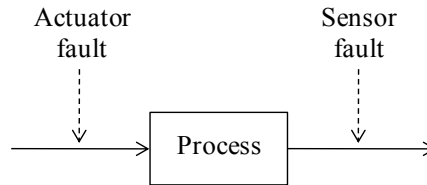
## 2.8 Faults Types

A fault is an unpermitted deviation of a parameter or characteristic of the system (e.g. physical quality). The fault can be distinguished in abrupt, incipient or intermittent faults. With respect to the way of introducing the fault in the system, the fault can be classified in additive or in multiplicative faults (Fortuna, 2007).

To test the approaches presented in Chapter 3 and Chapter 4, different types of faults are introduced and simulated in the considered testbed cases. The first type of fault is an additive abrupt fault, the second type of fault is an additive gradual fault, and the third type of fault is a multiplicative fault (see Figure 16). All types of faults are introduced in actuator and sensors (see Figure 17). An additive fault will modify the quantity of the nominal value by the addition of a quantity  $f(t)$ . An abrupt additive fault in actuators represent, for instance, a pump stuck or in sensors a constant bias in measurements. A gradual additive fault could be a progressive loss of electrical power in pump or a drift in the sensor measurements. Finally, a multiplicative fault is represented as a degradation of the nominal system, the nominal quantity is multiplied by a quantity  $f(t)$ . For example, actuator multiplicative fault is represented as  $u_f = \alpha u$ , where  $u_f$  represent the system input with the actuator fault,  $\alpha$  represents the degradation percentage of the actuator, and  $u$  is the nominal system input.



**Figure 16. Representation of Additive and Multiplicative Faults (Mahmoud et al., 2003).**



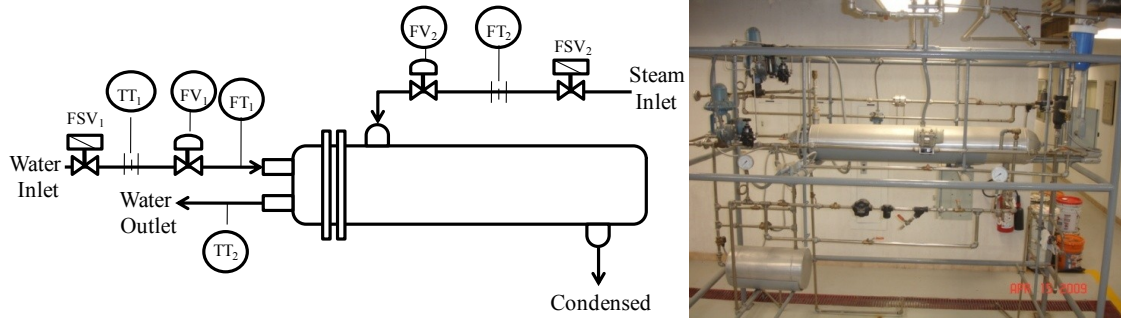
**Figure 17. Representation of Actuator and Sensor Faults.**

# **CHAPTER 3**

## **INDUSTRIAL HEAT EXCHANGER PROPOSED SCHEMES, EXPERIMENTS AND RESULTS**

### 3 Industrial Heat Exchanger

The first system used as test bed (shown in Figure 18 and Table 2) is a shell and tube Industrial Heat Exchanger that has two inputs: water and steam flows controlled by pneumatic valves ( $FSV_1$  and  $FSV_2$ , respectively). The water pass inside the tubes at room temperature and the steam pass through the tube walls in order to transfer heat to the water. In addition, the industrial heat exchanger has one output, in which the water temperature is measured by a thermistor ( $TT_2$ ). Variations in water and steam flows are determined by flow transmitters ( $FT_1$  and  $FT_2$ , respectively).



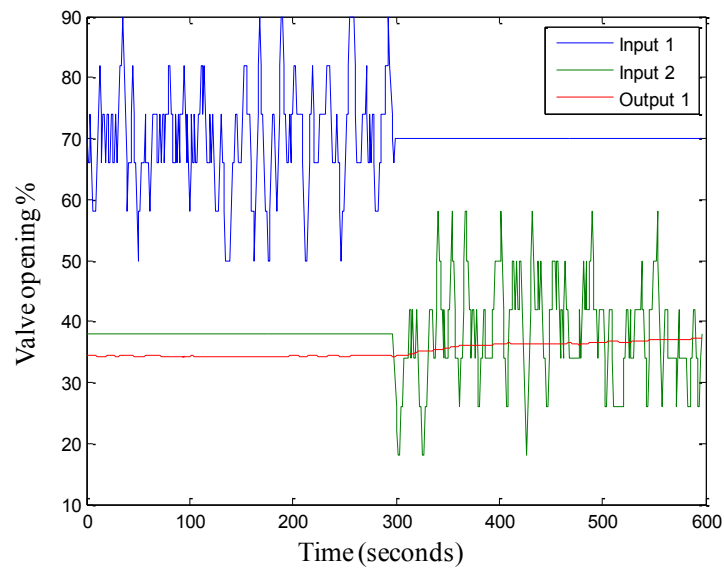
**Figure 18. Industrial Heat Exchanger used in the experiments.**

**Table 2. Industrial Heat Exchanger Sensors/Transmitters Description.**

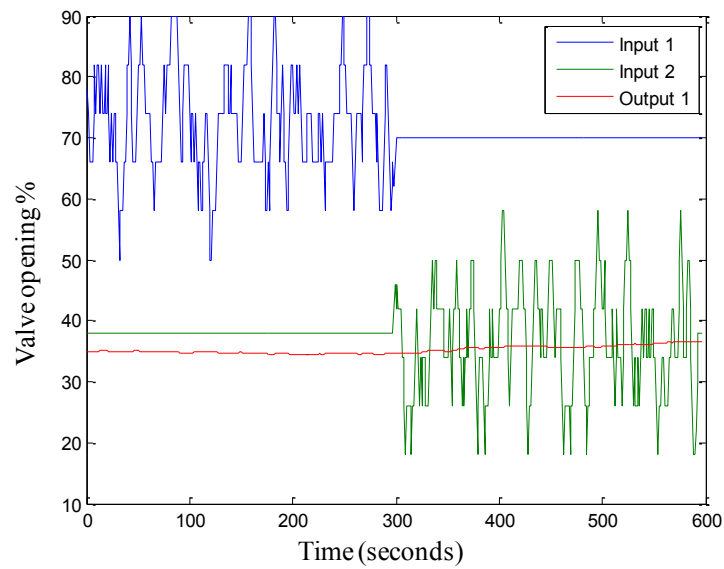
Tag Name	Description
$FSV_1$	Solenoid valve in the water inlet
$TT_1$	Temperature transmitter of the water inlet
$FV_1$	Pneumatic control valve in the water inlet
$FT_1$	Flow transmitter in the water inlet
$TT_2$	Temperature transmitter of the water outlet
$FV_2$	Pneumatic control valve in the steam inlet
$FT_2$	Flow transmitter of the steam inlet
$FSV_2$	Solenoid valve in the steam inlet

To obtain the continuous model of this process, an identification experiment was performed, where a Pseudo Random Binary Sequence (PRBS) was applied to water and steam valves, and variations in water temperature were recorded. With the data obtained in the PRBS test, the identification was achieved using Matlab®. This was done with the help of the system Identification Toolbox commands described below. The following commands are design for the estimation and validation of linear models from multiple-input/single-output (MISO) data to find the one that best represents the system dynamics.

- The variables of the data to estimate the model are: input1, time, and output1. The variables for the validation of the model are: input2, time, and output2. The input1 and input2 variables include in each of the steam and water valves opening percentage. The output 1 and output 2 variable represent the water temperature at the outlet of the Industrial Heat Exchanger. The next figure shows the representation of these variables.



**Figure 19. PRBS test for model estimation.**



**Figure 20. PRBS test for model validation.**

- Establish the sampling time interval:  
 $T_s = 1$ ; % Sampling interval is 1 seg



- Create two data objects to encapsulate data values and data properties into a single entity, ze is created for model estimation and zv is created for model validation.

```
ze = iddata(Output1,exp1,Ts);
```

```
zv = iddata(Output2,exp2,Ts);
```

- To view the properties of the estimation data, the next command was applied. This command return data properties as: domain, name, output data, output name, etc:

```
get(ze)
```

- To plot the data objects the following commands are used:

```
plot(ze) % Plot the estimation data
```

```
figure % Open a new MATLAB Figure window
```

```
plot(zv) % Plot the validation data
```

- Estimate the transfer function using spectral analysis for a fixed frequency resolution (estimate the frequency response):

```
Ge=spa(ze);
```

```
bode(Ge)
```

- Estimate the step response from the data:

```
step(ze,30)
```

- Estimate the delay in the system:

```
delayest(ze)
```

- Use the struc command to create a matrix of possible model orders and the selstruc command to compute the loss functions:

```
NN1 = struc(1:50,1:50,0);
```

```
selstruc(arxstruc(ze(:,1),zv(:,1),NN1)); %selects the model order
```

```
NN2 = struc(1:50,1:50,0);
```

```
selstruc(arxstruc(ze(:,2),zv(:,2),NN2));
```

- Use the following command to create two model structures, one for each input/output combination (P2 represents a transfer function with two poles)

```
midproc0 = idproc({'P2','P2'});
```

```
midproc = pem(ze,midproc0);
```

- To view the two resulting models, type the following command:

```
present(midproc)
```

- The following command compares the actual output and the model output:

```
compare(zv,midproc)
```

- The residual analysis is realized using:

```
resid(zv,midproc0)
```

Then, the following model was obtained:

$$G_p = G_{steam} + (-G_{water}) \quad (71)$$

$$G_p = \frac{0.00002}{s^2+0.004299s+0.00002} + \frac{-0.000013}{s^2+0.007815s+0.00008} \quad (72)$$

$$T(s) = \frac{0.00002}{s^2+0.004299s+0.00002} F_{steam}(s) - \frac{0.000013}{s^2+0.007815s+0.00008} F_{water}(s) \quad (73)$$

$$G_{steam} = \frac{T(s)}{F_{steam}(s)} = \frac{0.00002}{s^2+0.004299s+0.00002} \quad (74)$$

$$G_{water} = \frac{T(s)}{F_{water}(s)} = \frac{-0.000013}{s^2+0.007815s+0.00008} \quad (75)$$

where  $G_p$  represents the Process Model,  $G_{steam}$  and  $G_{water}$  describes the steam and water model of the industrial heat exchanger, respectively.  $T(s)$  describes the Water Temperature at the exit and  $F_{steam}(s)$  and  $F_{water}(s)$  represent the steam and water flow, respectively. It is important to mention, that the obtained model was composed of two additive parts  $G_{steam}$  and  $G_{water}$ , in order to implement this model in an MRAC system, the model was decomposed in to two subsystems Steam and Water. In addition, the obtained model is just an approximation of the Industrial Heat Exchanger process and it was used just as an example to implement the FTC schemes.

### 3.1 Model Reference Adaptive Controller (MRAC)

In order to derive a FTC scheme for the Industrial Heat Exchanger, a Model Reference Adaptive Controller was designed. The MRAC scheme was chosen as the based controller because it guarantees asymptotic output tracking and it has a direct physical interpretation.

In the design of the MRAC controller it is important to take in count the two second order systems: steam and water systems. With the background theory presented in Chapter 2, the following equations were developed:

$$y_{steam\_process} = G_{steam} * u = \left( \frac{0.00002}{s^2+0.004299s+0.00002} \right) (\theta_1 u_c - \theta_2 y_{process}) = \left( \frac{0.00002\theta_1}{s^2+0.004299s+0.00002+0.00002\theta_2} \right) u_c \quad (76)$$

$$y_{water\_process} = G_{water} * u = \left( \frac{-0.000013}{s^2+0.007815s+0.00008} \right) (\theta_1 u_c - \theta_2 y_{process}) = \left( \frac{-0.000013\theta_1}{s^2+0.007815s+0.00008-0.000013\theta_2} \right) u_c \quad (77)$$

Using equations (76) and (77), the error can be redefined as:

$$e_{steam} = \left( \frac{0.00002\theta_1}{s^2+0.004299s+0.00002+0.00002\theta_2} \right) u_c - (G_{steam\_reference\_model} * u_c) \quad (78)$$

$$e_{water} = \left( \frac{-0.000013\theta_1}{s^2+0.007815s+0.00008-0.000013\theta_2} \right) u_c - (G_{water\_reference\_model} * u_c) \quad (79)$$

Therefore, the error partial derivatives with respect to the adaptive feedforward ( $\theta_1$ ) and adaptive feedback ( $\theta_2$ ) gain are specified as equation 80 for the steam process and equation 81 for the water process:

$$\frac{\partial e}{\partial \theta_1} = \left( \frac{0.00002}{s^2 + 0.004299s + 0.00002 + 0.00002\theta_2} \right) u_c \quad \text{and} \quad \frac{\partial e}{\partial \theta_2} = - \left( \frac{0.00002\theta_1}{s^2 + 0.004299s + 0.00002 + 0.00002\theta_2} \right) y_{process} \quad (80)$$

$$\frac{\partial e}{\partial \theta_1} = \left( \frac{-0.000013}{s^2 + 0.007815s + 0.00008 - 0.000013\theta_2} \right) u_c \quad \text{and} \quad \frac{\partial e}{\partial \theta_2} = - \left( \frac{b_r \theta_1}{s^2 + 0.007815s + 0.00008 - 0.000013\theta_2} \right) y_{process} \quad (81)$$

Consequently, the Process characteristic equation can be transformed into equations (82) and (83), because the MRAC system aim is to approximate the Process Model with the Reference Model.

$$steam \rightarrow s^2 + 0.004299s + 0.00002 + 0.00002\theta_2 \approx s^2 + 0.004299s + 0.00002 \quad (82)$$

$$water \rightarrow s^2 + 0.007815s + 0.00008 - 0.000013\theta_2 \approx s^2 + 0.007815s + 0.00008 \quad (83)$$

Finally, from equations (82) and (83), the error partial derivatives are transformed; and employing the MIT rule, the update rules for the adaptive feedforward ( $\theta_1$ ) and adaptive feedback ( $\theta_2$ ) gain are obtained as follows:

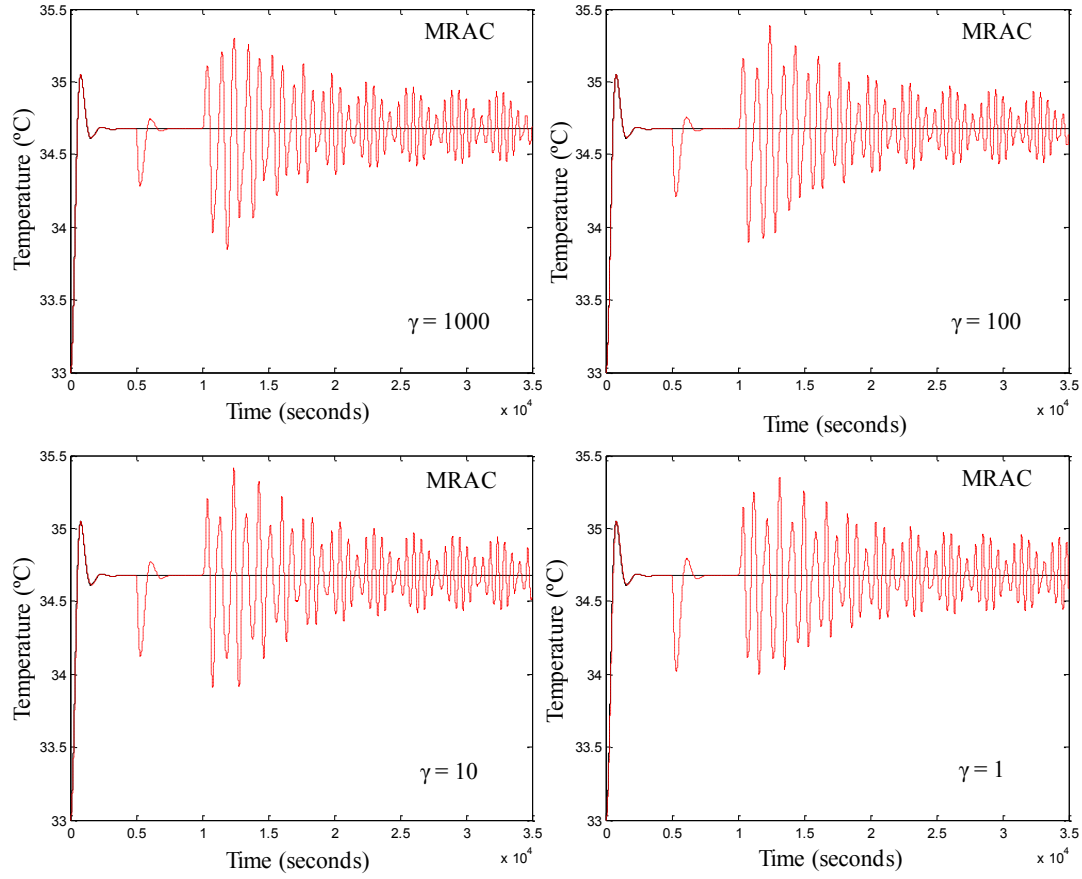
$$steam \rightarrow \frac{d\theta_1}{dt} = -\gamma \left( \frac{0.004299s + 0.00002}{s^2 + 0.004299s + 0.00002} u_c \right) e \quad \text{and} \quad \frac{d\theta_2}{dt} = \gamma \left( \frac{0.004299s + 0.00002}{s^2 + 0.004299s + 0.00002} y_{process} \right) e \quad (84)$$

$$water \rightarrow \frac{d\theta_1}{dt} = -\gamma \left( \frac{0.007815s + 0.00008}{s^2 + 0.007815s + 0.00008} u_c \right) e \quad \text{and} \quad \frac{d\theta_2}{dt} = \gamma \left( \frac{0.007815s + 0.00008}{s^2 + 0.007815s + 0.00008} y_{process} \right) e \quad (85)$$

To select the value of  $\gamma$ , different experiments with different  $\gamma$  sizes were realized. In these experiments a sensor fault of 10 % of nominal system deviation was introduced at time 5000 seconds and an actuator fault of 5 % of nominal system deviation was introduced at time 15000 seconds. In summary, four different size of  $\gamma$  were tested (1000, 100, 10, 1) and the results are showed in Table 3 and Figure 21.

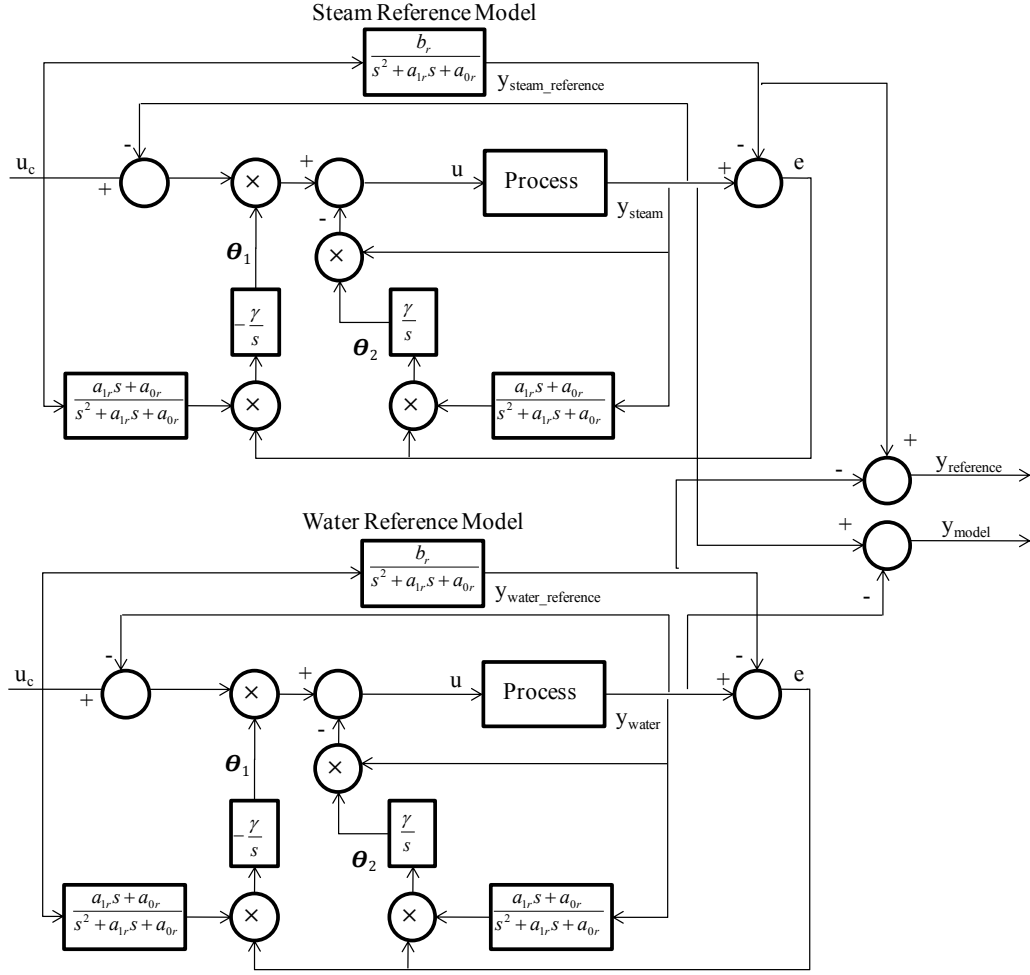
**Table 3. MSE of different sizes of  $\gamma$ .**

$\gamma$	Sensor Fault	Actuator Fault
1000	0.0687	0.0291
100	0.0982	0.0289
10	0.1387	0.0292
1	0.1937	0.0324



**Figure 21. MRAC Results testing different sizes of  $\gamma$ .**

From Figure 21 and Table 3 the selected value of  $\gamma$  to realize the following experiments was 1000, because it has the lower MSE in sensor and actuator faults. With the above equations the following controller was implemented (Figure 22):



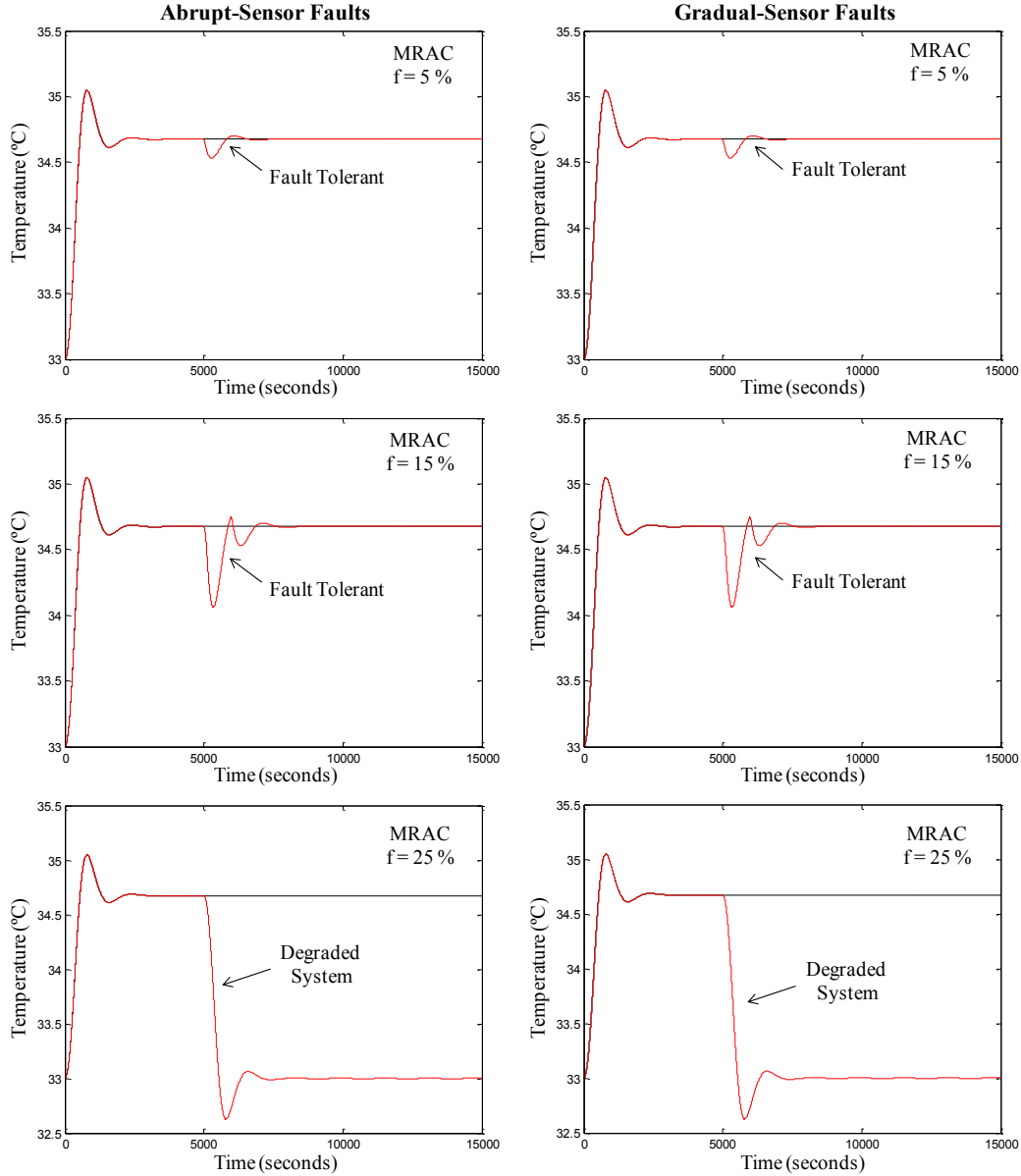
**Figure 22. Fault Tolerant MRAC Controller Structure.**

### 3.1.1 Experiments and Results

This section explains the different experiments that have been realized using the MRAC in the Industrial Heat Exchanger. In these experiments, two different types of faults were simulated in the implemented schemes: abrupt faults and gradual faults. In the abrupt faults case, the whole magnitude of the fault is developed in one moment of time and was simulated with a step function. On the other hand, gradual faults are developed during a period of time and are implemented with a ramp function.

Both types of faults, abrupt and gradual, have been considered in sensors (feedback), in which the properties of the process are not affected, but the sensor readings are wrong. And, they have also been considered in the actuators (process entry) in which the process behavior can change or can be interrupted. In each experiment, a fault was introduced at time 5000 seconds. Three different fault levels were simulated: 5%, 15% and 25%. The fault size is given in terms of percentage deviation from the normal operational value. In the next figures, three different results are explained: robust (no changes occur in the system after the

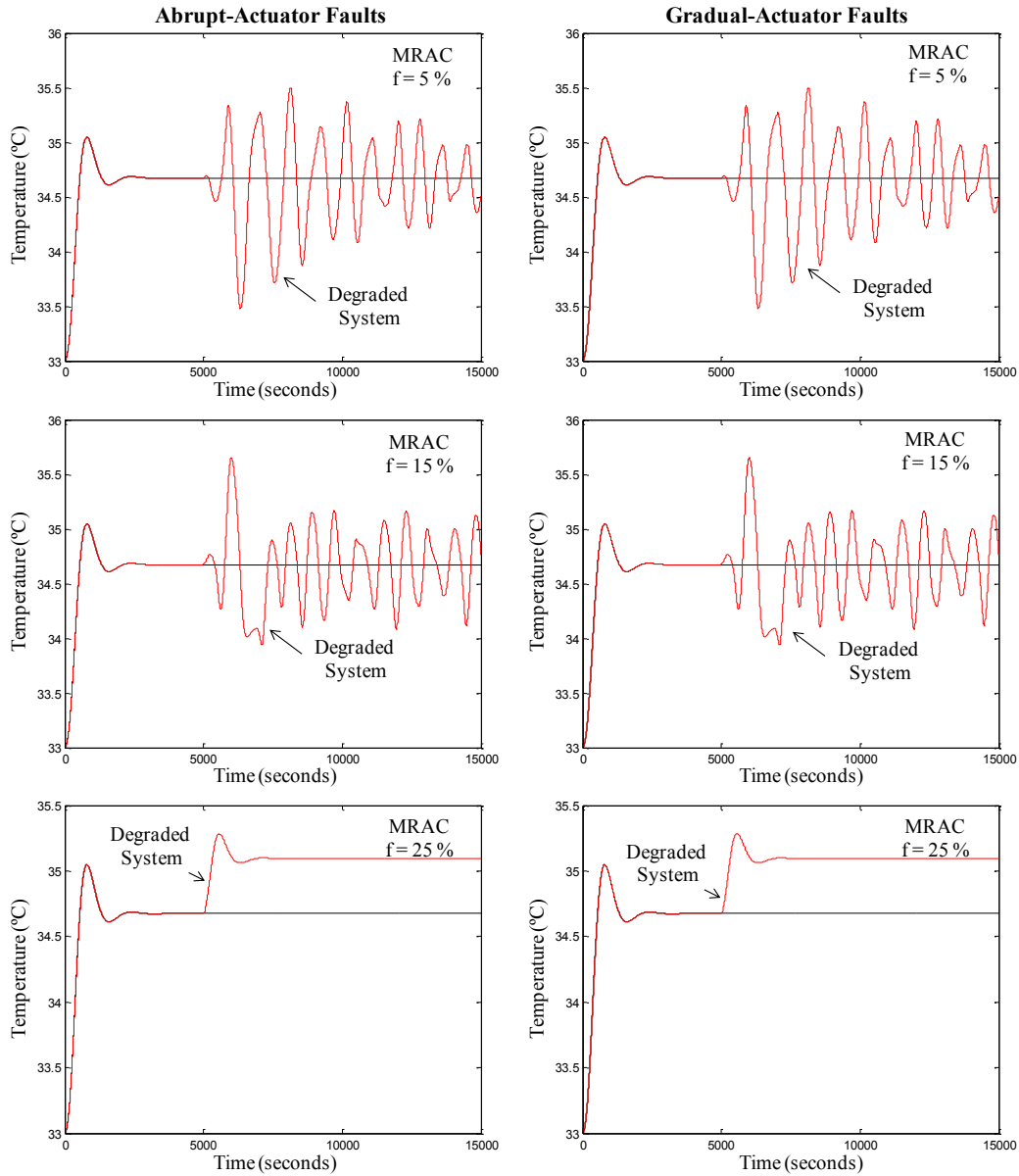
fault), fault tolerant (the system tolerates and accommodates the fault completely) and degraded system (the system does not tolerate the fault). For the gradual faults, the slope was of 10% deviation from the normal operational value per second but had a saturation block stopping this percentage at the different fault percentage values (5%, 15% and 25%). The following figures represent a comparison of the results applying different faults magnitudes.



**Figure 23. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC controller.**

From Figure 23, it can be observed that for abrupt and gradual sensor faults of 5% and 15% of system deviation the classical MRAC approach was fault tolerant against these types of faults. Fault tolerant

means that the fault is accommodated completely in the system. On the other hand, for abrupt and gradual sensor faults of 25% of system deviation the classical MRAC scheme resulted in a degraded system.



**Figure 24. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC controller.**

From Figure 24, it can be observed that for abrupt and gradual actuator faults of 5%, 15% and 25% of system deviation the classical MRAC scheme resulted in a degraded system after the occurrence of the fault. In addition, the classical MRAC scheme based on the MIT rule and a classical MRAC scheme based on Lyapunov theory were tested in a real physical implementation using Coupled-Tank system (Cruz-Reynoso, 2011). In these experiments additive faults were proved. And results showed the robustness of the MRAC FTC schemes to different fault scenarios (see Appendix A).

### 3.2 Model Reference Adaptive Controller plus PID Controller (MRAC-PID)

To overcome the limitations of the simple MRAC structure (smaller fault accommodation threshold than the MRAC controller in combination with other structures), a classical PID controller expressed in the form

$$PID = K_p + K_i/s + K_d s \quad (86)$$

is introduced in the feedforward part of the simple MRAC scheme (see Figure 25). The PID controller has the properties showed in Table 4.

**Table 4. PID Controller Parameters Properties (O'Dwyer, 2009).**

Controller Parameter	Rising Time	Overshoot	Settling Time	Steady State Error
$K_p$	Decreases	Increases	Does not influence	Decreases
$K_i$	Decreases	Increases	Increases	Eliminates
$K_d$	Does not influence	Decreases	Decreases	Does not influence

The PID parameters were obtained by using a Genetic Algorithm Pattern Search to track the desired system trajectory with the Matlab® Optimization Toolbox. The desired closed-loop behavior of the system is established through the model reference trajectory when there are no faults in the system. The parameters that need to be established for the desired optimization are shown in Table 5.

**Table 5. Matlab® Optimization Toolbox Parameters.**

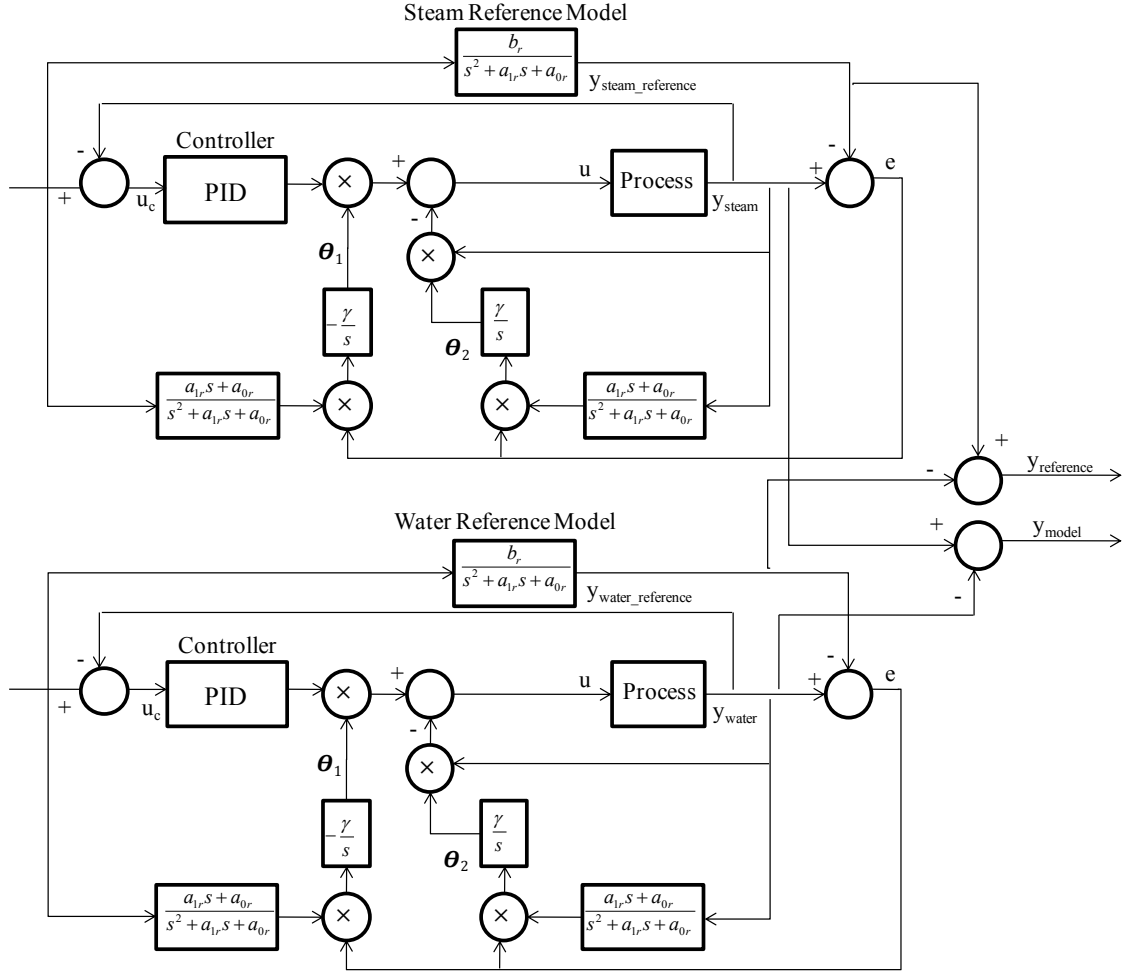
Parameters	Value
Step Initial Value	0
Step Final Value	2
Step Time	0 s
Rise time	400 s
% Rise	90
Settling Time	2500 s
% Settling	5
% Overshoot	20
% Undershoot	2

Then, the Genetic Algorithm obtains the best parameter optimization (see Table 6).

**Table 6. Obtained best PID Controller parameters using GA Optimization.**

Parameter	Steam Plant	Water Plant
Kp	1.7764	0.8276
Ki	1.2229	0.0010
Kd	0.0047	1.1475

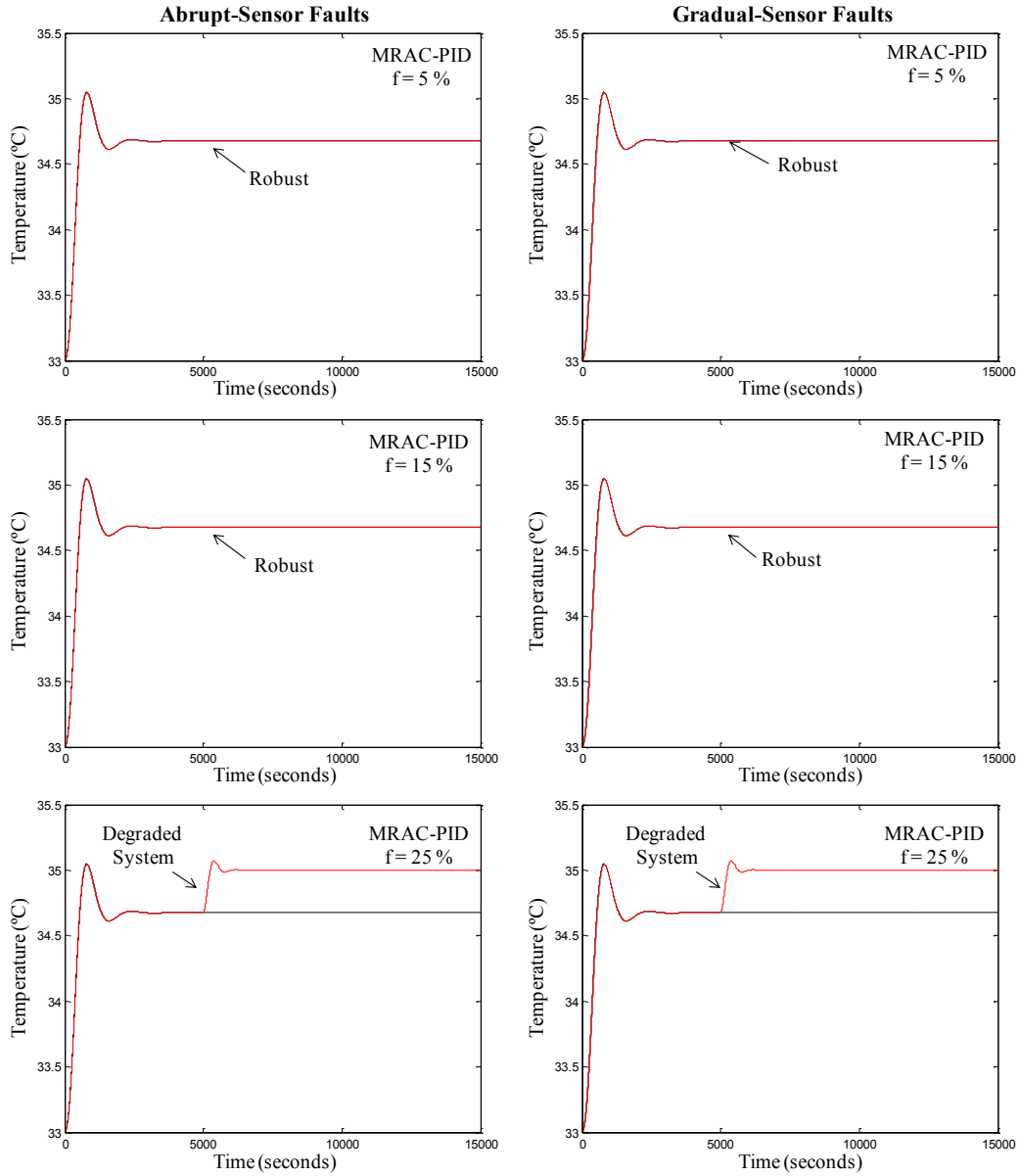




**Figure 25. Fault Tolerant MRAC-PID Controller structure**

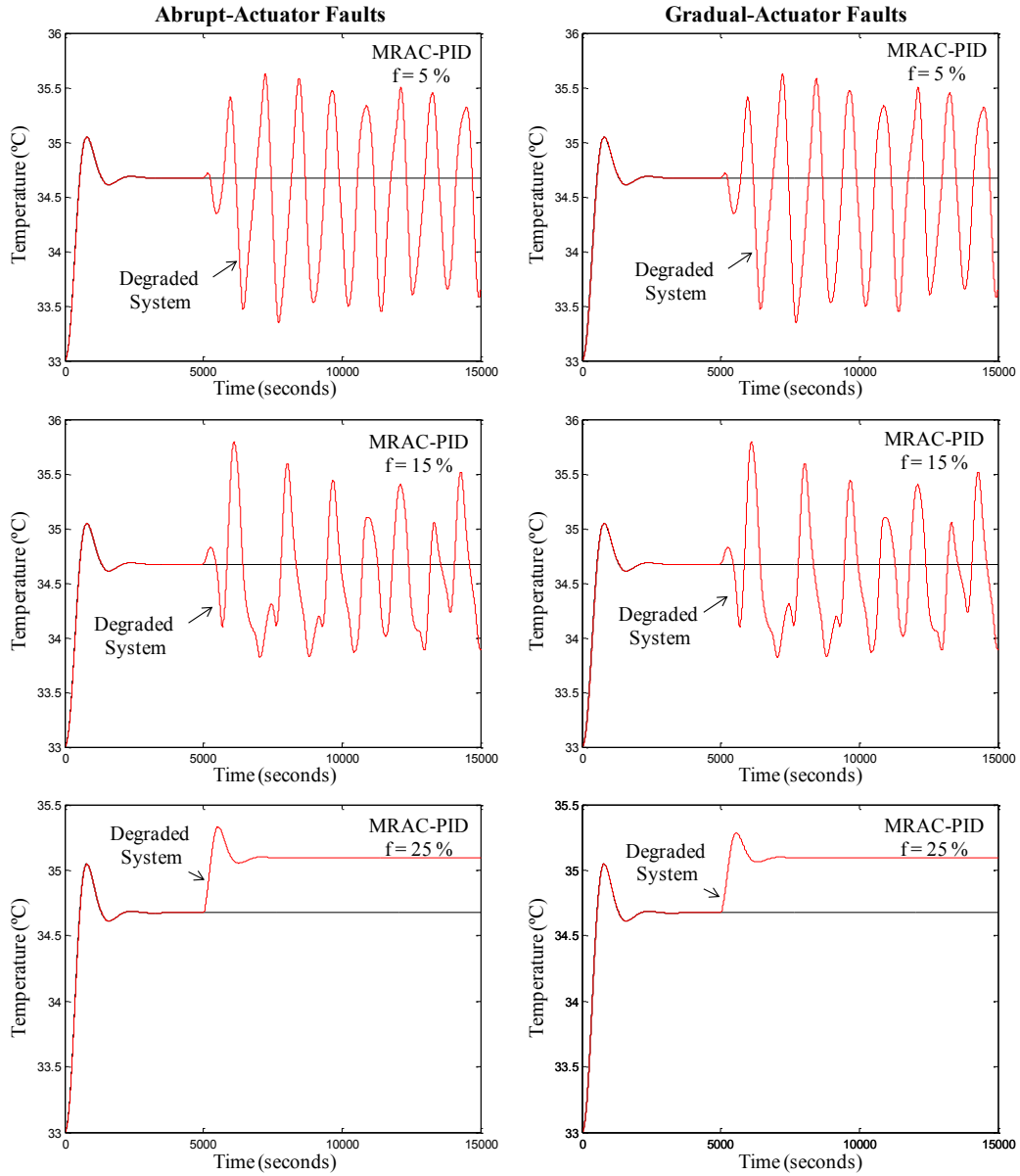
### 3.2.1 Experiments and Results

This section explains the different experiments that have been realized using the MRAC-PID controller in the Industrial Heat Exchanger. In these experiments, two different types of faults were simulated in the implemented schemes: abrupt faults and gradual faults. Both types of faults, abrupt and gradual, have been considered in sensors (feedback) and in the actuators (process entry). In each experiment, a fault was introduced at time 5000 seconds. Three different fault levels were simulated: 5%, 15% and 25%. The following figures represent a comparison of the results applying different faults magnitudes.



**Figure 26. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-PID controller.**

From Figure 26, it can be observed that for abrupt and gradual sensor faults of 5% and 15% of system deviation the MRAC controller in combination with the PID controller (MRAC-PID) resulted robust against these types of faults. On the other hand, for faults of 25% of system deviation the MRAC-PID scheme resulted in a degraded system after the occurrence of the fault.



**Figure 27. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-PID controller.**

From Figure 27, it can be observed that for abrupt and gradual actuator faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the PID controller (MRAC-PID) resulted in a degraded system after the occurrence of the fault.

### 3.3 Model Reference Adaptive Controller plus an Artificial Neural Network Controller (MRAC-ANN)

In this scheme, the PID controller was substituted by an Artificial Neural Network (see Figure 28). An ANN was selected as an additional controller because it has the capability of dealing with nonlinear system and it can be tarined to follow the ideal trajectory (non-faulty trajectory). The created ANN is a two-layer feed forward Artificial Neural Network with 20 sigmoid hidden neurons and a linear output neuron. To train the network, the Levenberg-Maquard backpropagation algorithm was used. This training algorithm is a combination of Gauss-Newton and gradient descent methods which integrates the benefits of the global and local convergence properties from the gradient descent and Gauss-Newton methods, respectively. The Artificial Neural Network was trained with the original process inputs as well as the desired outputs.

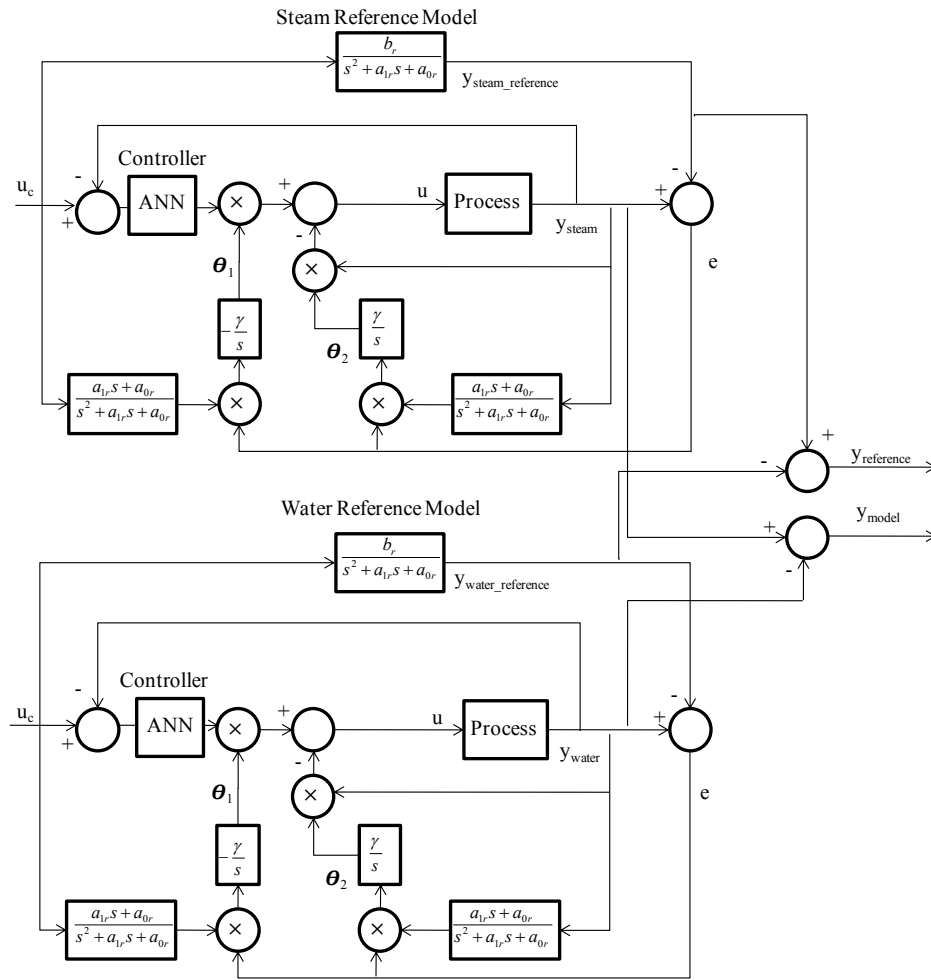


Figure 28. Fault Tolerant MRAC-ANN Controller Structure.

### 3.3.1 Experiments and Results

This section explains the different experiments that have been realized using the MRAC-ANN controller in the Industrial Heat Exchanger. In these experiments, two different types of faults were simulated in the implemented schemes: abrupt faults and gradual faults. Both types of faults, abrupt and gradual, have been considered in sensors (feedback) and in the actuators (process entry). In each experiment, a fault was introduced at time 5000 seconds. Three different fault levels were simulated: 5%, 15% and 25%. The following figures represent a comparison of the results applying different faults magnitudes.

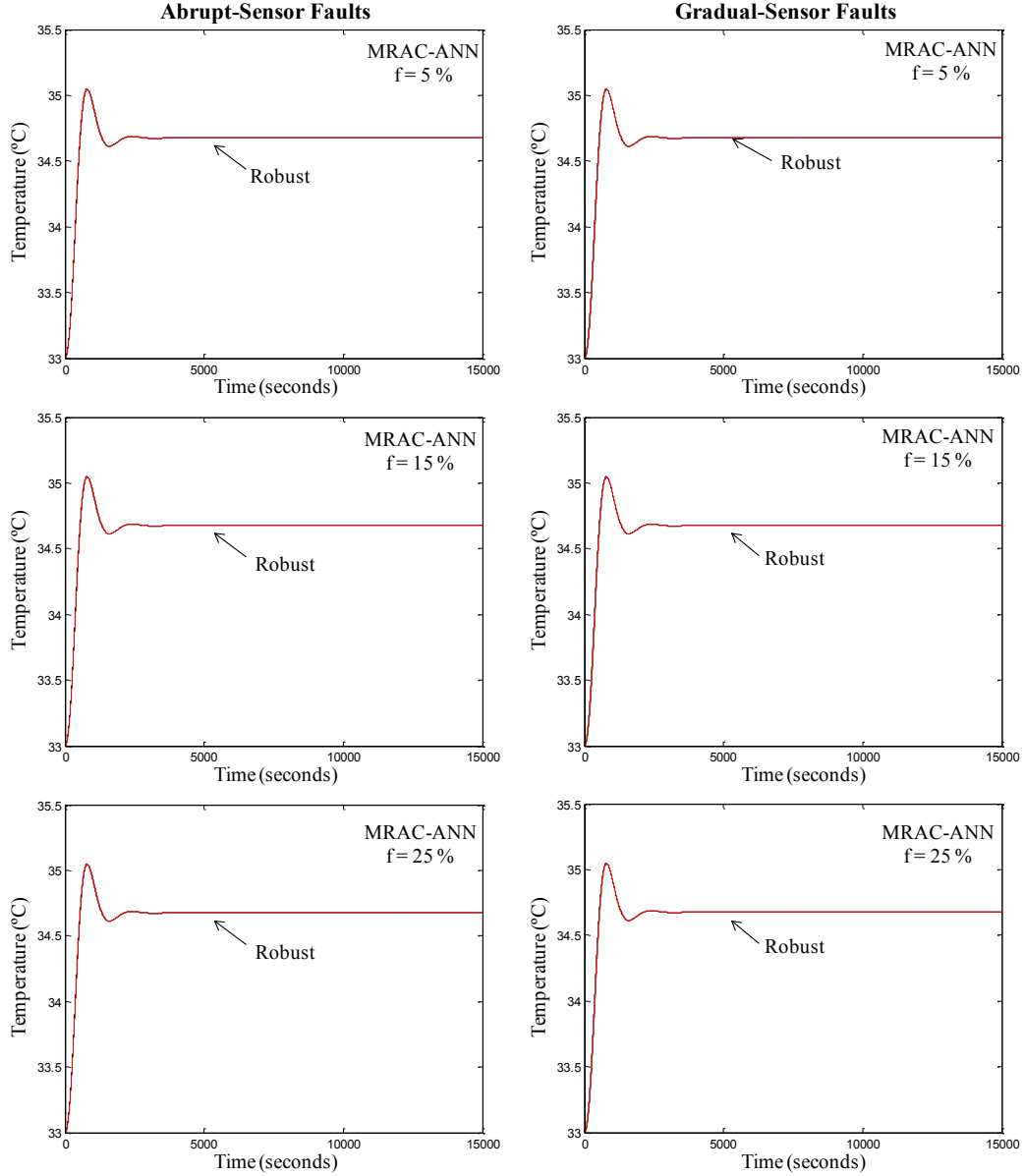
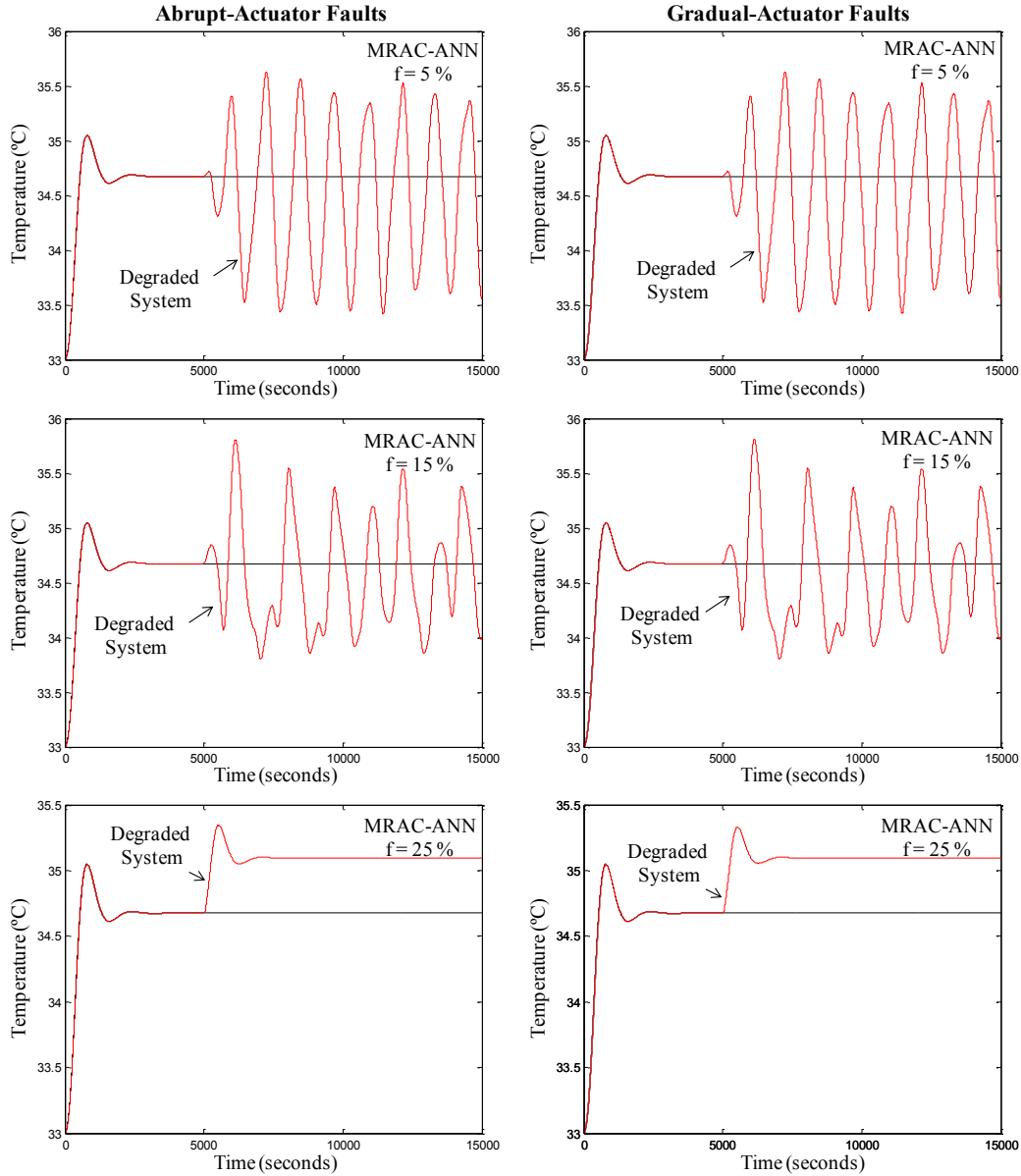


Figure 29. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-ANN controller.

From Figure 29, it can be observed that for abrupt and gradual sensor faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the ANN controller (MRAC-ANN) resulted in a robust system against these types of faults.



**Figure 30. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-ANN controller.**

From Figure 30, it can be observed that for abrupt and gradual actuator faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the ANN controller (MRAC-ANN) resulted in a degraded system after the occurrence of the fault.

### 3.4 Model Reference Adaptive Controller plus Artificial Neural Network and PID Controllers (MRAC-ANN-PID)

A combination of the MRAC-PID and the MRAC-ANN controllers were developed (see Figure 31), in order to increase the fault accommodation threshold of the system. Both controllers perform as feedforward controllers with the main intention to obtain a robust FTC structure. In this structure, the PID controller helps to attenuate the overshoot, undershoot and also helps to obtain the desired settling time and rise time. On the other hand, the Artificial Neural Network controller will try to attenuate the fault by helping the system to follow the desired reference trajectory. In addition, the controller structure adds robustness to the system, for example, the PID helps to attenuate the signal.

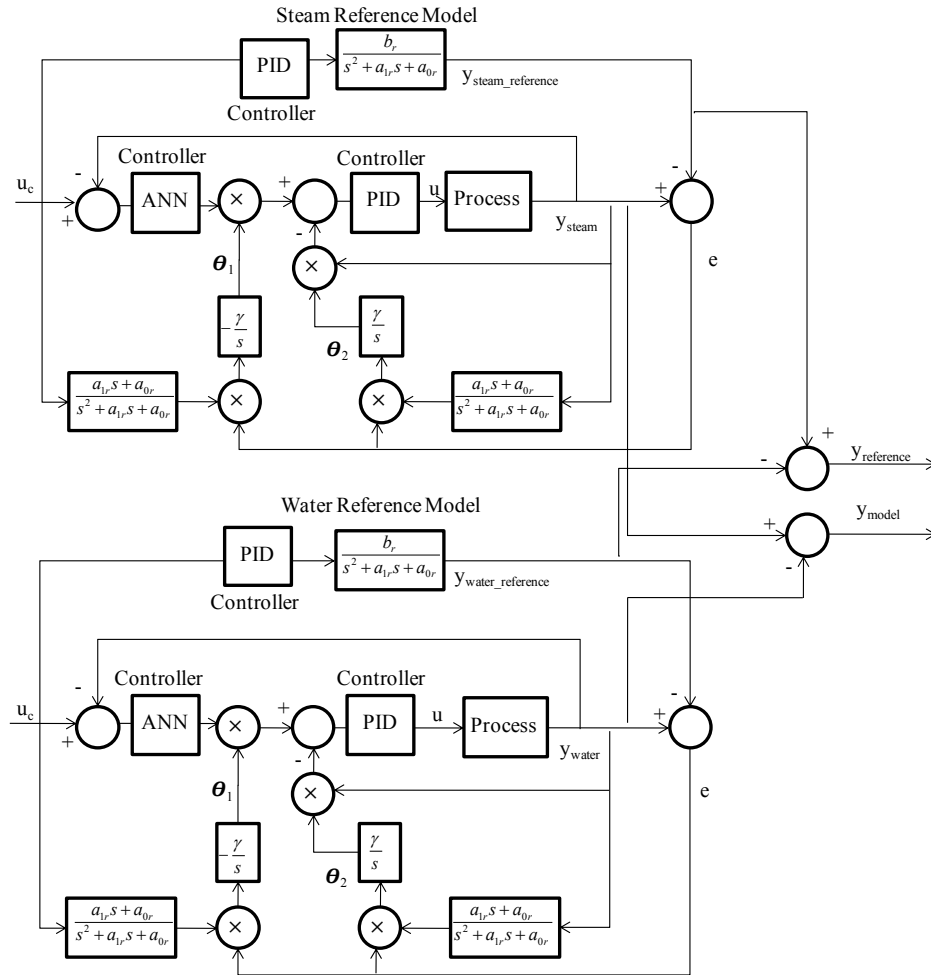
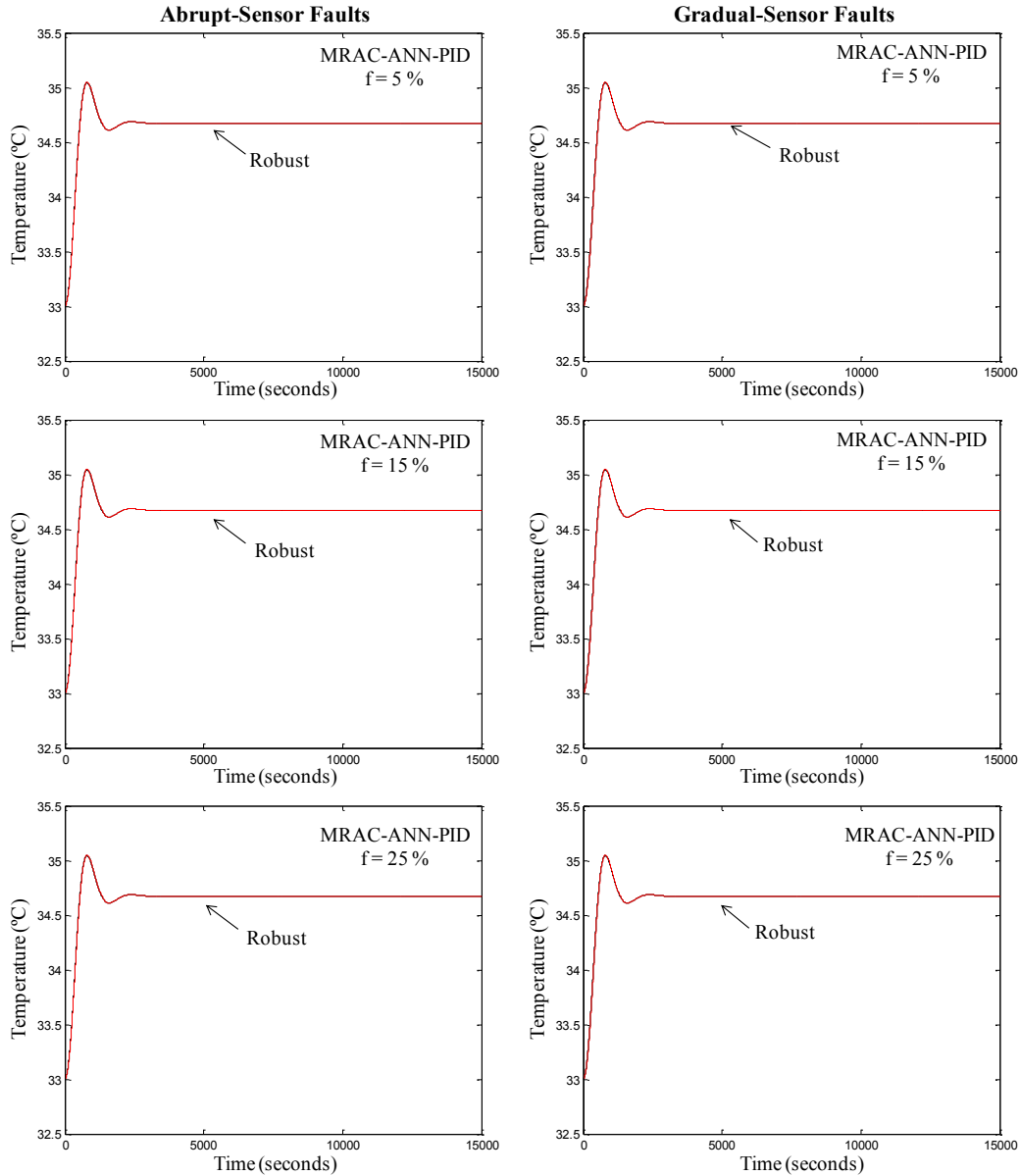


Figure 31. Fault Tolerant MRAC-ANN-PID Controller Structure

#### 3.4.1 Experiments and Results

This section explains the different experiments that have been realized using the MRAC-ANN-PID controller in the Industrial Heat Exchanger. In these experiments, two different types of faults were simulated

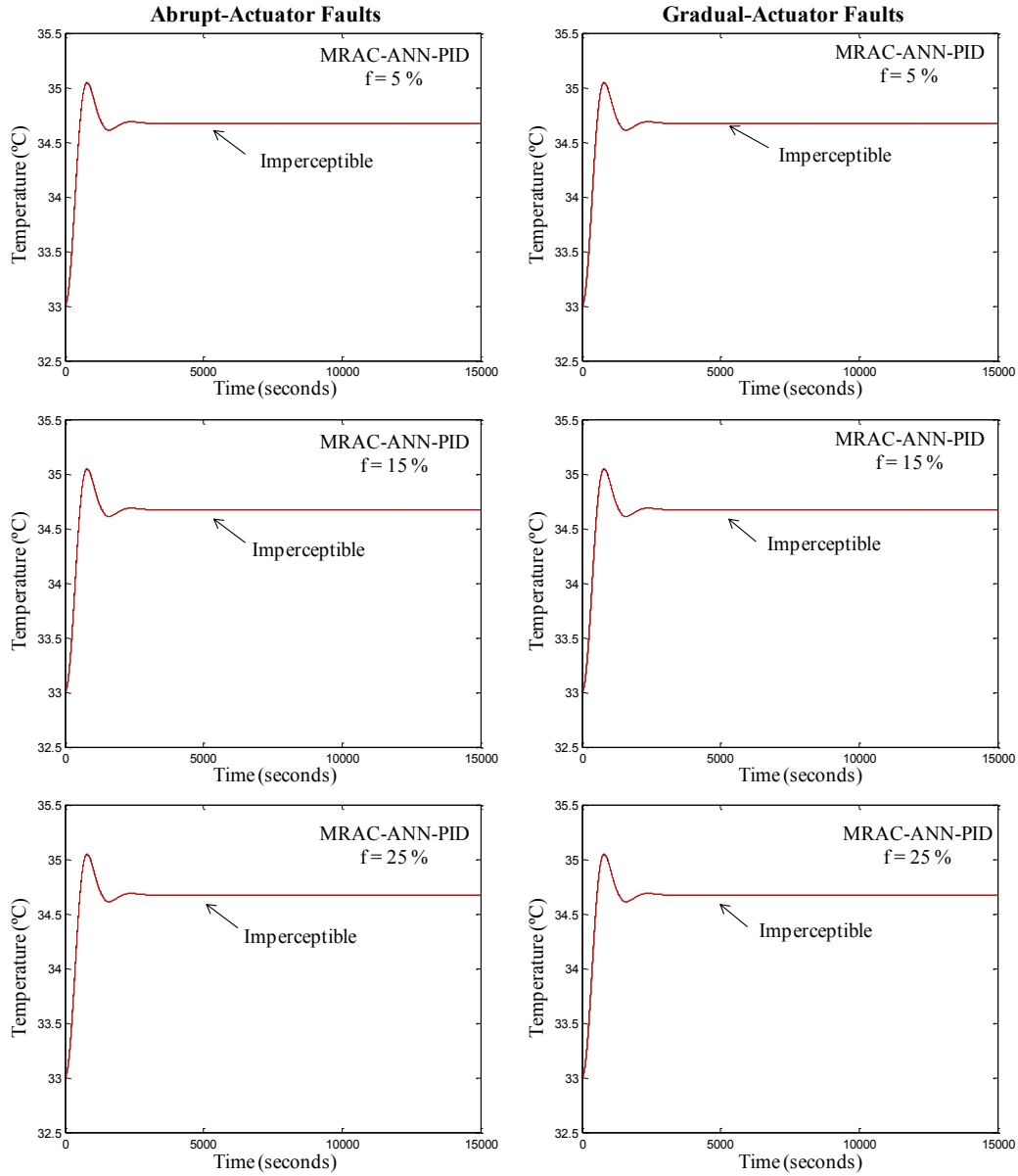
in the implemented schemes: abrupt faults and gradual faults. Both types of faults, abrupt and gradual, have been considered in sensors (feedback) and in the actuators (process entry). In each experiment, a fault was introduced at time 5000 seconds. Three different fault levels were simulated: 5%, 15% and 25%. The following figures represent a comparison of the results applying different faults magnitudes.



**Figure 32. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-ANN-PID controller.**

From Figure 32, it can be observed that for abrupt and gradual sensor faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the ANN controller and the PID controller (MRAC-ANN-PID) resulted in a robust system against these types of faults.





**Figure 33. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-ANN controller.**

From Figure 33, it can be observed that for abrupt and gradual actuator faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the ANN controller and the PID controller (MRAC-ANN-PID) resulted in an imperceptible fault system. By definition the actuator faults cannot be robust, because a change in the actuator has a physical affectation in the system, for that reason if the effect of an actuator fault is not visible in the system output, the correct terminology is imperceptible.

### 3.5 Model Reference Adaptive Controller plus an $H_\infty$ Controller (MRAC- $H_\infty$ )

The proposed  $H_\infty$  control was designed using the loop shaping method and the following steps were realized: First, the worst case of system faults (steam process faults and water process faults) were simulated and identified in the form of a Laplace function obtaining the following equations:

$$G_{steam\_faults} = \frac{0.00002}{s^2 + 0.00004309s + 0.00002} \quad (87)$$

$$G_{water\_faults} = \frac{-0.000013003}{s^2 + 0.007819304s + 0.0000819} \quad (88)$$

where  $G_{steam\_faults}$  and  $G_{water\_faults}$  represent the worst case faults in the steam process and water process, respectively. Second, the above Laplace functions are compared against the non-faulty steam and water process:

$$G_{steam} = \frac{0.00002}{s^2 + 0.004299s + 0.00002} \quad (89)$$

$$G_{water} = \frac{-0.000013}{s^2 + 0.007815s + 0.00008} \quad (90)$$

where  $G_{steam}$  is the non-faulty function describing the steam process and  $G_{water}$  is the non-faulty function describing the water process (the above has been explained in more detail in Chapter 2).

Third, a loop shaping control synthesis is performed to calculate an optimal  $H_\infty$  controller for the Laplace fault-functions (e.g.  $G_{water\_faults}$ ). This controller shapes the sigma plot of the Laplace fault-function and obtains the desired loop shaping (non-fault steam and water function) with a precision parameter called ‘‘GAM’’ (e.g. GAM should be  $\geq 1$  with GAM = 1 being a perfect match).

The next procedure is to calculate the stable-minimum-phase loop-shaping. This was realized squaring down a pre-filter  $W$ :

$$W_{steam} = C(sI - A)^{-1}B + D \quad (91)$$

$$W_{steam} = [1.311e^5 \quad 6.872e^{10} \quad 0 \quad -6.872e^{10}] \left( sI - \begin{bmatrix} -8192 & -4.295e^9 & 0 & 4.295e^9 \\ 0.003906 & -1.819e^{-12} & 0 & -0.001703 \\ 0 & 0 & -0.004299 & -0.00512 \\ 0 & 0 & 0.003906 & 4.748e^{-20} \end{bmatrix} \right)^{-1} \begin{bmatrix} -3.469e^{-18} \\ 2.367e^{-30} \\ -0.0625 \\ 0 \end{bmatrix} + [0] \quad (92)$$

$$W_{steam} = \frac{3.638e^{-12}s^3 + 1.678e^7s^2 + 2.929e^4s - 348.6}{s^4 + 8192s^3 + 1.678e^7s^2 + 7.213e^4s - 335.5} \quad (93)$$

In a manner that the shaped plant  $G_s$  is square in state space formulation:

$$G_s = G_{steam\_faults}W = C(sI - A)^{-1}B + D \quad (94)$$

$$(95)$$

$$G_s = [1.057e^{-14} \quad -167.8 \quad -6.617e^{-24} \quad 6.939e^{-18} \quad -6.617e^{-24} \quad -1.388e^{-17}] * \\ (sI - \begin{bmatrix} -8192 & -4096 & 3.815e^{-6} & 1 & 4.235e^{-22} & 2 & -1 & 0 \\ 4096 & -1.819e^{-12} & 2.541e^{-21} & 4.441e^{-16} & -2.118e^{-22} & 4.441e^{-16} & -7.175e^{-43} & \\ 0 & 0 & -4.309e^{-5} & -0.002561 & 1.291e^{-26} & 0.0001775 & -9.861e^{-32} & \\ 0 & 0 & 0.007813 & -3.939e^{-13} & 4.136e^{-25} & 0.003405 & 0 & \\ 0 & 0 & 0 & 0 & -0.004299 & -0.00512 & -0.0625 & \\ 0 & 0 & 0 & 0 & 0.003906 & 4.758e^{20} & 0 & \end{bmatrix}) \begin{bmatrix} -9.861e^{-32} \\ 0 \\ -5.049e^{-29} \end{bmatrix} + [-5.049e^{-29}]$$

$$G_s = G_{steam\_faults}W = \frac{3.725e^{-9}s^4 + 5.384e^{10}s^3 + 335.5s^2 + 0.5858s - 0.00697}{s^6 + 8192s^5 + 1.678e^7s^4 + 7.285e^4s^3 + 674.3s^2 + 1.457s - 0.006712} \quad (96)$$

Then, the desired shape of  $G_{steam}$  is accomplished with high precision in the frequency range by the shaped plant. After the above, the Normalized-Coprime-Factor control synthesis is used to calculate the ideal loop-shaping controller ( $K_s$ ):

$$K_s = C(sI - A)^{-1}B + D \quad (97)$$

$$K_s = [0.1002 \quad -0.03229 \quad 0.06381 \quad 0.0001991 \quad 3.415e^{-5}] * \\ \left( sI - \begin{bmatrix} -0.006639 & 0.004199 & -0.005592 & -1.938e^{-5} & -3.324e^{-6} \\ -0.003146 & -0.0003806 & 0.002807 & 2.63e^{-6} & 4.512e^{-7} \\ -0.004052 & -0.002715 & -0.005551 & -2.675e^{-5} & -4.59e^{-6} \\ 2.125e^{-5} & 8.556e^{-5} & 1.548e^{-5} & -2648 & -1448 \\ -3.443e^{-5} & -5.083e^{-6} & 1.278e^{-5} & 1448 & -5544 \end{bmatrix} \right) \begin{bmatrix} -0.1399 \\ -0.03378 \\ -0.06457 \\ 0.0001991 \\ -3.415e^{-5} \end{bmatrix} + [1.714]$$

$$K_s = \frac{1.714s^5 + 1.404e^4s^4 + 2.876e^7s^3 + 7.557e^4s^2 + 417.2s - 0.5835}{s^5 + 8192s^4 + 1.678e^7s^3 + 2.109e^5s^2 + 665.6s - 3.773} \quad (99)$$

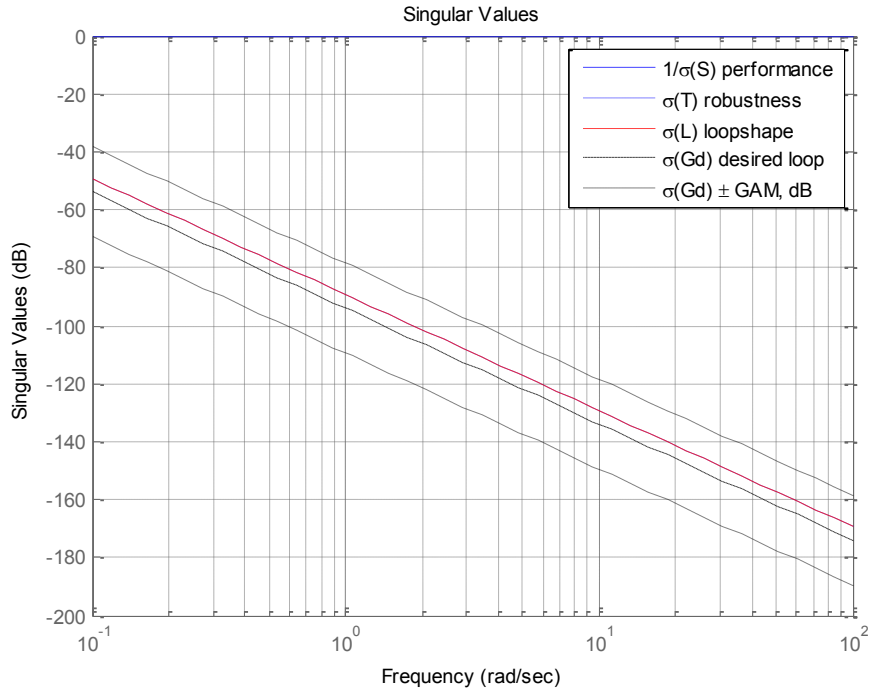
Finally, with the equation (99) the  $H_\infty$  controller is computed using:

$$H_{\infty\_steam} = W * K_s \quad (100)$$

$$(101)$$

$$H_{\infty\_steam} = \frac{-3.63e^{-12}s^8 + 2.876e^7s^7 + 2.356e^{11}s^6 + 4.825e^{14}s^5 + 2.11e^{12}s^4 + 1.924e^{10}s^3 + 3.208e^7s^2 + 1.427e^5s - 127}{s^9 + 1.638e^4s^8 + 1.007e^8s^7 + 2.749e^{11}s^6 + 2.815e^{14}s^5 + 4.749e^{12}s^4 + 3.201e^{10}s^3 + 1.821e^8s^2 + 4.954e^5s + 1266}$$

The parameter ‘‘GAM’’ which is a precision parameter for this controller is 1.9846 which is equal to 5.9535 dB. In Figure 34, the singular value plot of this controller can be viewed:



**Figure 34.  $H_{\infty}$  Controller Singular Value plot for the steam process.**

In the above figure,  $L$  is the open loop,  $T$  is the complementary sensitivity (or closed loop function) and  $S$  is the sensitivity function. It can be observed that these parameters are within the specified boundaries denoted by the singular values of the non-fault system  $\pm$  the value of GAM. The expressions of  $L$ ,  $T$  and  $S$  are, respectively:

$$L = G_{steam\_faults} * K \quad (102)$$

$$(103)$$

$$L = \frac{-1.49e^{-8}s^9 - 6.104e^{-5}s^8 + 575.3s^7 + 4.712e^6s^6 + 9.651e^9s^5 + 4.221e^7s^4 + 3.011e^5s^3 + 355.6s^2 + 1.352s - 0.00237}{s^{11} + 1.638e^4s^{10} + 1.007e^8s^9 + 2.749e^{11}s^8 + 2.815e^{14}s^7 + 4.761e^{12}s^6 + 3.784e^{10}s^5 + 2.784e^8s^4 + 1.143e^6s^3 + 4929s^2 + 9.963s + 0.02532}$$

$$S = \frac{1}{1 + G_{steam\_faults} * K} \quad (104)$$

$$(105)$$

$$S = \frac{s^{11} + 1.638e^4s^{10} + 1.007e^8s^9 + 2.749e^{11}s^8 + 2.815e^{14}s^7 + 4.761e^{12}s^6 + 3.784e^{10}s^5 + 2.784e^8s^4 + 1.143e^6s^3 + 4929s^2 + 9.963s + 0.02532}{s^{11} + 1.638e^4s^{10} + 1.007e^8s^9 + 2.749e^{11}s^8 + 2.815e^{14}s^7 + 4.761e^{12}s^6 + 4.749e^{10}s^5 + 3.206e^8s^4 + 1.445e^6s^3 + 5284s^2 + 11.32s + 0.02295}$$

$$T = \frac{G_{steam\_faults} * K}{1 + G_{steam\_faults} * K} \quad (106)$$

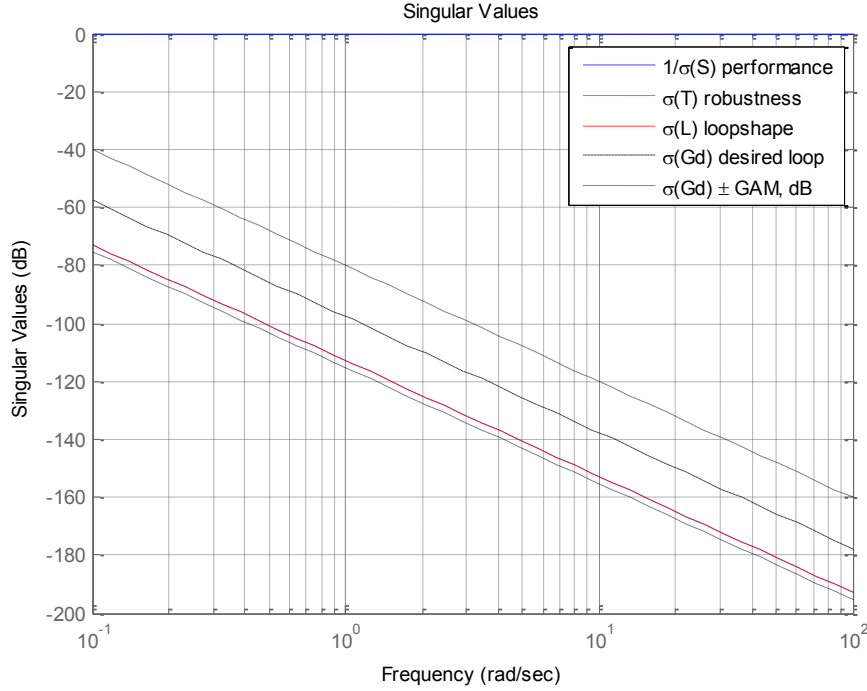
$$(107)$$

$$T = \frac{-1.49e^{-8}s^9 - 6.104e^{-5}s^8 + 575.3s^7 + 4.712e^6s^6 + 9.651e^9s^5 + 4.221e^7s^4 + 3.011e^5s^3 + 355.6s^2 + 1.352s - 0.00237}{s^{11} + 1.638e^4s^{10} + 1.007e^8s^9 + 2.749e^{11}s^8 + 2.815e^{14}s^7 + 4.761e^{12}s^6 + 4.749e^{10}s^5 + 3.206e^8s^4 + 1.445e^6s^3 + 5284s^2 + 11.32s + 0.02295}$$

The above procedure is repeated for the  $G_{water\_faults}$  process and the following  $H_\infty$  controller ( $H_{\infty\_water}$ ) is obtained:

$$H_{\infty\_water} = \frac{2.94e^6s^5 + 2.408e^{10}s^4 + 4.932e^{13}s^3 + 4.327e^{11}s^2 + 4.314e^9s - 2.38e^6}{s^7 + 1.638e^4s^6 + 1.007e^8s^5 + 2.749e^{11}s^4 + 2.815e^{14}s^3 + 5.932e^{12}s^2 + 5.169e^{10}s + 2.986e^8} \quad (108)$$

The parameter GAM for this controller is 1.0152 is equivalent to 0.1314 dB. Figure 35 presents the singular value plot of this controller.



**Figure 35.  $H_\infty$  Controller Singular Value plot for the water process.**

The open loop function  $L$ , the complementary sensitivity or closed loop function  $T$ , and the sensitivity function  $S$  are shown in equations (109) to (114). It can be observed that these parameters are within the specified boundaries denoted by the singular values of the non-fault system  $\pm$  the value of GAM.

$$L = G_{water\_faults} * K \quad (109)$$

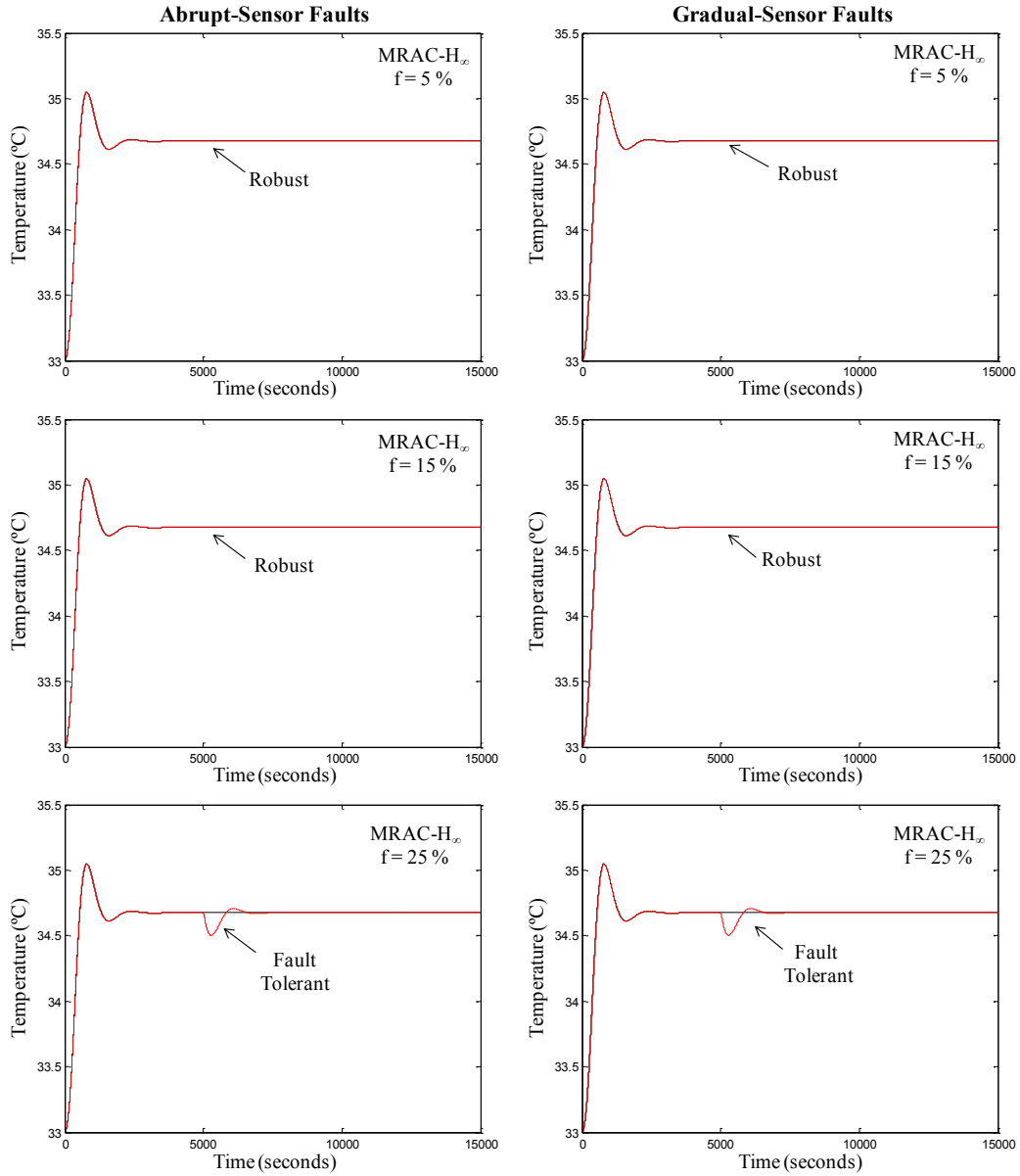
$$L = \frac{-6.1042e^{-5}s^6 + 38.19s^5 + 3.132e^5s^4 + 6.414e^8s^3 + 5.626e^6s^2 + 5.602e^4s + 48.82}{s^9 + 1.638e^4s^8 + 1.007e^8s^7 + 2.749e^{11}s^6 + 2.815e^{14}s^5 + 8.133e^{12}s^4 + 1.206e^{11}s^3 + 1.177e^9s^2 + 6.471e^6s + 2.389e^4} \quad (110)$$

$$S = \frac{1}{1 + G_{water\_faults} * K} \quad (111)$$

$$S = \frac{s^9 + 1.638e^4s^8 + 1.007e^8s^7 + 2.749e^{11}s^6 + 2.815e^{14}s^5 + 8.133e^{12}s^4 + 1.206e^{11}s^3 + 1.177e^9s^2 + 6.471e^6s + 2.389e^4}{s^9 + 1.638e^4s^8 + 1.007e^8s^7 + 2.749e^{11}s^6 + 2.815e^{14}s^5 + 8.133e^{12}s^4 + 1.212e^{11}s^3 + 1.183e^9s^2 + 6.527e^6s + 2.394e^4} \quad (112)$$

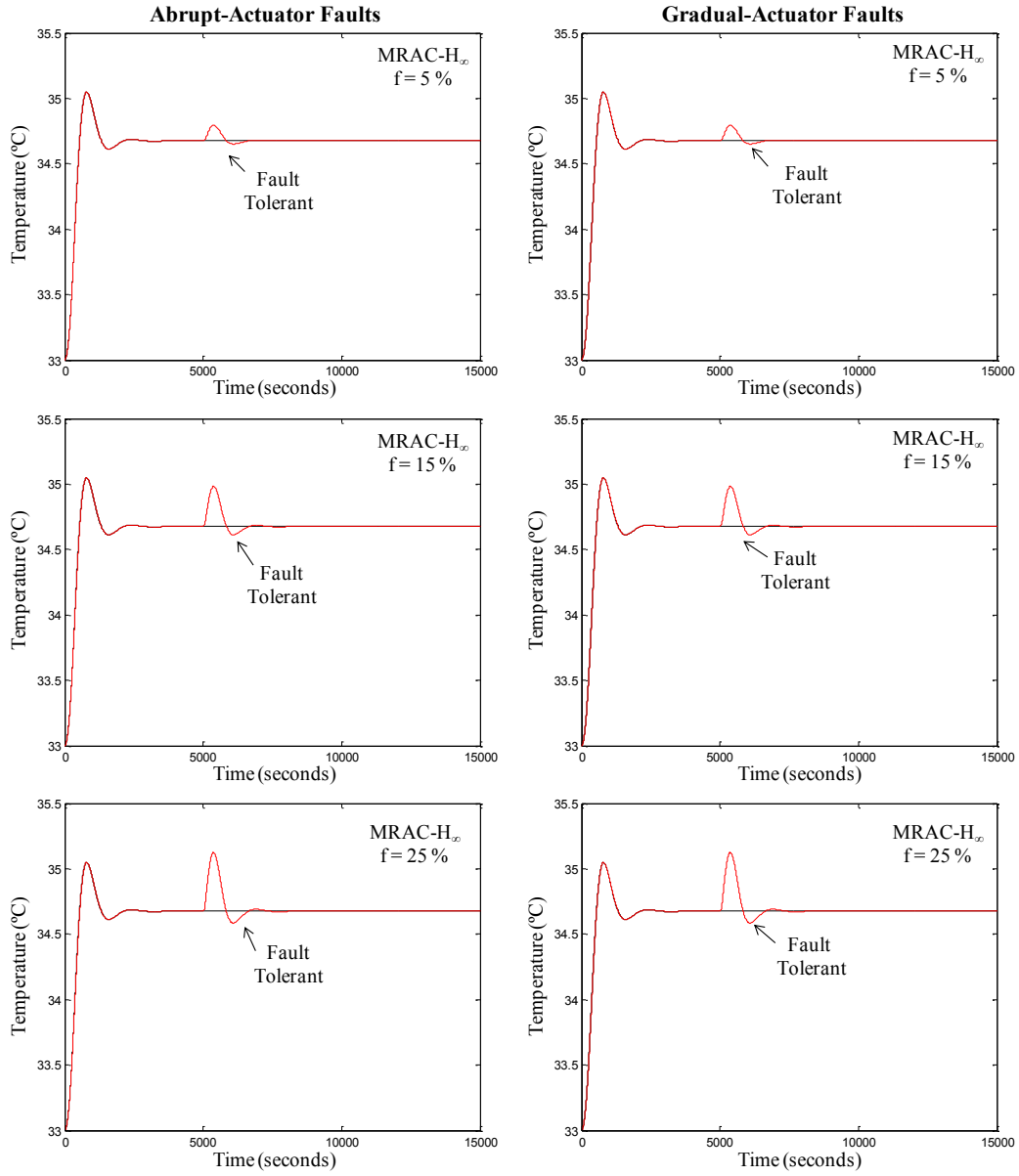
$$T = \frac{G_{water\_faults} * K}{1 + G_{water\_faults} * K} \quad (113)$$





**Figure 37. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC- $H_{\infty}$  controller.**

From Figure 37, it can be observed that for abrupt and gradual sensor faults of 5% and 15% of system deviation the MRAC controller in combination with the  $H_{\infty}$  controller (MRAC-  $H_{\infty}$ ) resulted in a robust system against these types of faults. On the other hand, for faults of 25% of system deviation the MRAC-  $H_{\infty}$  controller was fault tolerant.



**Figure 38. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-  $H_\infty$  controller.**

From Figure 38, it can be observed that for abrupt and gradual actuator faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the  $H_\infty$  controller (MRAC-  $H_\infty$ ) resulted fault tolerant against these types of faults.



### 3.6 Model Reference Adaptive Control plus Sliding Mode Control (MRAC-SMC)

Using the Sliding Mode Control theory presented in Section 2.4, the controller presented in Figure 39 is designed. The application and result of this controller is explained next.

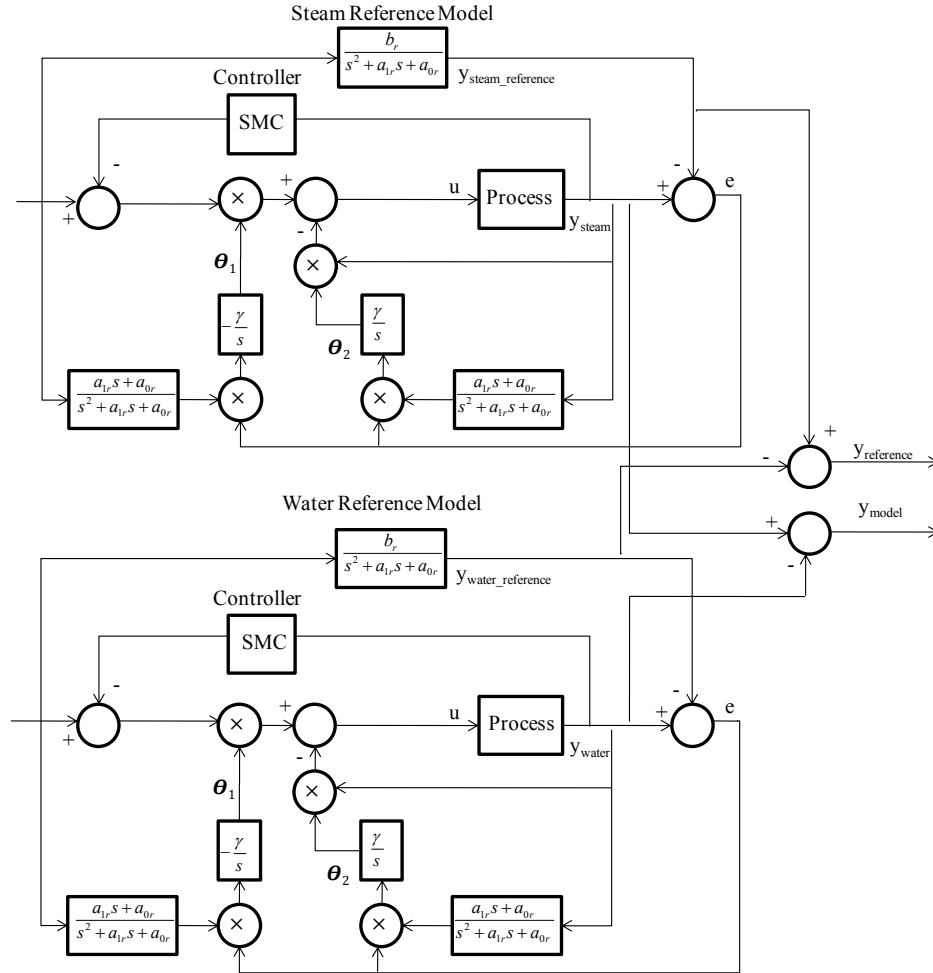
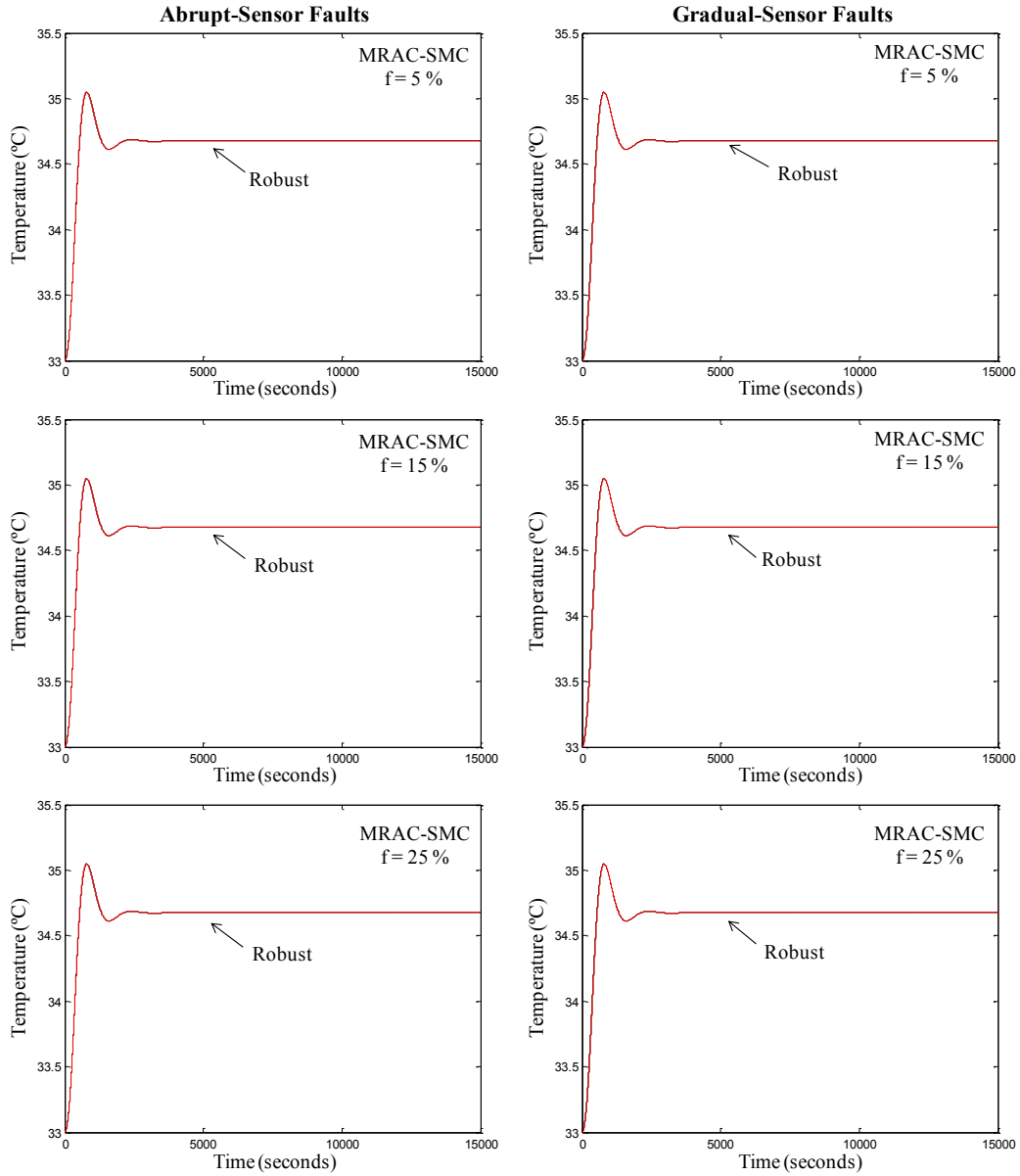


Figure 39. Fault Tolerant MRAC-SMC Controller Structure.

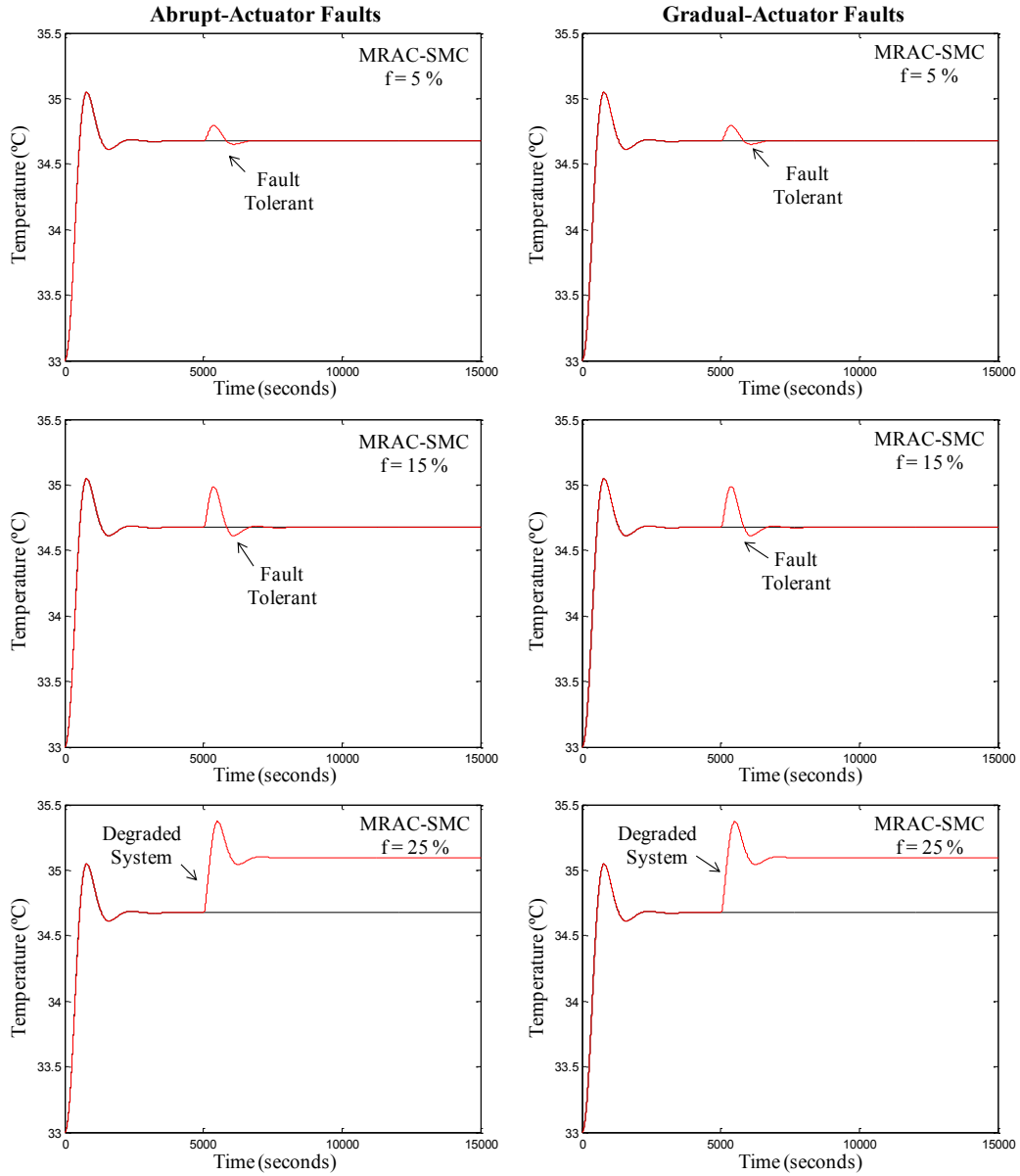
#### 3.6.1 Experiments and Results

This section explains the different experiments that have been realized using the MRAC-SMC controller in the Industrial Heat Exchanger. In these experiments, two different types of faults were simulated in the implemented schemes: abrupt faults and gradual faults. Both types of faults, abrupt and gradual, have been considered in sensors (feedback) and in the actuators (process entry). In each experiment, a fault was introduced at time 5000 seconds. Three different fault levels were simulated: 5%, 15% and 25%. The following figures represent a comparison of the results applying different faults magnitudes.



**Figure 40. Abrupt and Gradual Sensor Faults of 5%, 15% and 25% for the MRAC-SMC controller.**

From Figure 40, it can be observed that for abrupt and gradual sensor faults of 5%, 15% and 25% of system deviation the MRAC controller in combination with the SMC controller (MRAC-SMC) resulted in a robust system against these types of faults.



**Figure 41. Abrupt and Gradual Actuator Faults of 5%, 15% and 25% for the MRAC-SMC controller.**

From Figure 41, it can be observed that for abrupt and gradual actuator faults of 5% and 15% of system deviation the MRAC controller in combination with the SMC controller (MRAC-SMC) resulted fault tolerant against these types of faults. On the other hand, for faults of 25% of system deviation the system was degraded.

### 3.7 Mean Square Error for the Experiments in the Industrial Heat Exchanger

In addition, the Mean Square Error (MSE) was calculated for all the experiments, as follows:

$$MSE = \frac{(y_{reference} - y_{process})^2}{n-2} \quad (115)$$

where  $y_{reference}$  is the output of the reference model,  $y_{process}$  is the output of the actual process and  $n$  is the sampling period. The results are shown in Table 7 and Table 8 for sensor and actuator faults, respectively.

**Table 7. Mean Square Error for the Abrupt and Gradual Sensor Faults.**

Approaches	Abrupt Sensor Faults			Gradual Sensor Faults		
	f = 5%	f = 15%	f = 25%	f = 5%	f = 15%	f = 25%
MRAC-ANN-PID	1.107E-05	1.107E-05	1.107E-05	1.107E-05	1.107E-05	1.107E-05
MRAC-H <sub>∞</sub>	4.803E-06	4.803E-06	0.0007472	4.804E-06	4.804E-06	0.0007358
MRAC-SMC	1.843E-06	1.843E-06	1.843E-06	3.922E-06	3.922E-06	3.922E-06
MRAC-ANN	2.069E-05	2.069E-05	2.069E-05	2.069E-05	2.069E-05	2.069E-05
MRAC-PID	1.521E-05	1.521E-05	0.0698587	1.522E-05	1.522E-05	0.0698464
MRAC	0.0005245	0.0097135	1.8321538	0.0005246	0.0097138	1.8320687

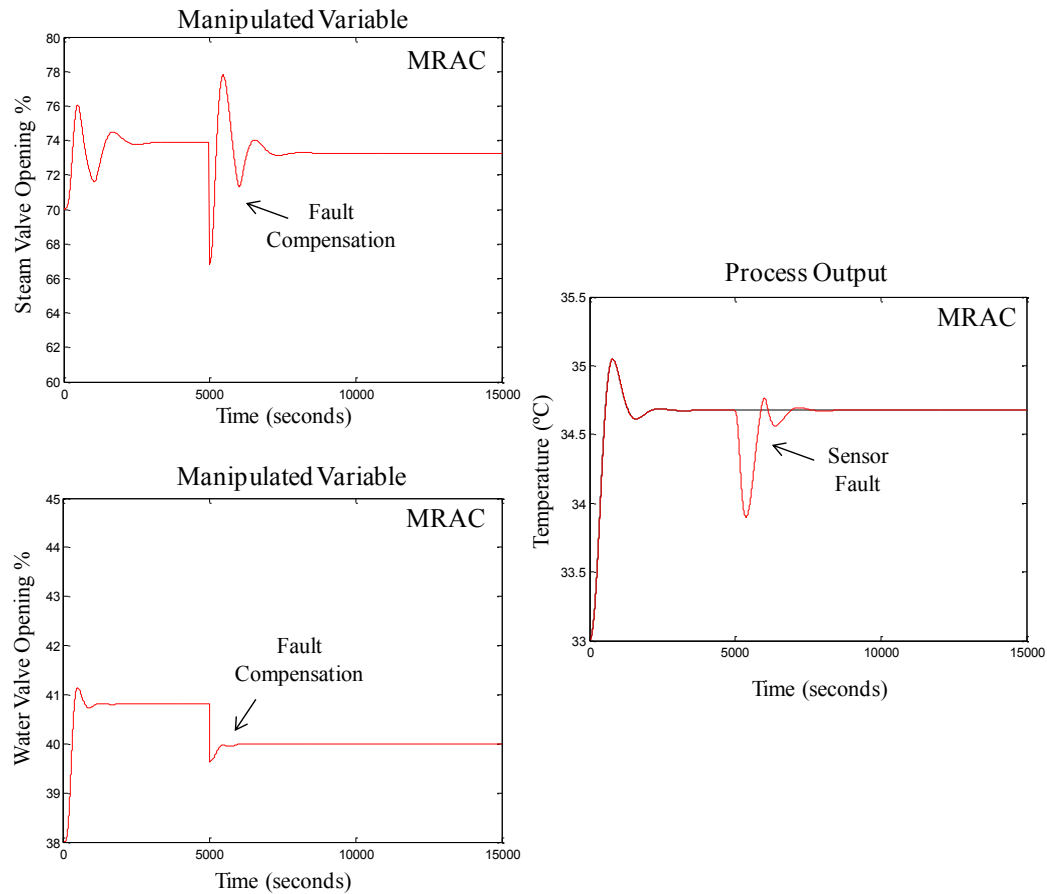
**Table 8. Mean Square Error for the Abrupt and Gradual Actuator Faults.**

Approaches	Abrupt Actuator Faults			Gradual Actuator Faults		
	f = 5%	f = 15%	f = 25%	f = 5%	f = 15%	f = 25%
MRAC-ANN-PID	1.107E-05	1.107E-05	1.107E-05	1.107E-05	1.107E-05	1.107E-05
MRAC-H <sub>∞</sub>	0.0003495	0.0023871	0.0050391	0.0003494	0.0023870	0.0050384
MRAC-SMC	0.0003501	0.0024160	0.1223235	0.0003504	0.0024173	0.1223123
MRAC-ANN	0.2993774	0.1766571	0.1210197	0.2993563	0.1766339	0.1210041
MRAC-PID	0.2973728	0.1771648	0.1205219	0.2973542	0.1771336	0.1205066
MRAC	0.1182904	0.0855958	0.1187836	0.1182899	0.0855955	0.1187681

From the above results, it can be seen that in general the MRAC-SMC has the lower MSE for the abrupt and gradual sensor faults (MSE = 1.843E-06 for the abrupt sensor faults and MSE = 3.922E-06 for the gradual sensor faults). On the other hand, the MRAC-ANN-PID scheme has the lower MSE for the abrupt and gradual actuator faults (MSE = 1.107E-05). It is important to mention that the combination of the MRAC scheme with other controllers (PID, ANN, SMC or H<sub>∞</sub>) results in an Active FTC + Passive FTC, because the MRAC can be considered as an Active FTC because this scheme accommodates the fault on-line and the other combinatorial controllers (PID, ANN, SMC or H<sub>∞</sub>) are Passive FTC because their fault accommodation capacity were design offline.

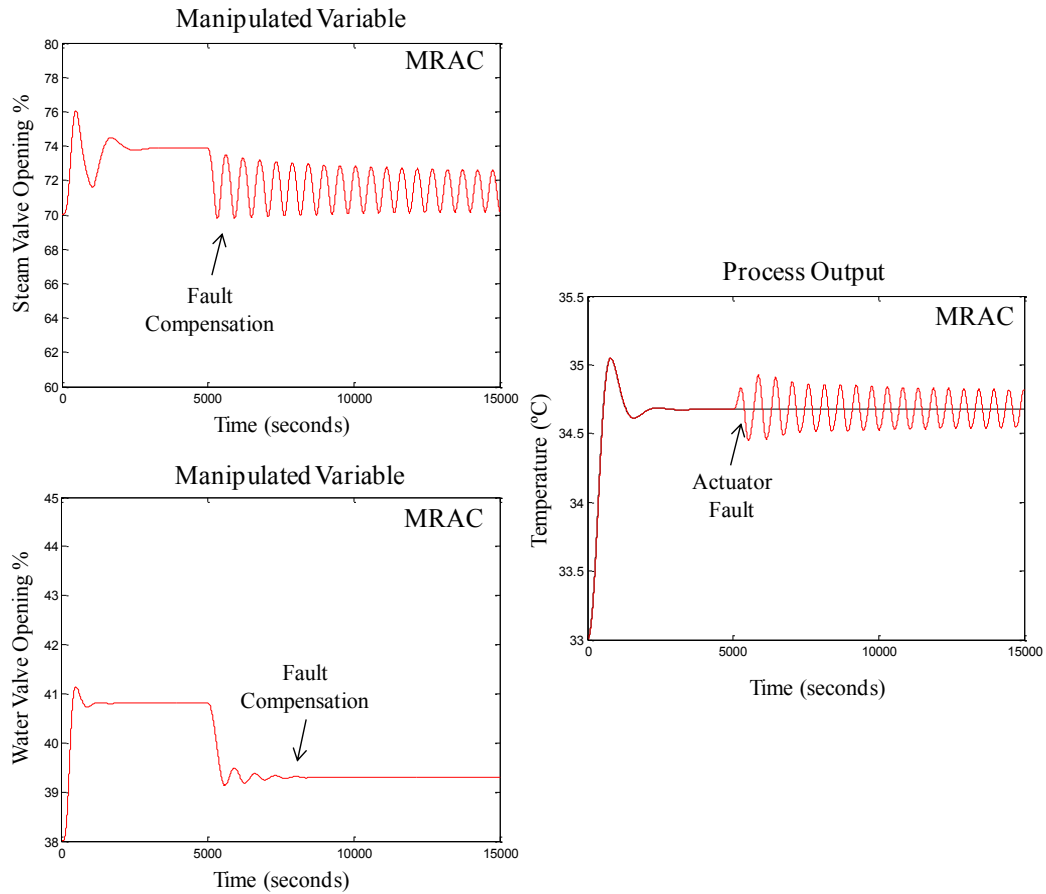
### 3.8 Manipulated Variable Analysis for the Industrial Heat Exchanger

The next figures show different experiments that were realized to analyze the manipulated variable of the Industrial Heat Exchanger.



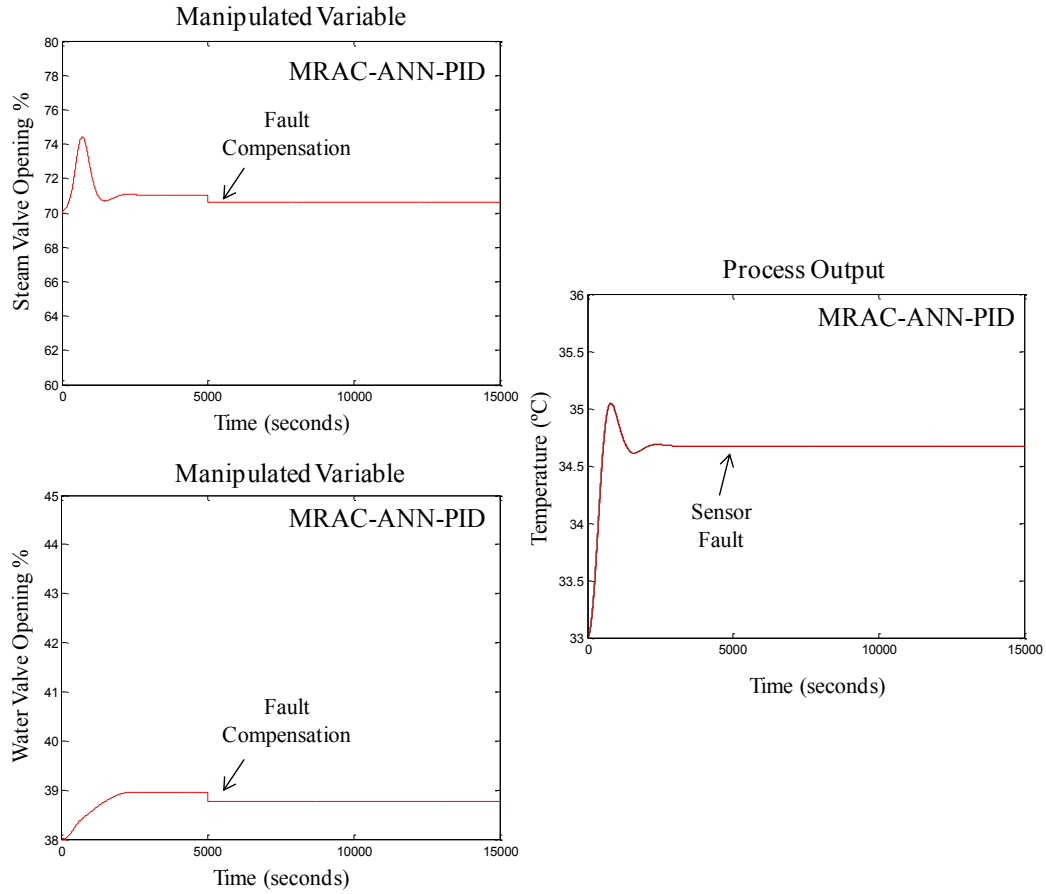
**Figure 42. Comparison between the manipulated variable and the system output of the MRAC scheme, applying an additive sensor fault of 10% at 5000 seconds.**

In the above figure, it can be observed that when the additive sensor fault appears, the manipulated variables compensate the fault. If the sensor faults decreases the valve opening % of the steam and water decreases until the fault is compensated.



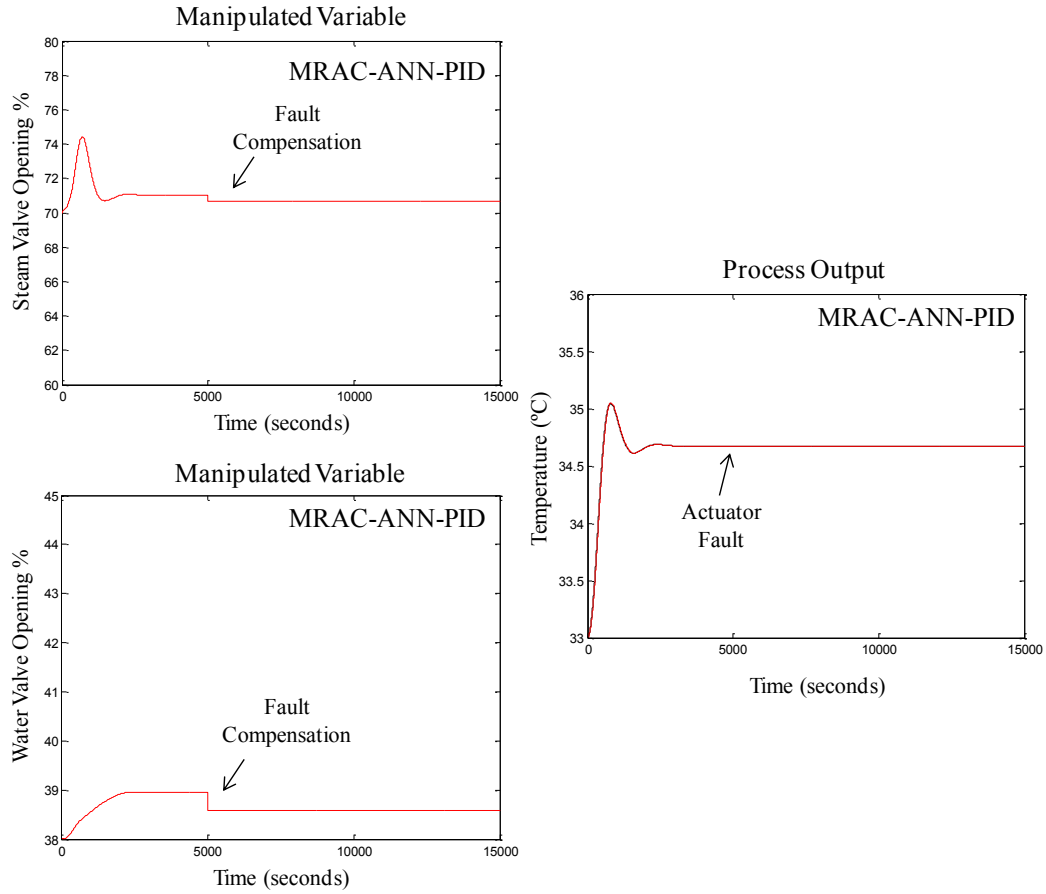
**Figure 43. Comparison between the manipulated variable and the system output of the MRAC scheme, applying an additive actuator fault of 10% at 5000 seconds.**

In the above figure, it can be observed that when the additive actuator fault appears, the system output remained with an oscillation, even though the manipulated variables tried to compensate the fault.



**Figure 44. Comparison between the manipulated variable and the system output of the MRAC-ANN-PID scheme, applying an additive sensor fault of 10% at 5000 seconds.**

In the above figure, it can be observed that the process output is robust against the sensor fault. The manipulated variables compensate the fault in order to give the desired output value.



**Figure 45. Comparison between the manipulated variable and the system output of the MRAC-ANN-PID scheme, applying an additive actuator fault of 10% at 5000 seconds.**

In the above figure, it can be observed that the additive actuator fault is imperceptible in the process output. The manipulated variables compensate the fault in order to give the desired output value.

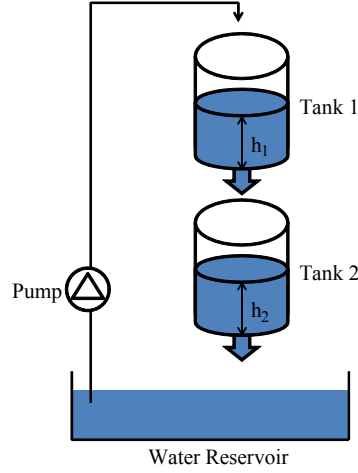


# **CHAPTER 4**

## **COUPLED-TANK SYSTEM PROPOSED SCHEMES, EXPERIMENTS AND RESULTS**

## 4 Coupled-Tank System

A second order coupled-tank system was chosen to test the Linear Parameter Varying approaches. This coupled-tank system designed by (Apkarian, 1999) is composed of two cylindrical tanks (see Figure 46): an upper and a lower tank (tank 1 and tank 2, respectively). In this system, a pump is used to transport water from the water reservoir to tank 1. Then, the outlet flow of tank 1 falls to tank 2 and finally the outlet flow of tank 2 goes to the water reservoir (Abdullah & Zribi, 2009).



**Figure 46.** Coupled-tank system designed by (Apkarian, 1999).

The water levels of the tanks are measured using pressure sensors located at the bottom of each tank. The model dynamics of these levels  $h_1(t)$  and  $h_2(t)$  can be represented as (Pan et al., 2005):

$$\dot{h}_1(t) = -\frac{a_1}{A_1} \sqrt{2g} \sqrt{h_1(t)} + \frac{k_p}{A_1} u(t) \quad (116)$$

$$\dot{h}_2(t) = \frac{a_1}{A_2} \sqrt{2g} \sqrt{h_1(t)} - \frac{a_2}{A_2} \sqrt{2g} \sqrt{h_2(t)} \quad (117)$$

$$y(t) = h_2(t) \quad (118)$$

In Table 9, the variables definition involved in the above system are explained.

**Table 9. Variable Definition**

Variable	Definition	Value
$h_1, h_2$	water level of tank 1 and tank 2	-
$A_1, A_2$	cross-section area of tank 1 and tank 2	15.5179 cm <sup>2</sup>
$a_1, a_2$	cross-section area of the outflow orifice of tank 1 and tank 2	0.1781 cm <sup>2</sup>
$U$	pump voltage	-
$k_p$	pump gain	3.3 cm <sup>3</sup> / V s
$G$	gravitational constant	981 cm/s <sup>2</sup>
$\alpha_4$	approximation constant	2.981 x 10 <sup>-7</sup>
$\alpha_3$	approximation constant	-3.659 x 10 <sup>-5</sup>
$\alpha_2$	approximation constant	1.73 x 10 <sup>-3</sup>
$\alpha_1$	approximation constant	-4.036 x 10 <sup>-2</sup>
$\alpha_0$	approximation constant	0.583

To derive an LPV model of the above system, the following procedure is followed: First, a polynomial fitting technique is used to approximate  $\sqrt{h_i}$  for  $0 \leq h_i \leq 30$  cm with  $\varphi_i h_i$ , where (Forsythe et al., 1977):

$$\varphi_i = \alpha_4 h_i^4 + \alpha_3 h_i^3 + \alpha_2 h_i^2 + \alpha_1 h_i + \alpha_0 \quad (119)$$

The parameters  $\varphi_1$  and  $\varphi_2$  are bounded with the following values taking into account the level operating ranges:

$$0.1 = \underline{\varphi}_1 \leq \varphi_1 \leq \overline{\varphi}_1 = 0.6 \quad (120)$$

$$0.1 = \underline{\varphi}_2 \leq \varphi_2 \leq \overline{\varphi}_2 = 0.6 \quad (121)$$

The LPV form of the water levels dynamic equations are written as:

$$\dot{x} = A(\varphi)x + Bu \quad (122)$$

$$y = Cx \quad (123)$$

where:

$$x = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \quad (124)$$

$$A(\varphi) = \begin{bmatrix} -0.5085\varphi_1 & 0 \\ 0.5085\varphi_1 & -0.5085\varphi_2 \end{bmatrix} \quad B = \begin{bmatrix} 0.2127 \\ 0 \end{bmatrix} \quad C = [0 \quad 1] \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (125)$$

#### 4.1 MRAC-4 Operating Points LPV Controller

To start testing the LPV model of the two tank system, a MRAC for the four extreme operating points was developed. These operating points are:  $\varphi_1=0.1$  and  $\varphi_2=0.1$ ,  $\varphi_1=0.1$  and  $\varphi_2=0.6$ ,  $\varphi_1=0.6$  and  $\varphi_2=0.1$ ,  $\varphi_1=0.6$  and  $\varphi_2=0.6$ .

For the first operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , the reference model is equal to the process model if the system is operating without a fault.

$$G_{reference\_model} = \frac{0.01082}{s^2 + 0.1017s + 0.002586} \quad (126)$$

$$G_{process\_model} = \frac{0.01082}{s^2 + 0.1017s + 0.002586} \quad (127)$$

The adaptive feed forward update rule for the MRAC based on the MIT rule ( $\theta_1$ ) is:

$$\frac{d\theta_1}{dt} = -\gamma \frac{\partial e}{\partial \theta_1} e = -\gamma \left( \frac{0.1017s + 0.002586}{s^2 + 0.1017s + 0.002586} u_c \right) e \quad (128)$$

While the adaptive feedback update rule for the MRAC based on the MIT rule ( $\theta_2$ ) is:

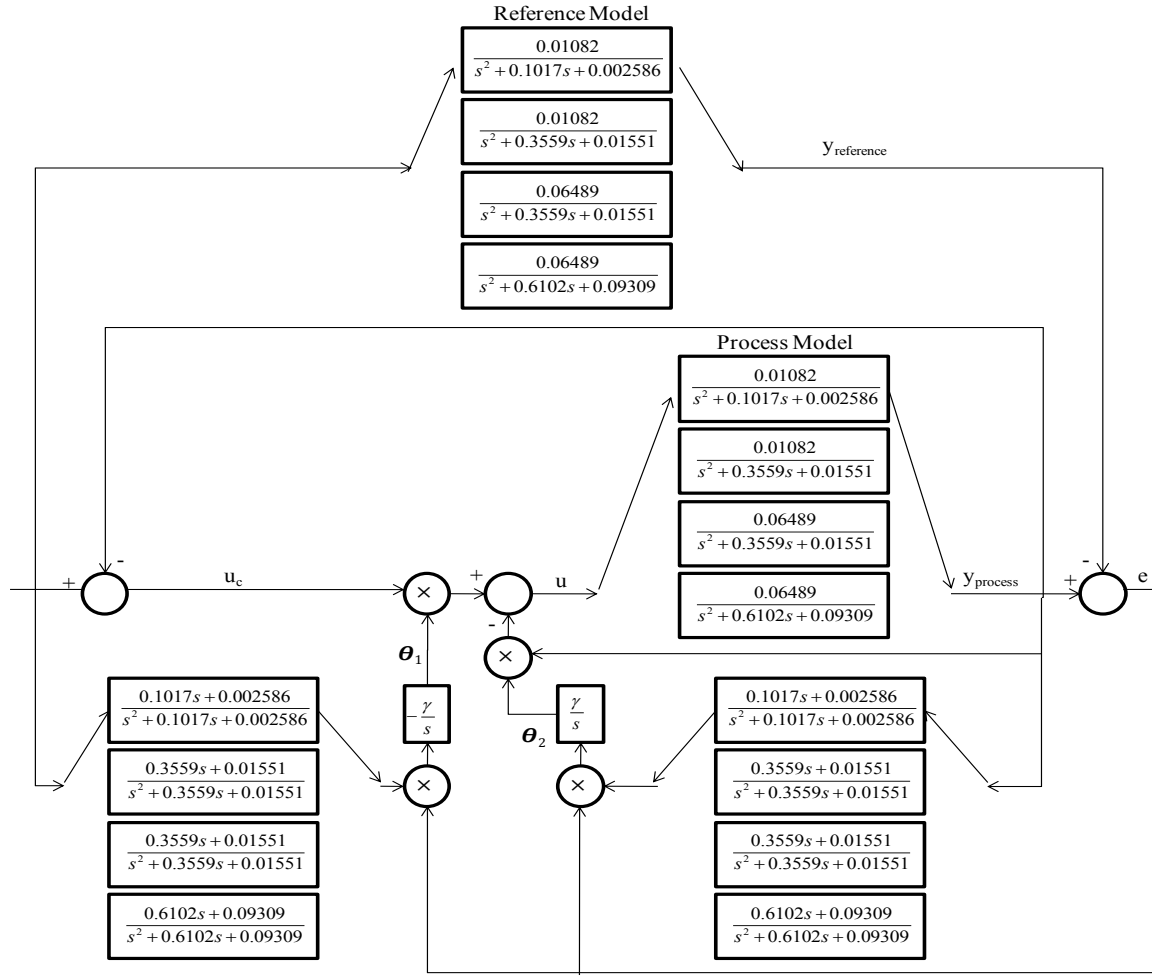
$$\frac{d\theta_2}{dt} = -\gamma \frac{\partial e}{\partial \theta_2} e = \gamma \left( \frac{0.1017s + 0.002586}{s^2 + 0.1017s + 0.002586} y_p \right) e \quad (129)$$

The second, third and fourth operating points were calculated in a similar form. In Table 10 a summary of the MRAC controllers based on the MIT rule developed for the four different operating points is presented.

**Table 10. Model Reference Adaptive Controller of the 4 Operating Points based on MIT rule.**

Operating Point	MRAC Equations for Each Operating Point		
	Reference Model = Process Model	Adaptive feed forward update rule ( $\theta_1$ )	Adaptive feedback update rule ( $\theta_2$ )
$\varphi_1=0.1$ $\varphi_2=0.1$	$\frac{0.01082}{s^2 + 0.1017s + 0.002586}$	$-\gamma \left( \frac{0.1017s + 0.002586}{s^2 + 0.1017s + 0.002586} u_c \right) e$	$\gamma \left( \frac{0.1017s + 0.002586}{s^2 + 0.1017s + 0.002586} y_p \right) e$
$\varphi_1=0.1$ $\varphi_2=0.6$	$\frac{0.01082}{s^2 + 0.3559s + 0.01551}$	$-\gamma \left( \frac{0.3559s + 0.01551}{s^2 + 0.3559s + 0.01551} u_c \right) e$	$\gamma \left( \frac{0.3559s + 0.01551}{s^2 + 0.3559s + 0.01551} y_p \right) e$
$\varphi_1=0.6$ $\varphi_2=0.1$	$\frac{0.06489}{s^2 + 0.3559s + 0.01551}$	$-\gamma \left( \frac{0.3559s + 0.01551}{s^2 + 0.3559s + 0.01551} u_c \right) e$	$\gamma \left( \frac{0.3559s + 0.01551}{s^2 + 0.3559s + 0.01551} y_p \right) e$
$\varphi_1=0.6$ $\varphi_2=0.6$	$\frac{0.06489}{s^2 + 0.6102s + 0.09309}$	$-\gamma \left( \frac{0.6102s + 0.09309}{s^2 + 0.6102s + 0.09309} u_c \right) e$	$\gamma \left( \frac{0.6102s + 0.09309}{s^2 + 0.6102s + 0.09309} y_p \right) e$

With the above equations, the MRAC controller presented in Figure 47 is implemented.



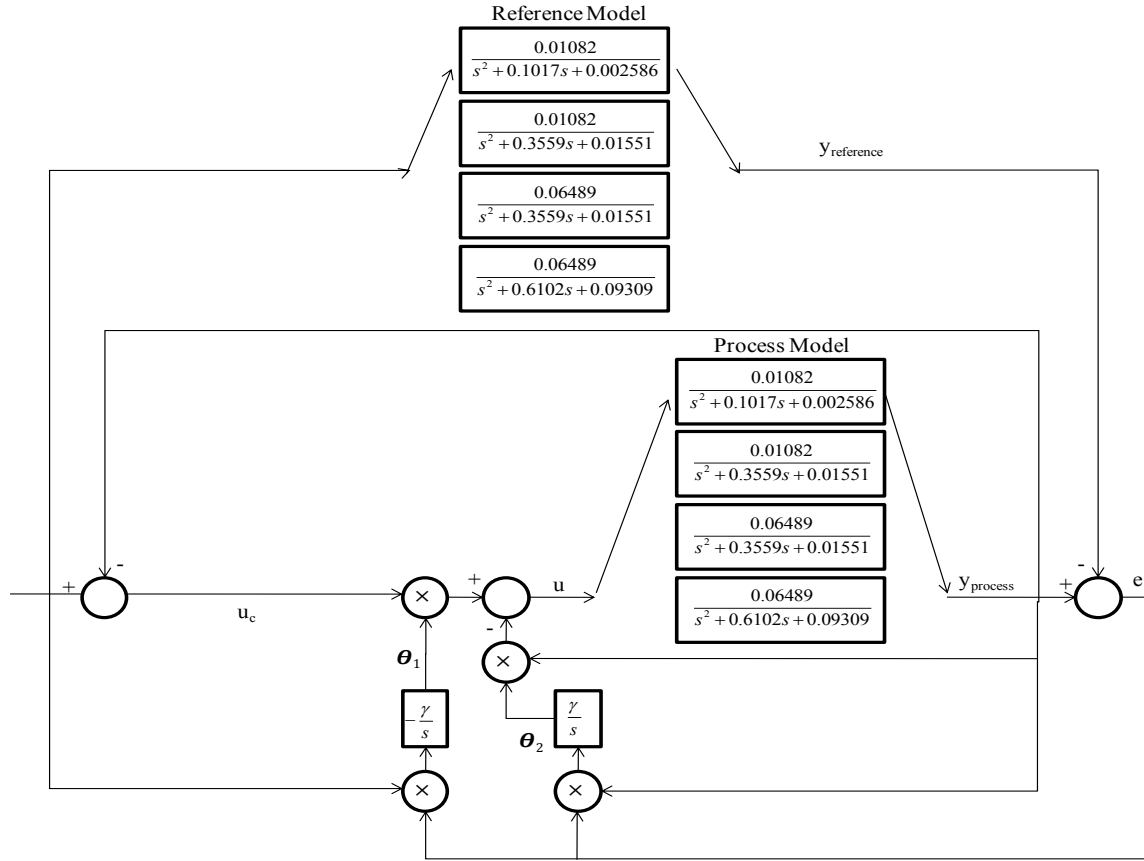
**Figure 47. MRAC-4 Operating Points LPV Controller based on MIT rule.**

On the other hand, using the MRAC controller based on Lyapunov theory the adaptive feedforward ( $\theta_1$ ) and the adaptive feedback ( $\theta_2$ ) update rules are:

$$\frac{d\theta_1}{dt} = -\gamma u_c e \quad (130)$$

$$\frac{d\theta_2}{dt} = \gamma y_p e \quad (131)$$

The above adaptation parameters are the same for the 4 different operation points. The implementation of the Lyapunov based MRAC controller is presented in Figure 48.

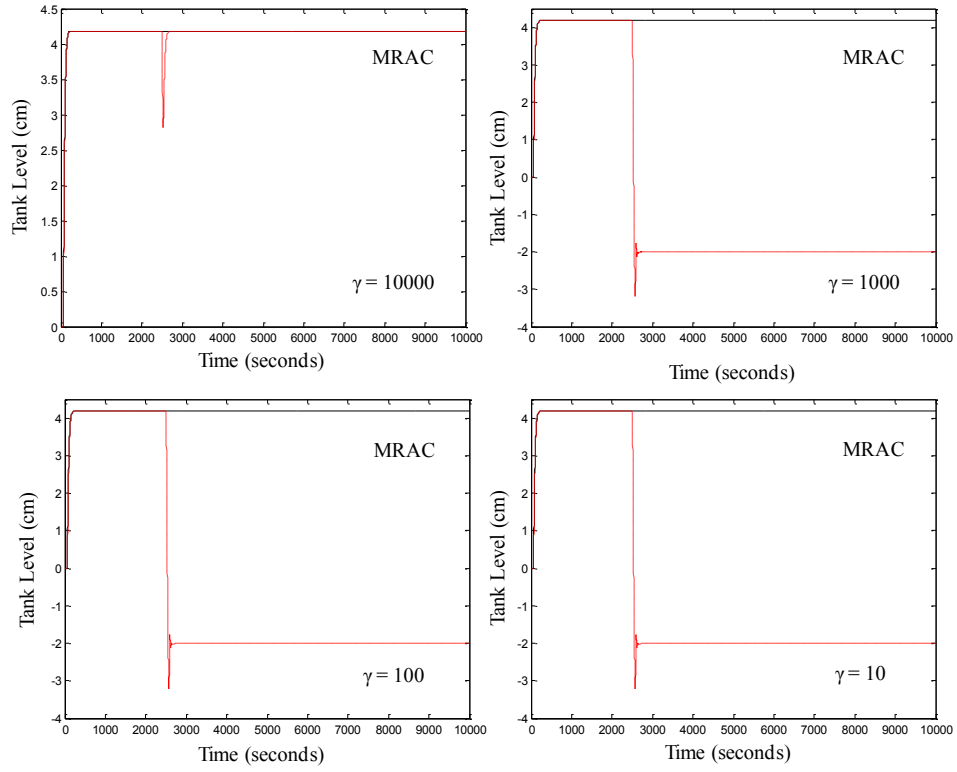


**Figure 48. MRAC-4 Operating Points LPV Controller based on Lyapunov theory.**

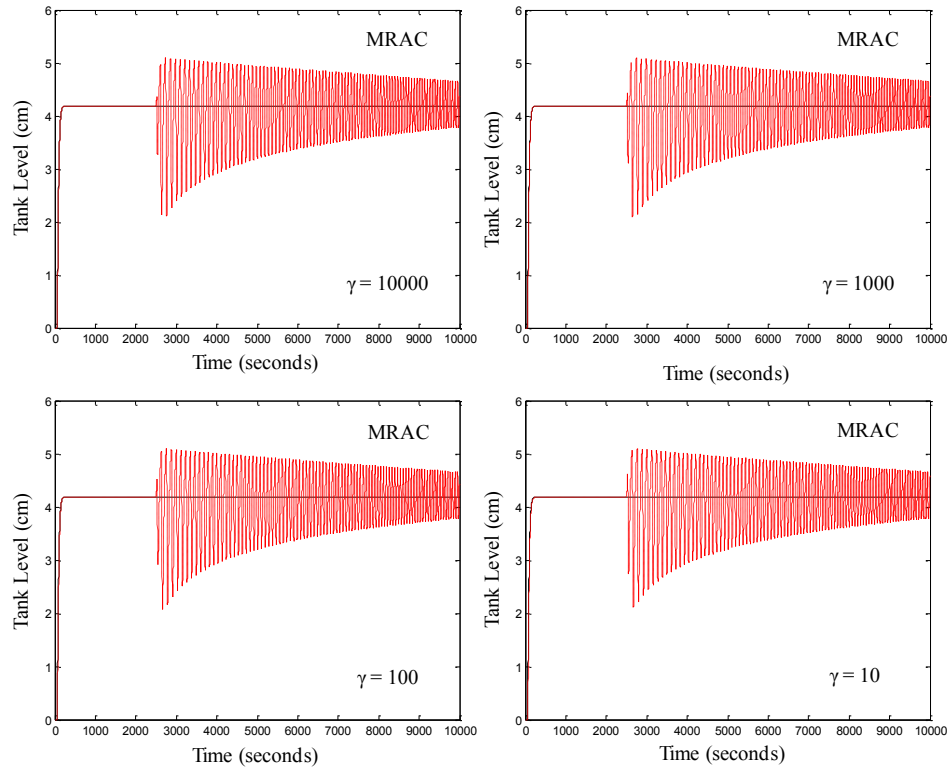
To select the value of  $\gamma$ , different experiments with different  $\gamma$  sizes were realized. In these experiments, first a sensor fault of 23.3% was introduced at time 2500 seconds. And then, an actuator fault of magnitude 3% was introduced at time 2500 seconds. Four different size of  $\gamma$  were tested (10000, 1000, 100, 10) and the results are showed in Table 11 and from Figure 49 to Figure 50.

**Table 11. MSE of different sizes of  $\gamma$ .**

$\gamma$	Sensor Fault	Actuator Fault
10000	0.0068	0.2513
1000	28.5951	0.2501
100	28.5983	0.2491
10	28.5983	0.2472



**Figure 49. MRAC Results testing different sizes of  $\gamma$  for sensor faults.**

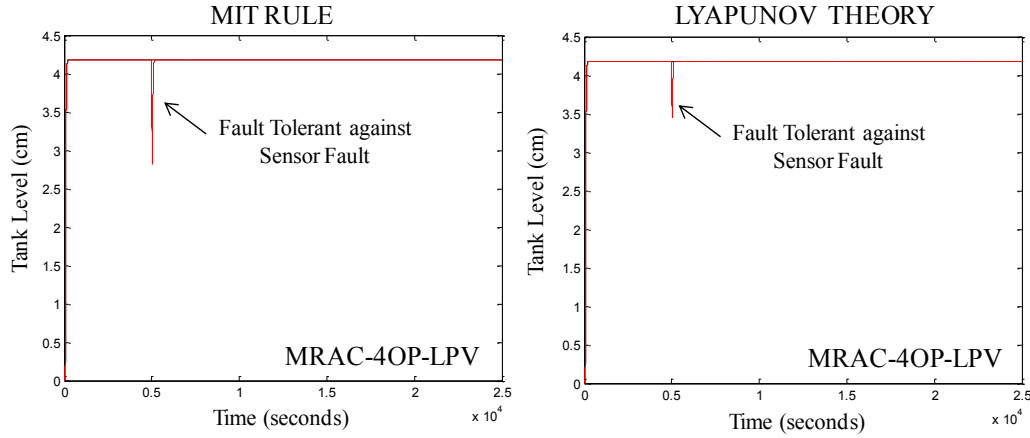


**Figure 50. MRAC Results testing different sizes of  $\gamma$  for actuator faults.**

From the above results the selected value of  $\gamma$  to realize the following experiments was 10000, because it has the lower MSE combination for sensor and actuator faults.

#### 4.1.1 Experiments and Results

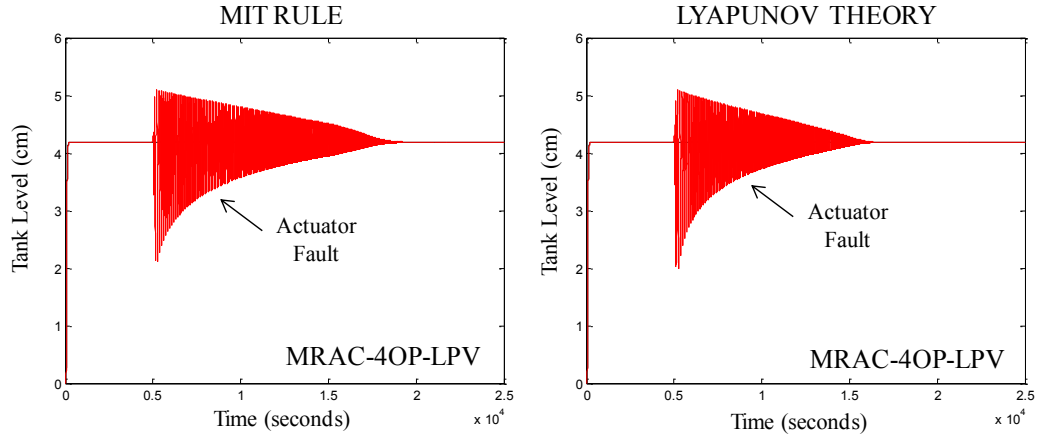
This section explains the different experiments realized in the Coupled-Tank System using the MRAC 4 Operating Points LPV Controller (MRAC-4OP-LPV) based on the MIT rule and based on the Lyapunov theory. Two different types of faults were simulated in the implemented schemes: abrupt and gradual faults. In the abrupt fault case, the whole magnitude of the fault is developed in one moment of time and was simulated with a step function. On the other hand, gradual faults are developed during a period of time and are implemented with a ramp function. Both types of faults, abrupt and gradual, can be implemented in sensors (feedback), in which the properties of the process are not affected, but the sensor readings are wrong. Also, they can be implemented in actuators (process entry) in which the process properties are not affected either, but the process behavior can change or can be interrupted. Abrupt faults in actuators represent for instance a pump stuck while in sensors a constant bias in measurements. A gradual fault in actuators could be a progressive loss of electrical power in pump, and in sensors a drift in the measurements. The next figures show the results of the experiments realized using the MRAC-4OP-LPV scheme.



**Figure 51. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-sensor fault of 23.3% at the first operating point.**

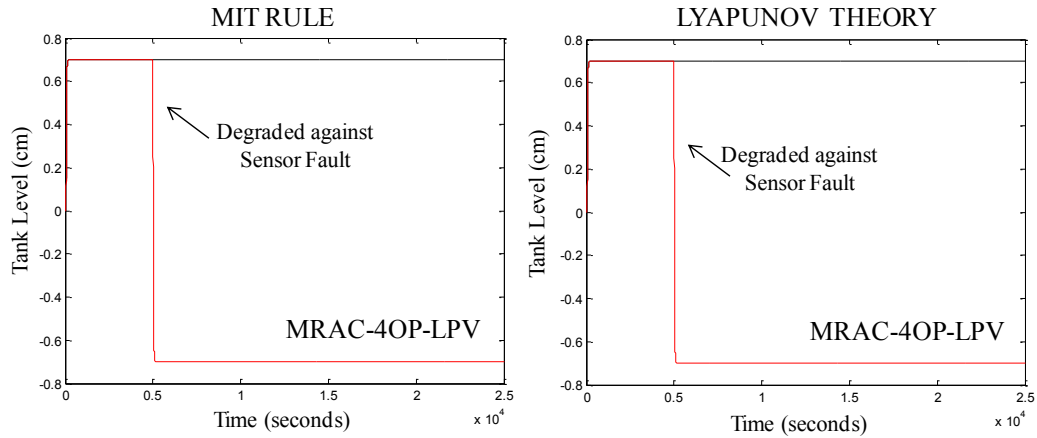
In Figure 51, the MRAC-4OP-LPV controllers based on the MIT rule and based on the Lyapunov theory are compared. The controller is working in the operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , an abrupt-sensor fault of magnitude 23.3% was introduced at time 5000 seconds. It can be observed, that the MRAC-4OP-LPV based on the MIT rule and based on the Lyapunov theory was fault tolerant against the fault.





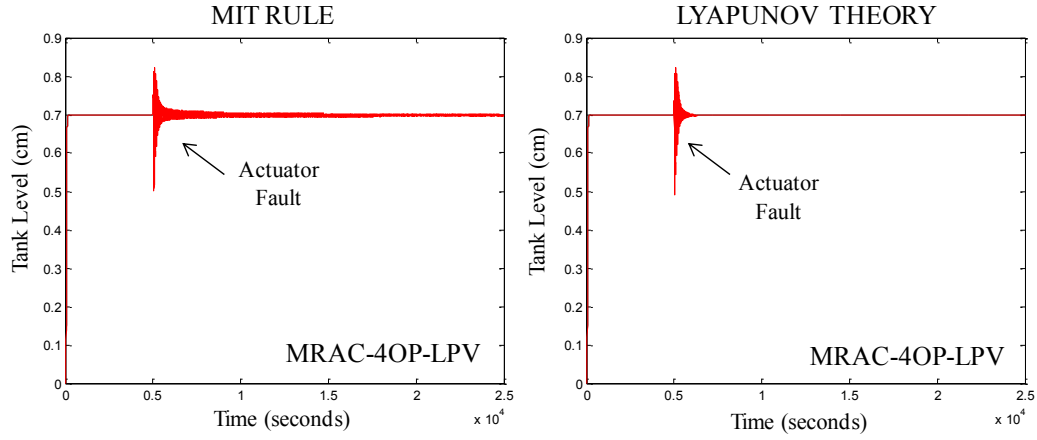
**Figure 52. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-actuator fault of 1% at the first operating point.**

In Figure 52, the MRAC-4OP-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While, the controllers are working in the operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , an abrupt-actuator fault of 1% was introduced at time 5000 seconds. In this figure the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory methods were fault tolerant against the actuator fault where the MRAC-4OP-LPV accommodates the fault in 15000 and 12500 seconds, respectively.



**Figure 53. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-sensor fault of 10% at the third operating point.**

In Figure 53, the MRAC-4OP-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While the controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , a gradual-sensor fault of 10% was introduced at time 5000 seconds. In this figure it can be observed that the MRAC-4OP-LPV was degraded for both methodologies.



**Figure 54. Comparison between the MRAC-4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-actuator fault of 1% at the third operating point.**

In Figure 54, the MRAC-4OP-LPV controllers based on MIT rule and on Lyapunov theory are compared. While the controllers are operating in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , a gradual-actuator fault of 1% was introduced at time 5000 seconds. In this figure, it can be observed that the MRAC-4OP-LPV based on Lyapunov theory was fault tolerant and could accommodate the fault in approximately 200 seconds. The MRAC-4OP-LPV based on MIT rule presented oscillations in the system after the occurrence of the actuator fault.

## 4.2 MRAC-Artificial Neural Network 4 Operating Points LPV Controller

Four Artificial Neural Networks controllers were added to the MRAC- 4 Operating Points LPV Controller. To create and train each Artificial Neural Network controller, the original process inputs were introduced as well as the desired outputs (no system fault case). The created ANN is a two-layer feedforward Artificial Neural Network with 50 sigmoid hidden neurons and a linear output neuron. To train the network the Levenberg-Maquard backpropagation algorithm was used. This training algorithm is a combination of Gauss-Newton and gradient descent methods which integrates the benefits of the global and local convergence properties from the gradient descent and Gauss-Newton methods, respectively (Ye, 2003).

The implementation of the MRAC-Artificial Neural Network 4 Operating Points LPV Controller based on MIT rule is presented in Figure 55:

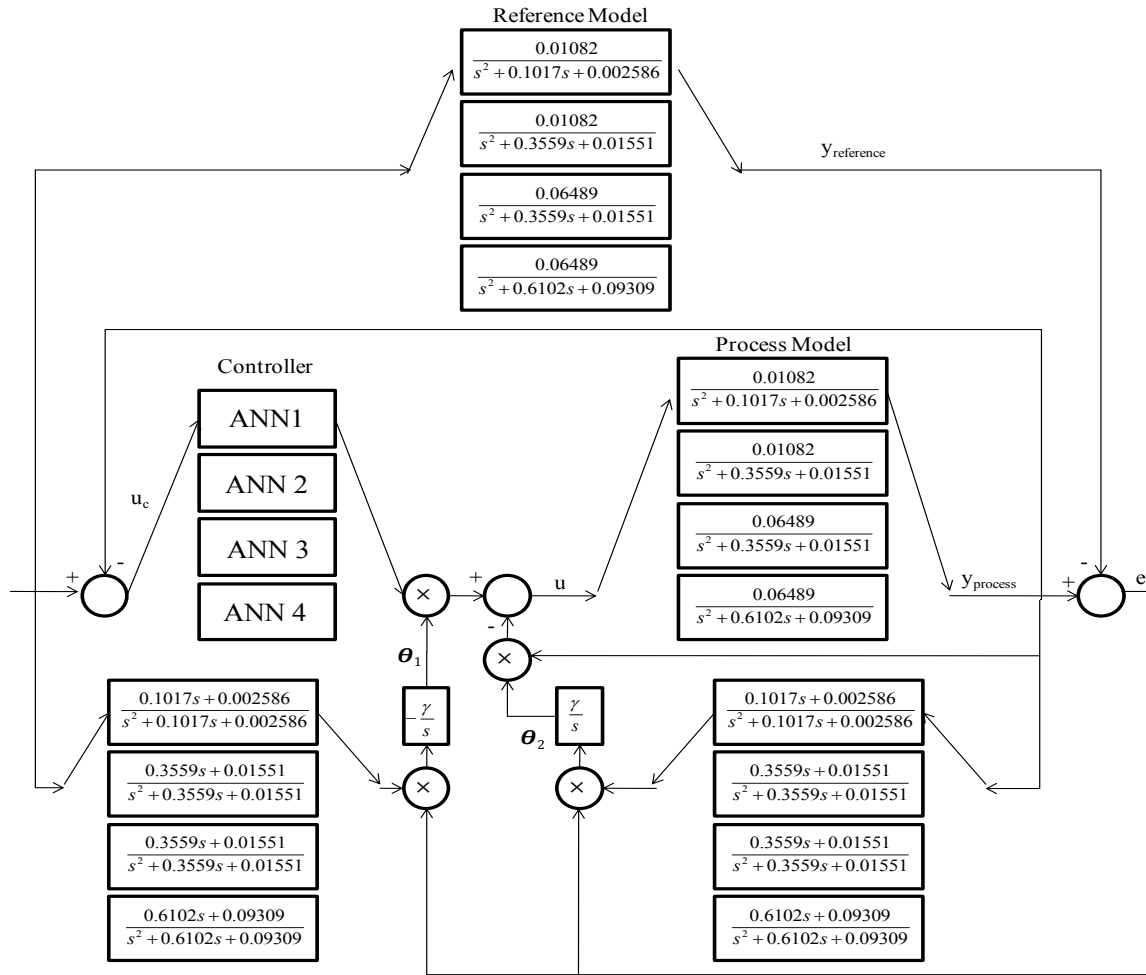
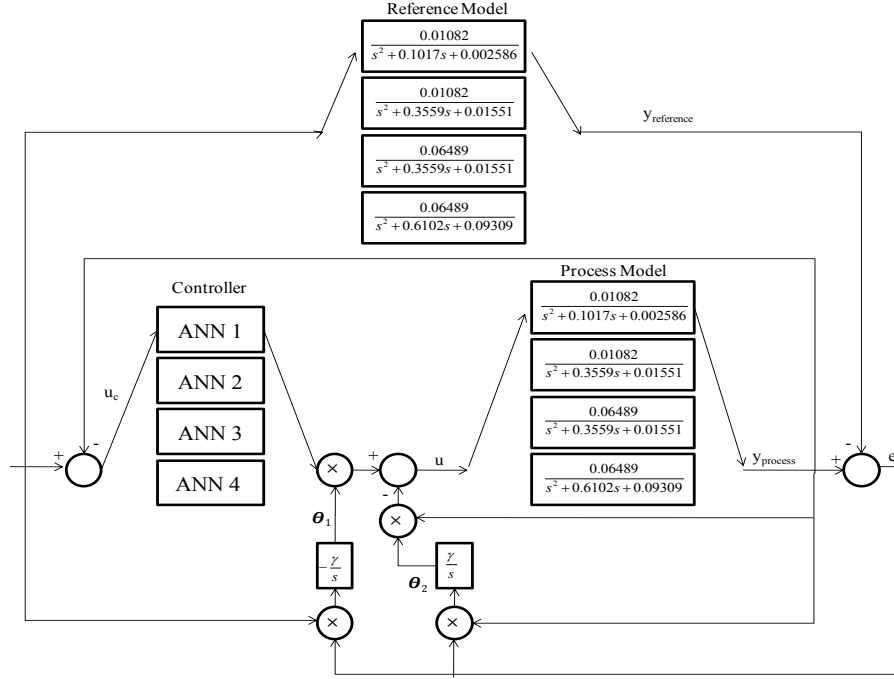


Figure 55. MRAC-Artificial Neural Network 4 Operating Points LPV Controller based on MIT rule.

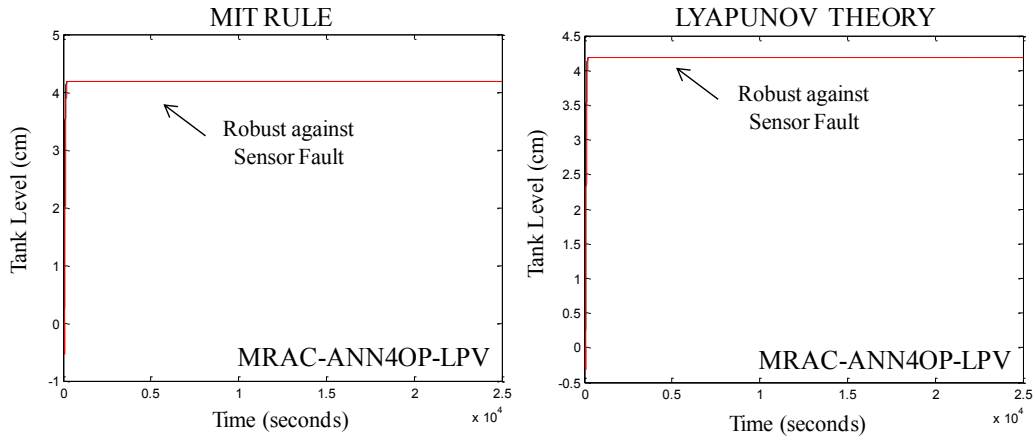
In addition, the implementation of the MRAC-Neural Network 4 Operating Points LPV Controller based on Lyapunov theory is shown in Figure 56.



**Figure 56. MRAC-Artificial Neural Network 4 Operating Points LPV Controller based on Lyapunov theory.**

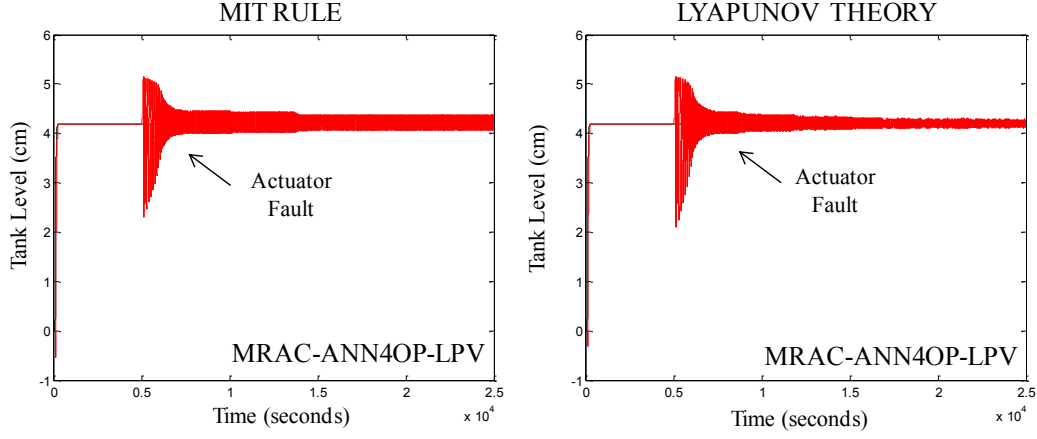
#### 4.2.1 Experiments and Results

This section explains the different experiments realized in the Coupled-Tank System using the MRAC-Artificial Neural Network 4 Operating Points LPV Controller (MRAC-ANN4OP-LPV) based on the MIT rule and based on the Lyapunov theory. Two different types of faults were simulated in the implemented schemes: abrupt and gradual faults. The next figures show the results of the experiments realized using this scheme.



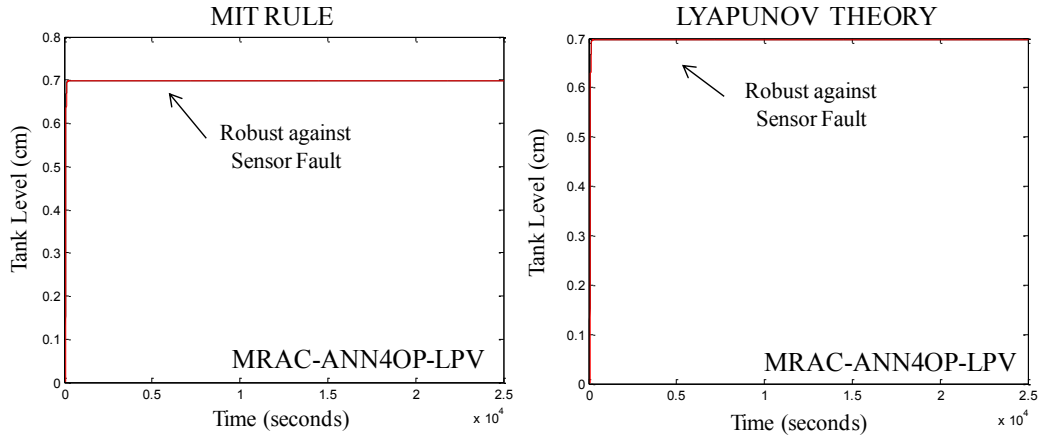
**Figure 57. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-sensor fault of 23.3% at the first operating point.**

In Figure 57, the MRAC-ANN4OP-LPV controllers based on the MIT rule and based on the Lyapunov theory are compared. The controller is working in the operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , an abrupt-sensor fault of 23.3% was introduced at time 5000 seconds. It can be observed, that the MRAC-ANN4OP-LPV based on the MIT rule and based on the Lyapunov theory were robust against the fault.



**Figure 58. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-actuator fault of 1% at the first operating point.**

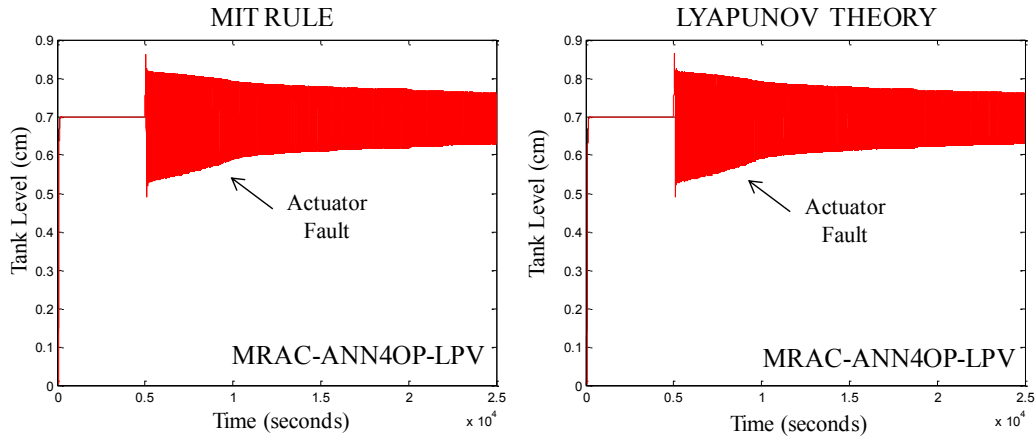
In Figure 58, the MRAC-ANN4OP-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While, the controllers are working in the operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , an abrupt-actuator fault of 1% was introduced at time 5000 seconds. In this figure the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory could not accommodate the fault completely because it can be observed that the system remained with certain oscillations.



**Figure 59. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-sensor fault of 10% at the third operating point.**

In Figure 59, the MRAC-ANN4OP-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While the controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , a gradual-

sensor fault of 10% was introduced at time 5000 seconds. In this figure it can be observed that the MRAC-ANN4OP-LPV based on MIT rule and based on Lyapunov theory were robust against the fault.



**Figure 60. Comparison between the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-actuator fault of 1% at the third operating point.**

In Figure 60, the MRAC-ANN4OP-LPV controllers based on MIT rule and on Lyapunov theory are compared. While the controllers are operating in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , a gradual-actuator fault of 1% was introduced at time 5000 seconds. In this figure, it can be observed that the MRAC-ANN4OP-LPV based on MIT rule and based on Lyapunov theory presented oscillations in the system after the occurrence of the actuator fault.

### 4.3 MRAC- $H_\infty$ 4 Operating Points LPV Controller

A  $H_\infty$  controller was designed for each of the 4 operating points. The  $H_\infty$  control proposed in this work was designed by using the loop shaping method and using the following steps: First, the worst case of system faults were simulated and identified in the form of a Laplace function. Second, these functions are compared against the non-faulty process. Third, a loop shaping control synthesis is performed to calculate an optimal  $H_\infty$  controller for the Laplace fault-functions. This controller shapes the sigma plot of the Laplace fault-function and obtains the desired loop shaping with a precision parameter called *GAM* (e.g. if *GAM* should be  $\geq 1$  with *GAM* = 1 being a perfect match). Figure 61 shows the implementation of the MRAC- $H_\infty$  4 Operating Points LPV Controller based on the MIT rule.

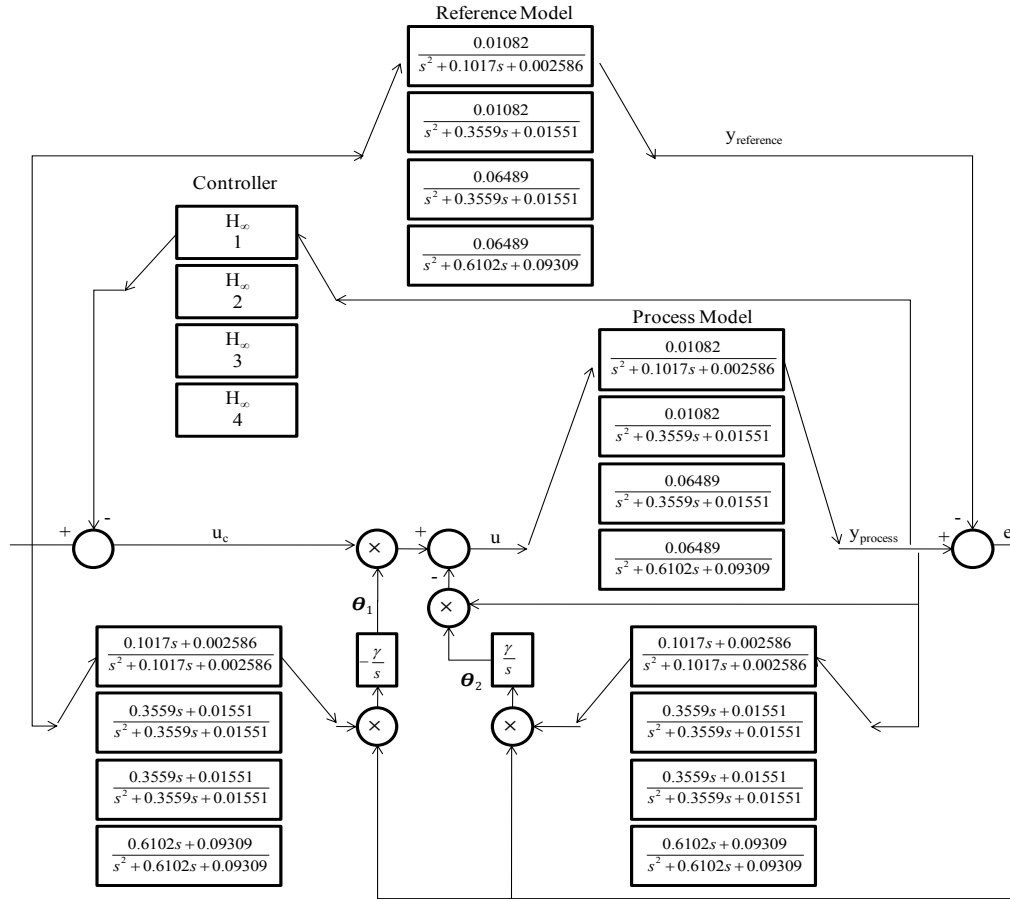


Figure 61. MRAC- $H_\infty$  4 Operating Points LPV Controller based on MIT rule.

The implementation of the MRAC- $H_\infty$  4 Operating Points LPV Controller based on Lyapunov theory is shown in Figure 62.

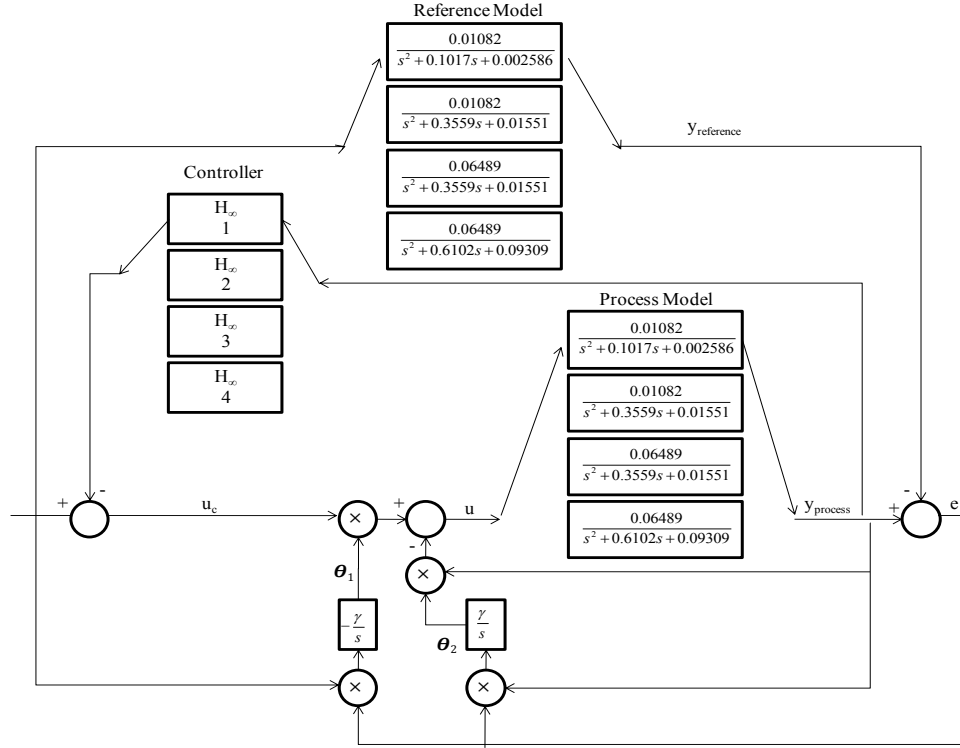


Figure 62. MRAC- $H_\infty$  4 Operating Points LPV Controller based on Lyapunov theory.

#### 4.3.1 Experiments and Results

This section explains the different experiments realized in the Coupled-Tank System using the MRAC- $H_\infty$  4 Operating Points LPV Controller (MRAC-  $H_\infty$ 4OP-LPV) based on the MIT rule and based on the Lyapunov theory. Two different types of faults were simulated in the implemented schemes: abrupt and gradual faults. The next figures show the results of the experiments realized using this scheme.

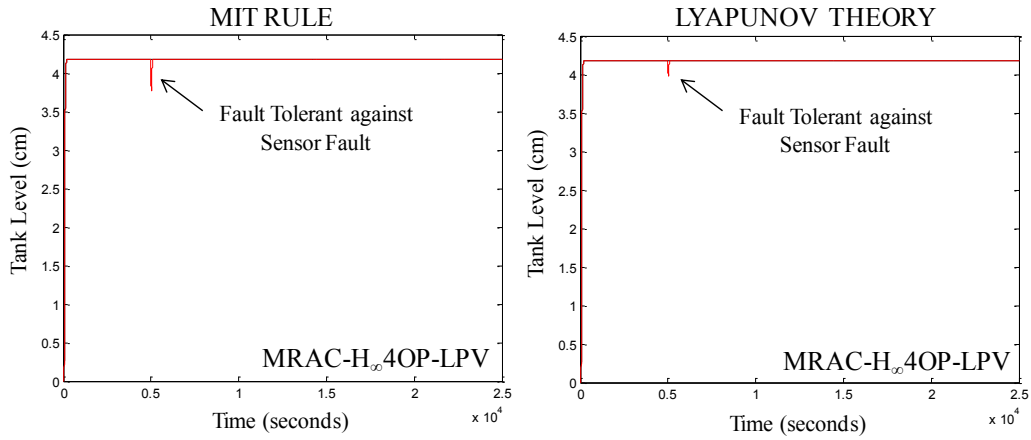
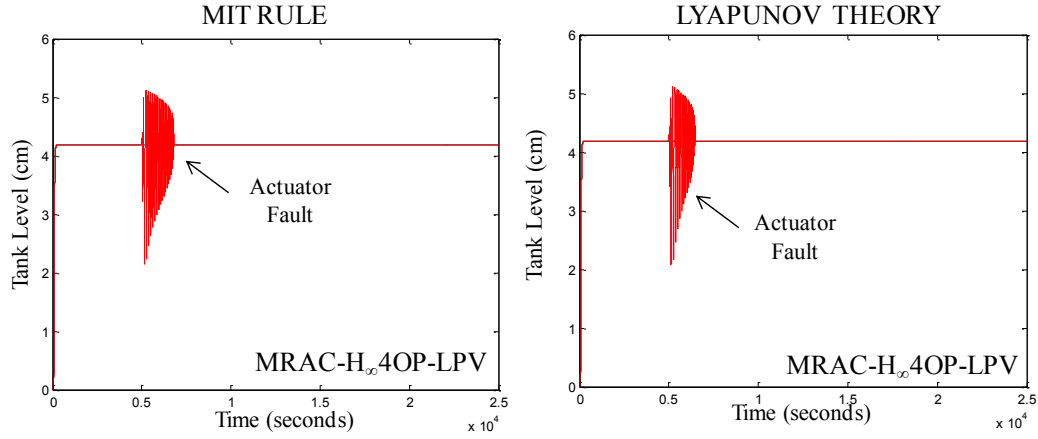


Figure 63. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-sensor fault of 23.3% at the first operating point.

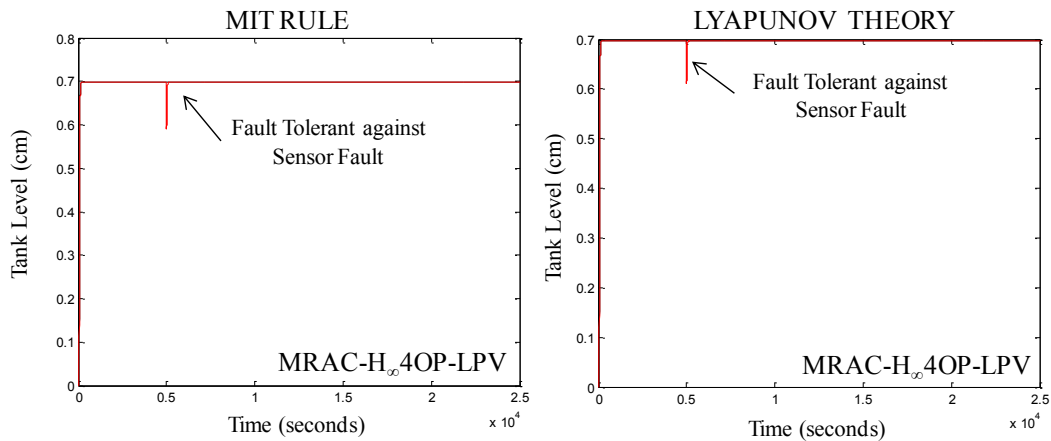


In Figure 63, the MRAC- $H_\infty$ 4OP-LPV controllers based on the MIT rule and based on the Lyapunov theory are compared. The controller is working in the operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , an abrupt-sensor fault of 23.3% was introduced at time 5000 seconds. It can be observed, that the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on the Lyapunov theory were fault tolerant against the fault.



**Figure 64. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with an abrupt-actuator fault of 1% at the first operating point.**

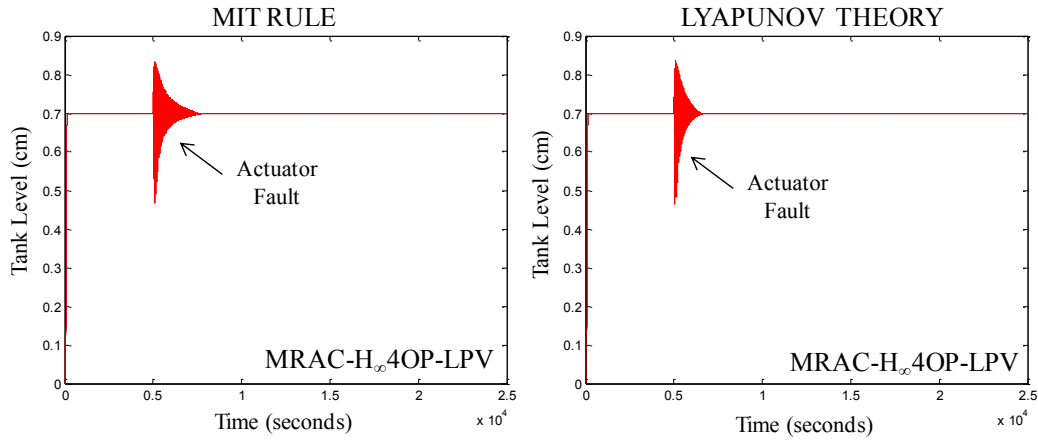
In Figure 64, the MRAC- $H_\infty$ 4OP-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While, the controllers are working in the operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , an abrupt-actuator fault of 1% was introduced at time 5000 seconds. In this figure the MRAC- $H_\infty$ 4OP-LPV based on based on the MIT rule and based on Lyapunov theory were fault tolerant and was able to accommodate the fault in 2000 to 2500 seconds.



**Figure 65. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-sensor fault of 10% at the third operating point.**

In Figure 65, the MRAC- $H_\infty$ 4OP-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While the controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , a gradual-sensor

fault of 10% was introduced at time 5000 seconds. In this figure it can be observed that the MRAC- $H_\infty$ 4OP-LPV based on MIT rule and based on Lyapunov theory were fault tolerant against the fault.



**Figure 66. Comparison between the MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and based on Lyapunov theory with a gradual-actuator fault of 1% at the third operating point.**

In Figure 66, the MRAC- $H_\infty$ 4OP-LPV controllers based on MIT rule and on Lyapunov theory are compared. While the controllers are operating in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , a gradual-actuator fault of 1% was introduced at time 5000 seconds. In this figure, it can be observed that the MRAC- $H_\infty$ 4OP-LPV MIT rule scheme was fault tolerant and was able to accommodate the fault in 4000 seconds and the MRAC- $H_\infty$ 4OP-LPV based on Lyapunov theory scheme was also fault tolerant and it accommodates the fault in 2500 seconds.

#### 4.3.2 Comparison between MRAC-4OP-LPV, MRAC-ANN4OP-LPV and the MRAC- $H_\infty$ 4OP-LPV based on MIT and based on Lyapunov theory

For each of the three proposed schemes based on the MIT rule and on the Lyapunov theory: MRAC 4 Operating Points LPV Controller (MRAC-4OP-LPV), MRAC-Artificial Neural Network 4 Operating Points LPV Controller (MRAC-ANN4OP-LPV) and MRAC- $H_\infty$  4 Operating Points LPV Controller (MRAC- $H_\infty$ 4OP-LPV) the summary of the results are explained in Table 12 and Table 13. These results explain the range in which the methodologies are robust, fault tolerant or degraded against the fault. In addition Figure 67 and Figure 68 show a summary of these controllers.

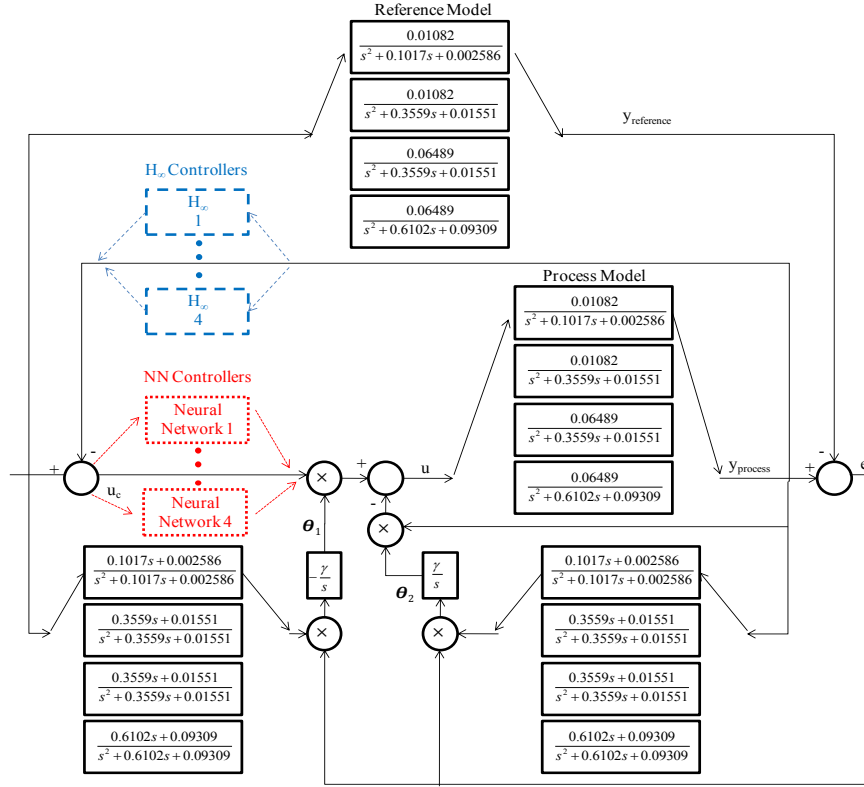


Figure 67. MRAC + ANN +  $H_\infty$  4 Operating Points LPV Controllers based on MIT rule.

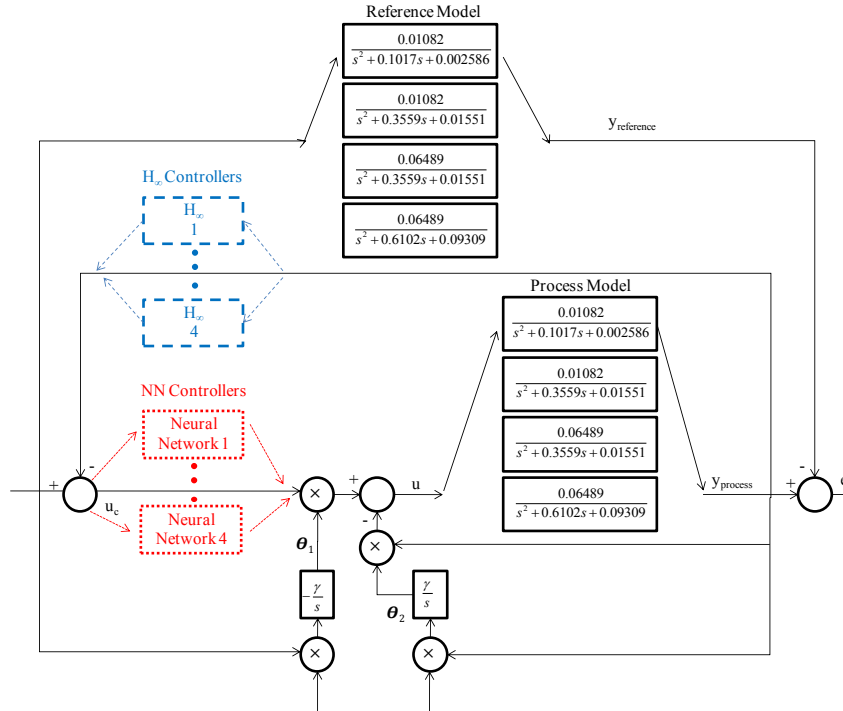


Figure 68. MRAC + ANN +  $H_\infty$  4 Operating Points LPV Controllers based on Lyapunov theory.

**Table 12. Results of experiments of the MRAC-4OP-LPV, MRAC-ANN4OP-LPV and MRAC-H<sub>∞</sub>4OP-LPV methodologies based on MIT rule.**

Approach	Sensor Faults		Actuator Faults	
	Abrupt	Gradual	Abrupt	Gradual
<b>Process Model 1</b>				
<b>MRAC-4OP-LPV</b>	$f < 15\% \rightarrow R$ $15\% < f < 26\% \rightarrow FT$ $f > 26\% \rightarrow D$	$f < +/- 15\% \rightarrow R$ $+/- 15\% < f < +/- 26\% \rightarrow FT$ $f > +/- 26\% \rightarrow D$	$0 < f < 10\% \rightarrow FT$ $f > 10\% \rightarrow D$	$+/- 0 < f < +/- 10\% \rightarrow FT$ $f > +/- 10\% \rightarrow D$
<b>MRAC-ANN4OP-LPV</b>	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
<b>MRAC-H<sub>∞</sub>4OP-LPV</b>	$f < 17\% \rightarrow R$ $17\% < f < 53\% \rightarrow FT$ $f > 53\% \rightarrow D$	$f < +/- 17\% \rightarrow R$ $+/- 17\% < f < +/- 53\% \rightarrow FT$ $f > +/- 53\% \rightarrow D$	$0 < f < 2\% \rightarrow FT$ $f > 2\% \rightarrow D$	$+/- 0 < f < +/- 2\% \rightarrow FT$ $f > +/- 2\% \rightarrow D$
<b>Process Model 2</b>				
<b>MRAC-4OP-LPV</b>	$f < 16\% \rightarrow R$ $16\% < f < 27\% \rightarrow FT$ $f > 27\% \rightarrow D$	$f < +/- 16\% \rightarrow R$ $+/- 16\% < f < +/- 27\% \rightarrow FT$ $f > +/- 27\% \rightarrow D$	$0 < f < 1\% \rightarrow FT$ $f > 1\% \rightarrow D$	$+/- 0 < f < +/- 1\% \rightarrow FT$ $f > +/- 1\% \rightarrow D$
<b>MRAC-ANN4OP-LPV</b>	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
<b>MRAC-H<sub>∞</sub>4OP-LPV</b>	$f < 20\% \rightarrow R$ $20\% < f < 52\% \rightarrow FT$ $f > 52\% \rightarrow D$	$f < +/- 20\% \rightarrow R$ $+/- 20\% < f < +/- 52\% \rightarrow FT$ $f > +/- 52\% \rightarrow D$	$0 < f < 2\% \rightarrow FT$ $f > 2\% \rightarrow D$	$+/- 0 < f < +/- 2\% \rightarrow FT$ $f > +/- 2\% \rightarrow D$
<b>Process Model 3</b>				
<b>MRAC-4OP-LPV</b>	$f < 2\% \rightarrow R$ $2\% < f < 2.5\% \rightarrow FT$ $f > 2.5\% \rightarrow D$	$f < +/- 2\% \rightarrow R$ $+/- 0.52 < f < +/- 2\% \rightarrow FT$ $f > +/- 2.5\% \rightarrow D$	$0 < f < 2\% \rightarrow FT$ $f > 2\% \rightarrow D$	$+/- 0 < f < +/- 2\% \rightarrow FT$ $f > +/- 2\% \rightarrow D$
<b>MRAC-ANN4OP-LPV</b>	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
<b>MRAC-H<sub>∞</sub>4OP-LPV</b>	$f < 10\% \rightarrow R$ $10\% < f < 16\% \rightarrow FT$ $f > 16\% \rightarrow D$	$f < +/- 10\% \rightarrow R$ $+/- 10\% < f < +/- 16\% \rightarrow FT$ $f > +/- 16\% \rightarrow D$	$0 < f < 2\% \rightarrow FT$ $f > 2\% \rightarrow D$	$+/- 0 < f < +/- 2\% \rightarrow FT$ $f > +/- 2\% \rightarrow D$
<b>Process Model 4</b>				
<b>MRAC-4OP-LPV</b>	$f < 2\% \rightarrow R$ $2\% < f < 2.5\% \rightarrow FT$ $f > 2.5\% \rightarrow D$	$f < +/- 2\% \rightarrow R$ $+/- 2\% < f < +/- 2.5\% \rightarrow FT$ $f > +/- 2.5\% \rightarrow D$	$0 < f < 2\% \rightarrow FT$ $f > 2\% \rightarrow D$	$+/- 0 < f < +/- 2\% \rightarrow FT$ $f > +/- 2\% \rightarrow D$
<b>MRAC-ANN4OP-LPV</b>	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
<b>MRAC-H<sub>∞</sub>4OP-LPV</b>	$f < 7\% \rightarrow R$ $7\% < f < 16\% \rightarrow FT$ $f > 16\% \rightarrow D$	$f < +/- 7\% \rightarrow R$ $+/- 7\% < f < +/- 16\% \rightarrow FT$ $f > +/- 16\% \rightarrow D$	$0 < f < 2\% \rightarrow FT$ $f > 2\% \rightarrow D$	$+/- 0 < f < +/- 2\% \rightarrow FT$ $f > +/- 2\% \rightarrow D$

**R=Robust, FT = Fault Tolerant, D = Degraded**

**Table 13. Results of experiments of the MRAC-4OP-LPV, MRAC-ANN4OP-LPV and MRAC- $H_\infty$ 4OP-LPV methodologies based on Lyapunov theory.**

Approach	Sensor Faults		Actuator Faults	
	Abrupt	Gradual	Abrupt	Gradual
<b>Process Model 1</b>				
MRAC-4OP-LPV	$f < 18\% \rightarrow R$	$f < +/- 18\% \rightarrow R$	$0 < f < 2.5\% \rightarrow FT$	$+/- 0 < f < +/- 2.5\% \rightarrow$
	$18\% < f < 26\% \rightarrow$	$+/- 18\% < f < +/- 26\% \rightarrow$	$f > 2.5\% \rightarrow D$	$FT$
	$FT$	$FT$		$f > +/- 2.5\% \rightarrow D$
	$f > 26\% \rightarrow D$	$f > +/- 26\% \rightarrow D$		
MRAC-ANN4OP-LPV	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
MRAC- $H_\infty$ 4OP – LPV	$f < 20\% \rightarrow R$	$f < +/- 20\% \rightarrow R$	$0 < f < 2\% \rightarrow FT$	$+/- 0 < f < +/- 2\% \rightarrow$
	$20\% < f < 53\% \rightarrow$	$+/- 20\% < f < +/- 53\% \rightarrow$	$f > 2\% \rightarrow D$	$FT$
	$FT$	$FT$		$f > +/- 2\% \rightarrow D$
	$f > 53\% \rightarrow D$	$f > +/- 53\% \rightarrow D$		
<b>Process Model 2</b>				
MRAC-4OP-LPV	$f < 18\% \rightarrow R$	$f < +/- 18\% \rightarrow R$	$0 < f < 3\% \rightarrow FT$	$+/- 0 < f < +/- 3\% \rightarrow$
	$18\% < f < 27\% \rightarrow$	$+/- 18\% < f < +/- 27\% \rightarrow FT$	$f > 3\% \rightarrow D$	$FT$
	$FT$	$f > +/- 27\% \rightarrow D$		$f > +/- 3\% \rightarrow D$
	$f > 27\% \rightarrow D$			
MRAC-ANN4OP-LPV	R	R	$f > 0 \rightarrow$ Degraded System	$f > 0 \rightarrow$ Degraded System
MRAC- $H_\infty$ 4OP – LPV	$f < 22\% \rightarrow R$	$f < +/- 22\% \rightarrow R$	$0 < f < 2\% \rightarrow FT$	$+/- 0 < f < +/- 2\% \rightarrow$
	$22\% < f < 50\% \rightarrow$	$+/- 22\% < f < +/- 50\% \rightarrow FT$	$f > 2\% \rightarrow D$	$FT$
	$FT$	$f > +/- 50\% \rightarrow D$		$f > +/- 2\% \rightarrow D$
	$f > 50\% \rightarrow D$			
<b>Process Model 3</b>				
MRAC-4OP-LPV	$f < 1.8\% \rightarrow R$	$f < +/- 1.8\% \rightarrow R$	$0 < f < 6\% \rightarrow FT$	$+/- 0 < f < +/- 6\% \rightarrow$
	$1.8\% < f < 2.4\% \rightarrow$	$+/- 1.8\% < f < +/- 2.4\% \rightarrow FT$	$f > 6\% \rightarrow D$	$FT$
	$FT$	$f > +/- 2.4\% \rightarrow D$		$f > +/- 6\% \rightarrow D$
	$f > 2.4\% \rightarrow D$			
MRAC-ANN4OP-LPV	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
MRAC- $H_\infty$ 4OP – LPV	$f < 8.5\% \rightarrow R$	$f < +/- 8.5\% \rightarrow R$	$0 < f < 0.5\% \rightarrow FT$	$+/- 0 < f < +/- 2\% \rightarrow$
	$8.5\% < f < 11\% \rightarrow$	$+/- 8.5\% < f < +/- 11\% \rightarrow FT$	$f > 0.5\% \rightarrow D$	$FT$
	$FT$	$f > +/- 11\% \rightarrow D$		$f > +/- 2\% \rightarrow D$
	$f > 11\% \rightarrow D$			
<b>Process Model 4</b>				
MRAC-4OP-LPV	$f < 1.8\% \rightarrow R$	$f < +/- 1.8\% \rightarrow R$	$0 < f < 6\% \rightarrow FT$	$+/- 0 < f < +/- 6\% \rightarrow$
	$1.8\% < f < 2.4\% \rightarrow$	$+/- 1.8\% < f < +/- 2.4\% \rightarrow FT$	$f > 6\% \rightarrow D$	$FT$
	$FT$	$f > +/- 2.4\% \rightarrow D$		$f > +/- 6\% \rightarrow D$
	$f > 2.4\% \rightarrow D$			
MRAC-ANN4OP-LPV	R	R	$f > 0 \rightarrow D$	$f > 0 \rightarrow D$
MRAC- $H_\infty$ 4OP – LPV	$f < 7.5\% \rightarrow R$	$f < +/- 7.5\% \rightarrow R$	$0 < f < 2\% \rightarrow FT$	$+/- 0 < f < +/- 2\% \rightarrow$
	$7.5\% < f < 13\% \rightarrow$	$+/- 7.5\% < f < +/- 13\% \rightarrow FT$	$f > 2\% \rightarrow D$	$FT$
	$FT$	$f > +/- 13\% \rightarrow D$		$f > +/- 2\% \rightarrow D$
	$f > 13\% \rightarrow D$			

**R=Robust, FT = Fault Tolerant, D = Degraded**

In addition, the Mean Square Error (MSE) was calculated for all the experiments. The results are shown in Table 14. It is important to mention that the results were the same for abrupt and gradual faults.

**Table 14. MSE Results of the comparison between the MRAC-4OP-LPV, MRAC-ANN4OP-LPV and MRAC- $H_\infty$ 4OP-LPV based on MIT rule and based on Lyapunov theory.**

Process Model 1				
Methodology	MRAC MIT based design		MRAC Lyapunov based design	
	Sensor Faults (23.3%)	Actuator Faults (1%)	Sensor Faults (23.3%)	Actuator Faults (1%)
MRAC-4OP-LPV	0.00272417	0.10804153	0.000780457	0.084395641
MRAC-ANN4OP-LPV	0.009755478	0.047977847	0.006817029	0.043201383
MRAC- $H_\infty$ 4OP-LPV	0.00026123	0.035337832	5.98686E-05	0.029824218
Process Model 3				
Methodology	MRAC MIT based design		MRAC Lyapunov based design	
	Sensor Faults (23.3%)	Actuator Faults (1%)	Sensor Faults (magnitude 3)	Actuator Faults (1%)
MRAC-4OP-LPV	1.554861422	8.14169E-05	1.554861325	6.68611E-05
MRAC-ANN4OP-LPV	6.50112E-05	0.00314155	7.72308E-05	0.003103844
MRAC- $H_\infty$ 4OP-LPV	7.90349E-06	0.000183443	5.34546E-06	0.000163684

In all the results of Table 14, it can be seen that the MSE from the schemes using the MRAC based on the Lyapunov theory is lower than the schemes using the MRAC based on MIT, because the Lyapunov theory adds stability to the closed-loop system.

#### 4.4 MRAC-LPV Controller

Now, taking the above results as a pre-test, an LPV Model Reference Adaptive Controller for the LPV system was designed (MRAC-LPV). First, the state-space LPV model was transformed to a continuous space model:

$$G(s)_{LPV\_model} = C(sI - A)^{-1}B + D \quad (132)$$

$$G(s)_{LPV\_model} = [0 \quad 1] \left( \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} -0.5085 \varphi_1 & 0 \\ 0.5085 \varphi_1 & -0.5085 \varphi_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0.2127 \\ 0 \end{bmatrix} + [0] \quad (133)$$

$$G(s)_{LPV\_model} = \frac{0.108158 \varphi_1}{(s+0.5085 \varphi_2)(s+0.5085 \varphi_1)} \quad (134)$$

$$G(s)_{LPV\_model} = \frac{0.108158 \varphi_1}{s^2 + (0.5085 \varphi_1 s + 0.5085 \varphi_2 s) + 0.258572 \varphi_1 \varphi_2} \quad (135)$$

$$G(s)_{LPV\_model} = \frac{0.108158 \varphi_1}{s^2 + 0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2} \quad (136)$$

Then, the reference model is:

$$G(s)_{Reference\_model} = \frac{0.108158 \varphi_1}{s^2 + 0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2} \quad (137)$$

As in the previous cases, this reference model is the same as the process model when the system has no faults.

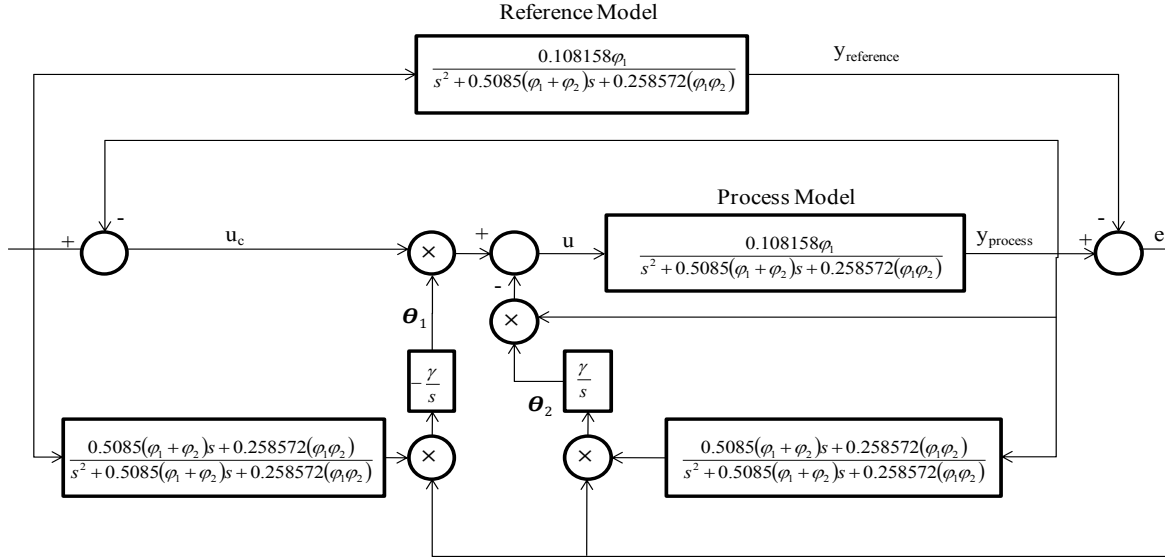
$$G(s)_{Process\_model} = \frac{0.108158 \varphi_1}{s^2 + 0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2} \quad (138)$$

The adaptive feed forward update rule ( $\theta_1$ ) and the adaptive feedback update rule ( $\theta_2$ ) for the MRAC-LPV based on MIT rule are:

$$\frac{d\theta_1}{dt} = -\gamma \frac{\partial e}{\partial \theta_1} e = -\gamma \left( \frac{0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2}{s^2 + 0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2} u_c \right) e \quad (139)$$

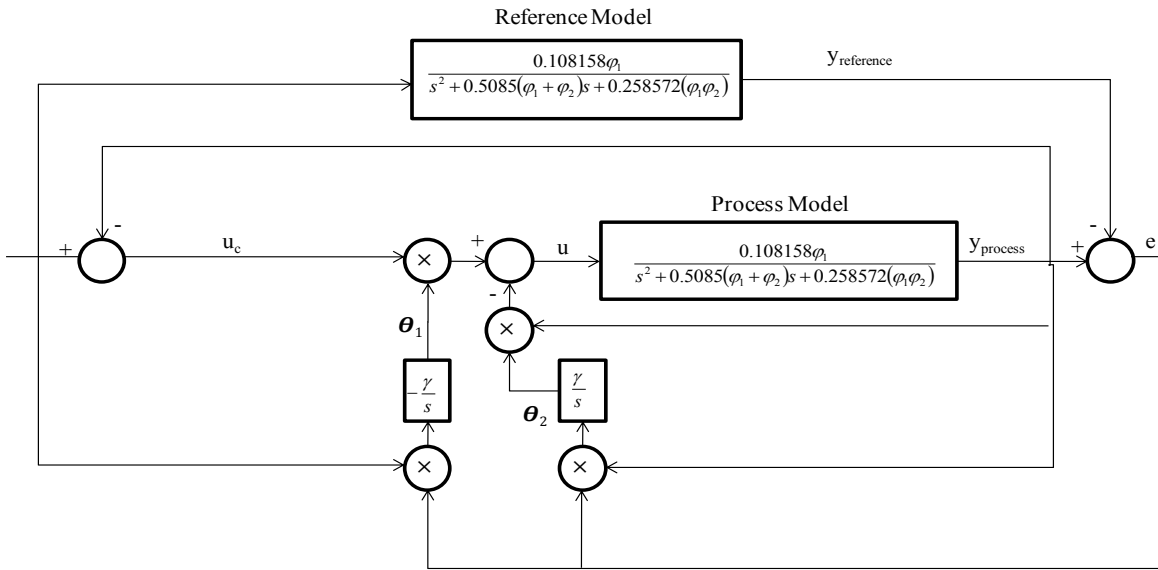
$$\frac{d\theta_2}{dt} = -\gamma \frac{\partial e}{\partial \theta_2} e = \gamma \left( \frac{0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2}{s^2 + 0.5085(\varphi_1 + \varphi_2)s + 0.258572 \varphi_1 \varphi_2} y_p \right) e \quad (140)$$

Finally, the MRAC-LPV system based on the MIT rule is represented in Figure 69:



**Figure 69. MRAC-LPV system based on MIT rule.**

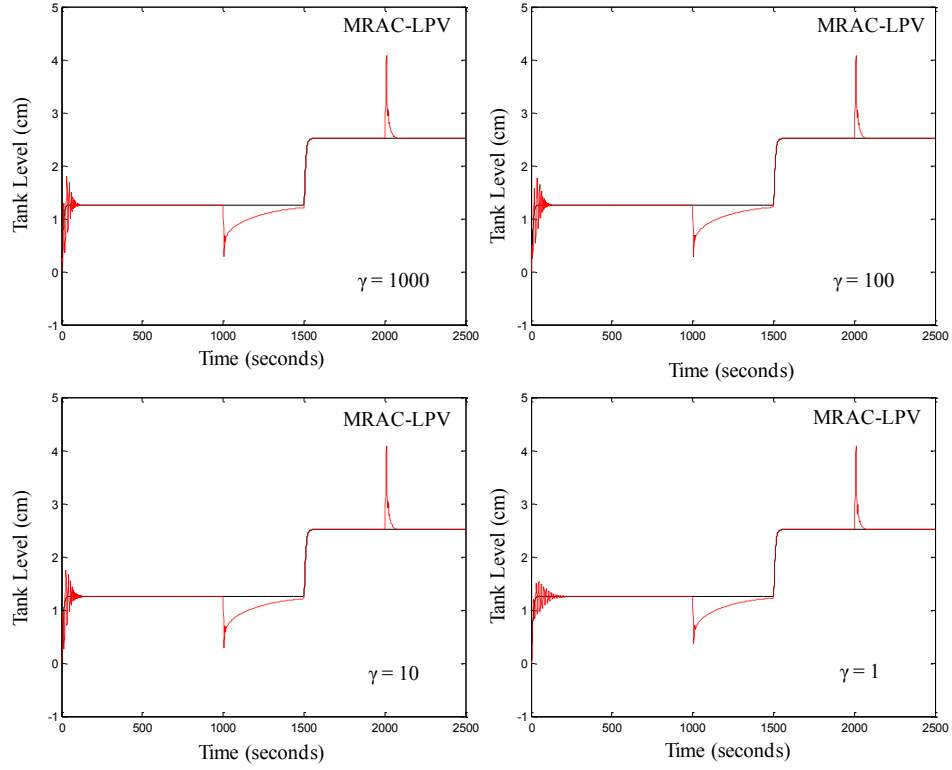
The MRAC-LPV based on Lyapunov theory is shown in Figure 70.



**Figure 70. MRAC-LPV system based on Lyapunov theory.**

To select the value of  $\gamma$ , different experiments with different  $\gamma$  sizes were realized. In these experiments a sensor fault of 3.3% was introduced at time 1000 seconds and an actuator fault of 3% was introduced at time 2000 seconds. In addition, a change in the operating point was applied at time 1500 seconds. In summary, four different size of  $\gamma$  were tested (1000, 100, 10, 1) and the results are showed in Figure 71 and Table 15.





**Figure 71. MRAC-LPV Results testing different sizes of  $\gamma$ .**

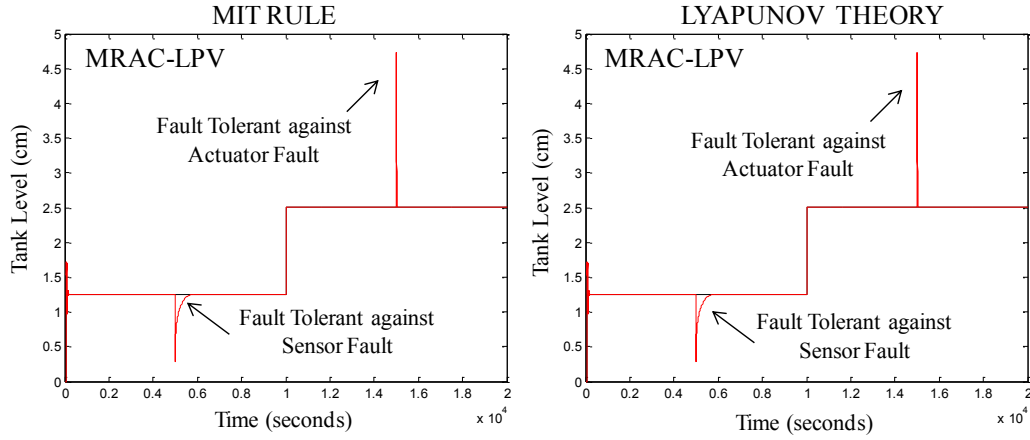
**Table 15. MSE of the MRAC-LPV results of different sizes of  $\gamma$ .**

$\gamma$	Sensor Fault	Actuator Fault	Total MSE
1000	0.0748	0.0398	0.0128
100	0.0714	0.0388	0.0122
10	0.0726	0.0393	0.0124
1	0.0748	0.0398	0.0128

From Figure 71 and Table 15 the selected value of  $\gamma$  to realize the following experiments was 100, because it has the lower MSE in sensor and actuator faults.

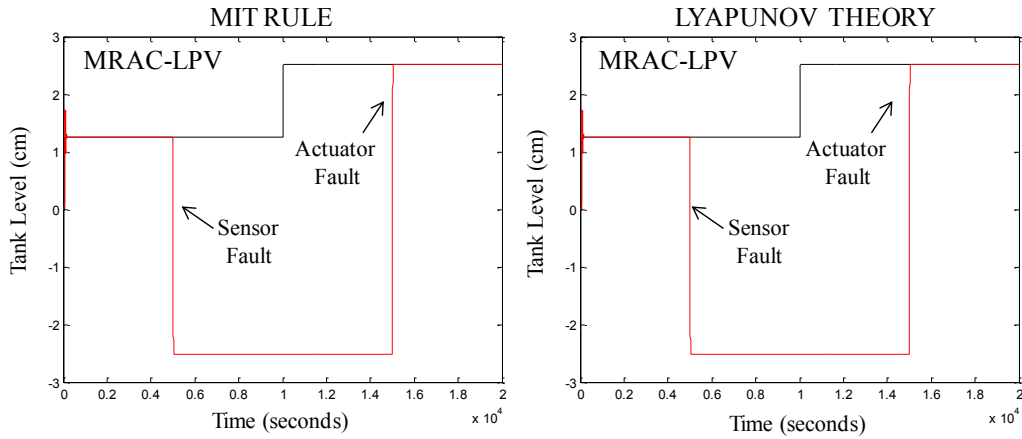
#### 4.4.1 Experiments and Results

This section explains the different experiments realized in the Coupled-Tank System using the MRAC-LPV Controller (MRAC-LPV) based on the MIT rule and based on the Lyapunov theory. Two different types of faults were simulated in the implemented schemes: abrupt and gradual faults. The next figures show the results of the experiments realized using this scheme.



**Figure 72. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of magnitude 3.3% and an abrupt-actuator fault of magnitude 20% for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$ .**

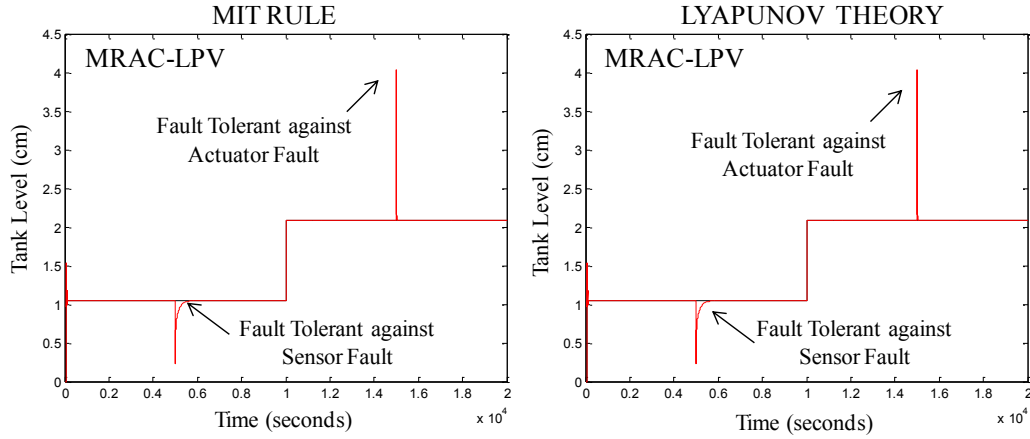
In Figure 72, the MRAC-LPV Controllers based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , an abrupt-sensor fault of 3.3% was introduced at time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. It can be observed that the MRAC-LPV scheme was fault tolerant for the sensor and the actuator fault and could tolerate the change in the operation points for both methodologies.



**Figure 73. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of magnitude 160% and an abrupt-actuator fault of magnitude 20% for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$ .**

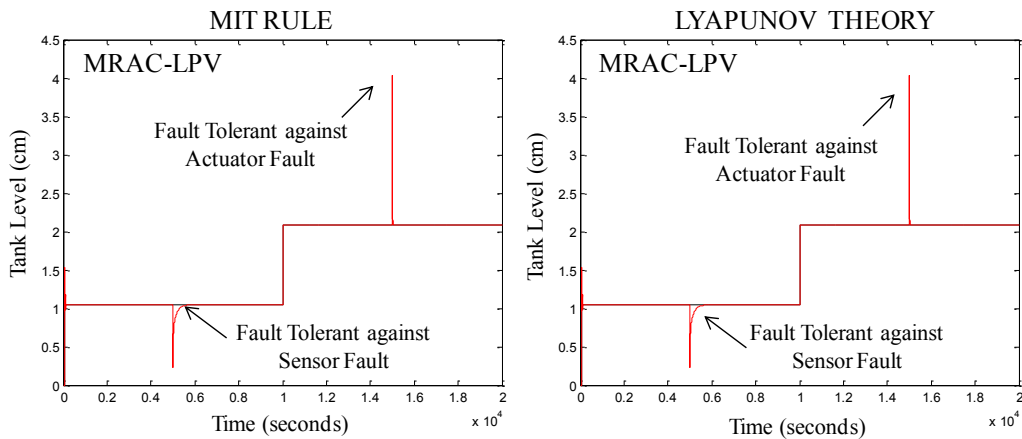
In Figure 73, the MRAC-LPV Controllers based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , an abrupt-sensor fault of 160% was introduced in time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000

seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, the MRAC-LPV scheme became degraded after the occurrence of the sensor fault, and was able to accommodate the fault after 10000 seconds for both methodologies (MIT rule and Lyapunov theory).



**Figure 74. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 3.3% and a gradual-actuator fault of 20% for the operating points  $\varphi_1=0.6$  and  $\varphi_2=0.6$ .**

In Figure 74, the MRAC-LPV Controllers based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.6$ , a gradual-sensor fault of 3.3% was introduced in time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC-LPV scheme was fault tolerant for the sensor and the actuator fault and could tolerate the change in the operation points for both methodologies.



**Figure 75. Comparison between the MRAC-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 160% and a gradual-actuator fault of 20% for the operating points  $\varphi_1=0.6$  and  $\varphi_2=0.6$ .**

In Figure 75, the MRAC-LPV Controller based on the MIT rule and on the Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.6$ , a gradual-sensor fault of 160% was introduced in time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure it can be observed that the MRAC-LPV was fault tolerant against the sensor and actuator fault and could tolerate the change in the operating point for both methodologies (MIT rule and Lyapunov theory).

#### 4.5 $H_\infty$ Gain Scheduling -MRAC LPV Controller

In addition a  $H_\infty$  Gain Scheduling Control was developed. The  $H_\infty$  Gain Scheduling Control is applicable to affine parameter-dependent plants, where the dependent parameter ( $p(t)$ ) is a time-varying vector of physical parameters (velocity, set point, etc.). And the matrices of the system A, B, C and D are affine functions of  $p(t)$ . The dependent parameters need to be measured in real time. For this reason a controller that incorporates such measurements of the parameters to adjust the operating condition to these changes must be designed (Packard, 1994). This type of controller is named scheduling control.

Therefore, if the parameter vector  $p(t)$  takes values in a box with corners  $\{\Pi_i\}_{i=1}^N$  ( $N = 2^n$ ), the plant system matrix:

$$S(p) := \begin{pmatrix} A(p) & B(p) \\ C(p) & D(p) \end{pmatrix} \quad (141)$$

ranges in a matrix polytope with vertices  $S(\Pi_i)$ . Particularly, given any convex decomposition:

$$p(t) = \alpha_1 \Pi_1 + \dots + \alpha_N \Pi_N, \quad \alpha_i \geq 0, \quad \sum_{i=1}^N \alpha_i = 1 \quad (142)$$

of  $p$  over the corners of the parameter box, the system matrix  $S(p)$  is presented by

$$S(p) = \alpha_1 S(\Pi_1) + \dots + \alpha_N S(\Pi_N) \quad (143)$$

The above implies to seek for parameter dependent controllers with equations

$$K(., p) \begin{cases} \zeta = A_K(p)\zeta + B_K(p)y \\ u = C_K(p)\zeta + D(p)y \end{cases} \quad (144)$$

whose vertex property is: Given the convex decomposition  $p(t) = \sum_{i=1}^N \alpha_i \Pi_i$  of a current parameter value  $p(t)$ , the values of  $A_K(p)$ ,  $B_K(p)$ , ... are derived from the values of  $A_K(\Pi_i)$ ,  $B_K(\Pi_i)$ , ... at the corners of the parameter box by

$$\begin{pmatrix} A_K(p) & B_K(p) \\ C_K(p) & D_K(p) \end{pmatrix} = \sum_{i=1}^N \alpha_i \begin{pmatrix} A_K(\Pi_i) & B_K(\Pi_i) \\ C_K(\Pi_i) & D_K(\Pi_i) \end{pmatrix} \quad (145)$$

The above means that the controller state-space matrices in a specific operating point  $p(t)$  are obtained by convex interpolation of the LTI vertex controllers

$$K_i := \begin{pmatrix} A_K(\Pi_i) & B_K(\Pi_i) \\ C_K(\Pi_i) & D_K(\Pi_i) \end{pmatrix} \quad (146)$$

That generates a smooth scheduling of the controller matrices by the parameter measurements  $p(t)$ .

The designed gain scheduled controller  $K(., p)$  must satisfy the vertex property and the closed-loop system should be stable for all admissible parameters trajectories  $p(t)$  (Apkarian & Gahinet, 1995; Apkarian et al., 1995; Becker & Packard, 1994; Packard, 1994).

In order to design the  $H_\infty$  Gain Scheduling LPV Controller, two weighting functions were established ( $W_I$  and  $W_2$ ). To obtain  $W_I$ , the following procedure is used: First, four plants were established using the extreme operation points. Also, a nominal plant was obtained using the average of the operation points (see Table 16).

**Table 16. Different Plants at the four and nominal operating points.**

Plant Number	Operating Points	Plants Transfer Functions
Plant 1	$\varphi_1=0.1$	$\frac{0.01082}{s^2 + 0.1017s + 0.002586}$
	$\varphi_2=0.1$	
Plant 2	$\varphi_1=0.1$	$\frac{0.01082}{s^2 + 0.3559s + 0.01551}$
	$\varphi_2=0.6$	
Plant 3	$\varphi_1=0.6$	$\frac{0.06489}{s^2 + 0.3559s + 0.01551}$
	$\varphi_2=0.1$	
Plant 4	$\varphi_1=0.6$	$\frac{0.06489}{s^2 + 0.6102s + 0.09309}$
	$\varphi_2=0.6$	
Nominal Plant	$\varphi_1=0.35$	$\frac{0.06489}{s^2 + 0.6102s + 0.09309}$
	$\varphi_2=0.35$	

Then, the multiplicative uncertainty for each plant is calculated with the following equation:

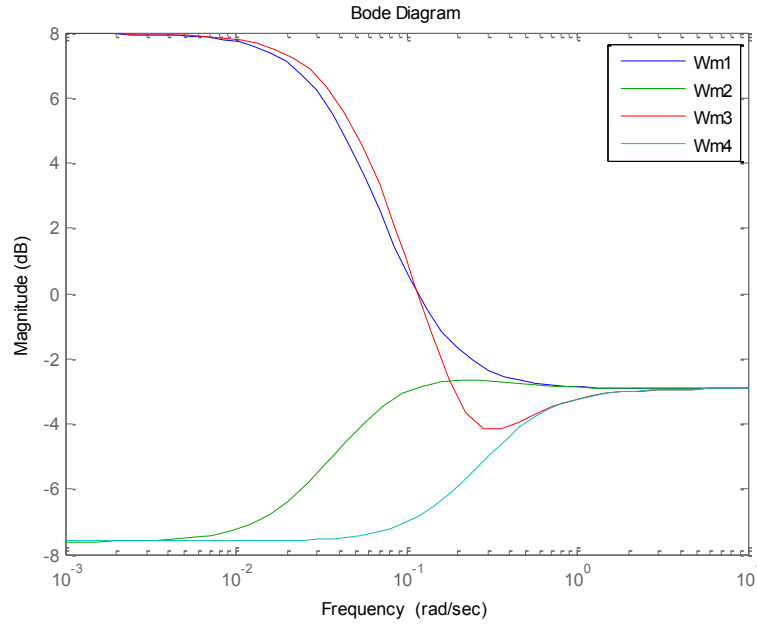
$$W_{mi} = \frac{(Plant\ i - Nominal\ Plant)}{Nominal\ Plant} \quad (147)$$

Table 17 shows the resulting Multiplicative uncertainty of each Plant.

**Table 17. Multiplicative Uncertainty of each Plant.**

Plant Number	Operating Points	Multiplicative Uncertainty
Plant 1	$\varphi_1=0.1$	$W_{m1} = \frac{-0.02703s^4 - 0.009618s^3 - 0.0006107s^2 + 8.717e^{-5}s + 7.752e^{-6}}{0.03785s^4 + 0.01732s^3 + 0.002667s^2 + 0.0001567s + 3.1e^{-6}}$
	$\varphi_2=0.1$	
Plant 2	$\varphi_1=0.1$	$W_{m2} = \frac{-0.02703s^4 - 0.01924s^3 - 0.004524s^2 - 0.0003916s - 7.74e^{-6}}{0.03785s^4 + 0.02694s^3 + 0.00658s^2 + 0.0006356s + 1.859e^{-5}}$
	$\varphi_2=0.6$	
Plant 3	$\varphi_1=0.6$	$W_{m3} = \frac{0.02704s^4 + 0.01925s^3 + 0.005749s^2 + 0.0008272s + 4.649e^{-5}}{0.03785s^4 + 0.02694s^3 + 0.00658s^2 + 0.0006356s + 1.859e^{-5}}$
	$\varphi_2=0.1$	
Plant 4	$\varphi_1=0.6$	$W_{m4} = \frac{0.02704s^4 + 0.009622s^3 - 0.0006126s^2 - 0.0005227s - 4.65e^{-5}}{0.03785s^4 + 0.03657s^3 + 0.01294s^2 + 0.001985s + 0.0001116}$
	$\varphi_2=0.6$	

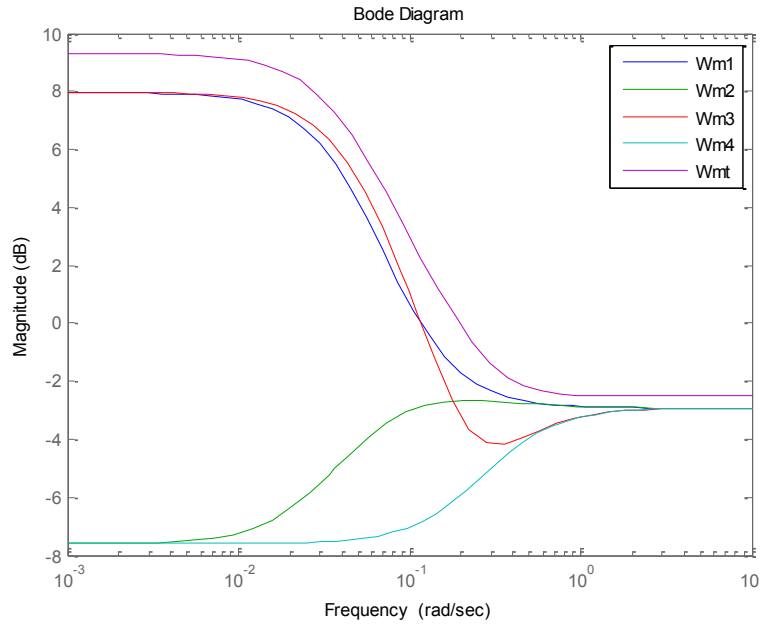
The next step is to plot a Bode diagram of the above uncertainties (see Figure 76).



**Figure 76. Bode Diagram of the 4 Plants Multiplicative Uncertainties.**

With the above Bode diagram, a multiplicative uncertainty function that includes  $W_{m1}$ ,  $W_{m2}$ ,  $W_{m3}$  and  $W_{m4}$  is obtained:

$$W_1 = W_{mt} = \frac{0.75s^4 + 0.33s^3 + 0.02s^2 - 0.00882s}{s^4 + 0.568837s^3 + 0.091878s^2 + 0.003019s} \quad (148)$$



**Figure 77. Bode Diagram of all Multiplicative Uncertainties.**

To obtain  $W_2$ , the next procedure is applied. The additive uncertainty calculus for each plant is calculated with:

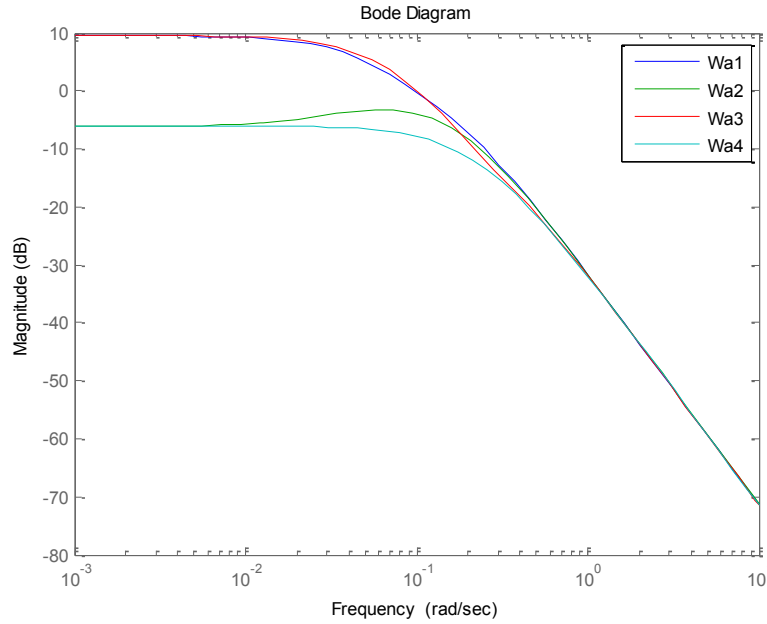
$$W_{ai} = \text{Plant } i - \text{Nominal Plant} \quad (149)$$

Table 18 shows the resulting Additive uncertainty of each Plant.

**Table 18. Additive Uncertainty of each Plant.**

Plant Number	Operating Points	Additive Uncertainty
Plant 1	$\varphi_1=0.1$	$W_{a1} = \frac{-0.02703s^2 + 1.493e^{-6}s + 0.0002448}{s^4 + 0.4576s^3 + 0.07045s^2 + 0.004141s + 8.19e^{-5}}$
	$\varphi_2=0.1$	
Plant 2	$\varphi_1=0.1$	$W_{a2} = \frac{-0.02703s^2 - 0.00962s - 0.0002444}{s^4 + 0.7118s^3 + 0.1738s^2 + 0.01679s + 0.0004912}$
	$\varphi_2=0.6$	
Plant 3	$\varphi_1=0.6$	$W_{a3} = \frac{0.02704s^2 + 0.009624s + 0.001468}{s^4 + 0.7118s^3 + 0.1738s^2 + 0.01679s + 0.0004912}$
	$\varphi_2=0.1$	
Plant 4	$\varphi_1=0.6$	$W_{a4} = \frac{0.02704s^2 - 1.719e^{-6}s - 0.001468}{s^4 + 0.9661s^3 + 0.3419s^2 + 0.05246s + 0.002948}$
	$\varphi_2=0.6$	

The next step is to plot a Bode diagram of the above uncertainties (see Figure 78).

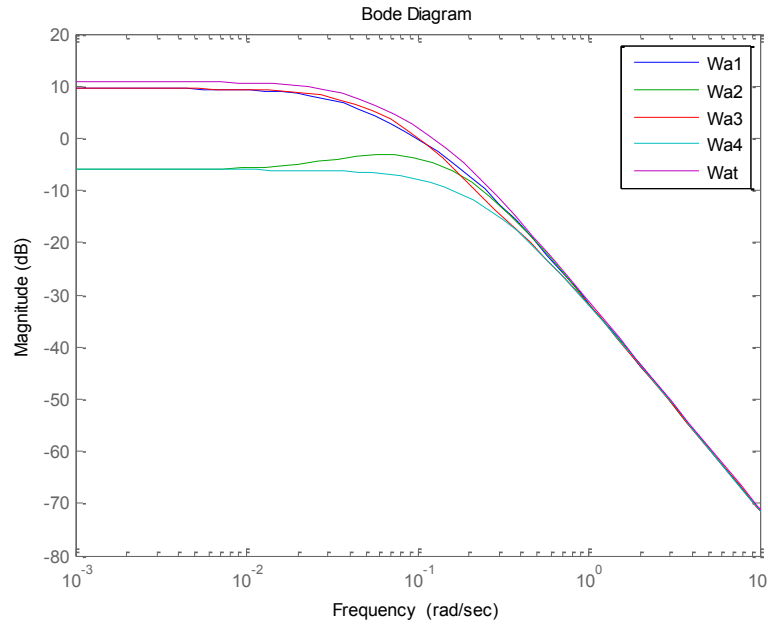


**Figure 78. Bode Diagram of the 4 Plants Additive Uncertainties.**



Finally, with the above bode diagram an additive uncertainty function that includes  $W_{a1}$ ,  $W_{a2}$ ,  $W_{a3}$  and  $W_{a4}$  is obtained (see Figure 79):

$$W_2 = W_{at} = \frac{0.02839 s^4 + 0.01249 s^3 + 0.000757 s^2 - 0.0003338 s}{s^6 + 0.9247 s^5 + 0.326 s^4 + 0.05373 s^3 + 0.003984 s^2 + 9.561e^{-5} s} \quad (150)$$



**Figure 79. Bode Diagram of all Additive Uncertainties.**

After calculating  $W_1$  and  $W_2$  the following procedure implemented in Matlab<sup>®</sup> was realized:

- First, the value of the learning rate and the specific desired operation points were established:

```
%LPV-MRAC-Hinfgs Controller
%Input Data
gamma=100;
phi1=.3;
phi2=.5;
%.1<=phi1<=.6
%.1<=phi2<=.6
```

- Second,  $W_1$  and  $W_2$  have to be transformed into a Linear Time Invariant (LTI) system:

```
%Filter Shape W1 y W2
n1=[0.02839 0.01249 0.000757 -0.0003338];
d1=[1 0.9247 0.326 0.05373 0.003984 9.561e-005];
```

```
w1=ltisys('tf',n1,d1);
```

```
n2=[0.75 0.33 0.02 -0.00882];
```

```
d2=[1 0.568837 0.091878 0.003019];
```

```
w2=ltisys('tf',n2,d2);
```

- Third, the parameter range has to be specified in order to obtain the variation range of values of the time-varying parameters or uncertain vector. In the considered case study, there are two parameters varying within the operating point. This means that the range of values of this parameters form a multi-dimensional box is given by

```
%Specify the range of parameter values (parameter box)
```

```
Phi1min=.1; Phi1max=.6;
```

```
Phi2min=.1; Phi2max=.6;
```

```
pv=pvec('box', [Phi1min Phi1max; Phi2min Phi2max]);
```

- Fourth, the state space LPV model is transformed into an LTI system and then the parameter varying system is specified as follows:

```
%Specify the parameter-dependent model with PSYS
```

```
s0=ltisys([0 0;0 0],[0.2127;0],[0 1], [0]);
```

```
s1=ltisys([-0.5085 0;0.5085 0],[0;0],[0 0], [0],0); %Phi1_al component
```

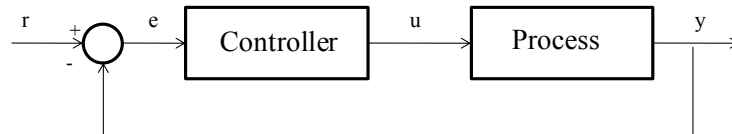
```
s2=ltisys([0 0;0 -0.5085],[0;0],[0 0], [0],0); %Phi2_al component
```

```
pdG=psys(pv,[s0 s1 s2]);
```

- Fifth, the loop shaping structure of the LPV system is specified, where  $r$  is the exogenous input,  $y=K$  represents the outputs generated by the control loop,  $K:r-y$  specify the controller and its inputs,  $G:K$  represents that the input of  $G$  is the output of  $K$  and  $pdG$  is the system matrix.

```
% Specify the loop-shaping control structure with SCONNECT
```

```
[pdP,r]=sconnect('r','y=K','K:r-y','G:K',pdG);
```



**Figure 80. Loop Shaping Structure.**

- Sixth, the augmented plant is formed.

```
% Augment with the shaping filters
```

```
Paug=smult(pdP,sdiag(w1,w2));
```

```
%Paug=smult(pdP);
```

- Seventh, the  $H_\infty$  Gain Scheduling Controller is calculated using *hinfgs* function. This function calculates and  $H_\infty$  gain scheduled control for parameter varying systems with an affine dependence. The parameters are assumed to be measured in real time. To calculate the controller, the function implements the quadratic  $H_\infty$  performance approach where *Paug* is the parameter varying plant, *r* is a vector specifying the dimensions of D22, *1e-2* is the target value for *gopt*, *1e-4* is the desired relative accuracy of the optimal performance of *gopt*, *pdK* is the polytopic representation of the gain-scheduled controller and *gopt* is the optimal performance of the controller. Also, the *psinfo* function gives the type of system (affine or polytopic), the number *K* of system matrices involved in its definition, the number of states, inputs and outputs, respectively.

```
% LMI-BASED SYNTHESIS OF THE LPV CONTROLLER
```

```
% Minimization of gamma for the loop-shaping criterion
```

```
[gopt,pdK]=hinfgs(Paug,r,1e-2,1e-4);
```

```
%typ= affine or polytopic, K=# of K system matrices
```

```
%NS=# states, NI=# inputs and NO=#outputs
```

```
[typ,K,NS,NI,NO]=psinfo(pdK);
```

- Eighth, the desired operating points are specifying in order to return the convex decomposition *c* of *p* over the set *vertex* of box corners.

```
%corresponding state-space parameters of the controller given any
```

```
%p of the parameter vector p(t)
```

```
p1=phi1;
```

```
p2=phi2;
```

```
p=[p1;p2];
```

```
[c,vertex]=polydec(pv,p);
```

```
%NOTE: p has to be inside the range of phi1 y phi2: i.e. [.1 .5; .2 .5]
```

```
% [Phi1min Phi1max; Phi2min Phi2max]
```

- Ninth, the evaluation of the desired operating points in the polytopic representation of the gain-scheduled controller is realized. From this evaluation the state space matrices are extracted and then transformed in to a continuous space system *num\_controller* and *den\_controller* whose values are sent online to the simulation in SIMULINK®.

```
%Controller calculus
```

```
Kp = psinfo(pdK,'eval',c);
```

```
[a,b,c,d,e]=ltiss(Kp);
```

```
A=a;
```

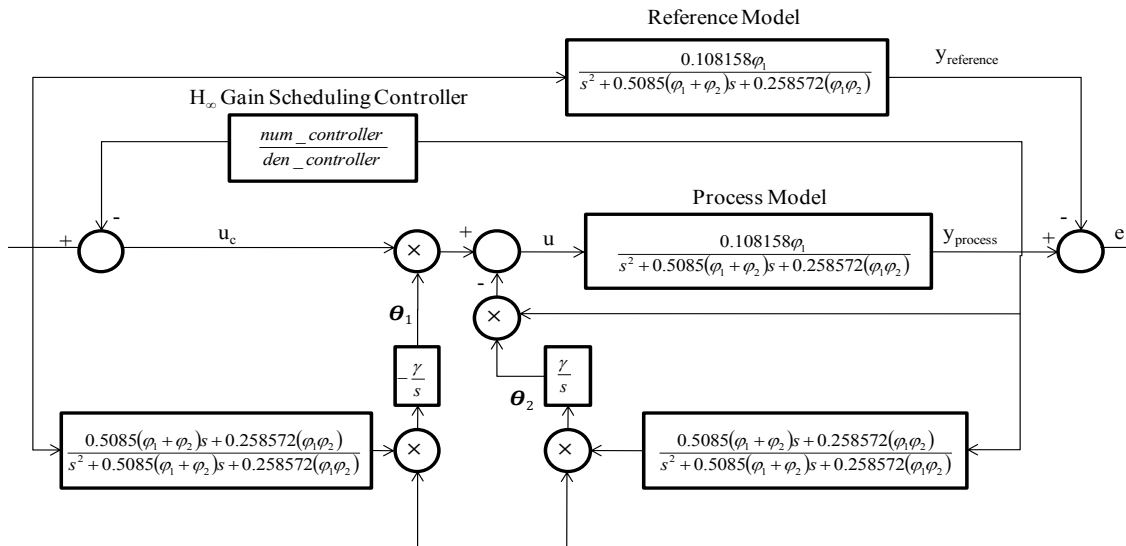
```
B=b;
```

```

C=c;
D=d;
%Transformation from ss to tf
[b,a] = ss2tf(A,B,C,D);
num_controller=b
den_controller=a
modelName = 'MRAC_LPV';
modelName2 = 'MRAC_Hinfgs_LPV';
open_system(modelName)
open_system(modelName2)

```

The  $H_\infty$  Gain Scheduling – MRAC LPV Controller based on MIT rule is represented in Figure 81.



**Figure 81.  $H_\infty$  Gain Scheduling – MRAC LPV Controller based on MIT rule.**

Finally, the  $H_\infty$  Gain Scheduling – MRAC LPV Controller based on Lyapunov theory is shown in Figure 82.

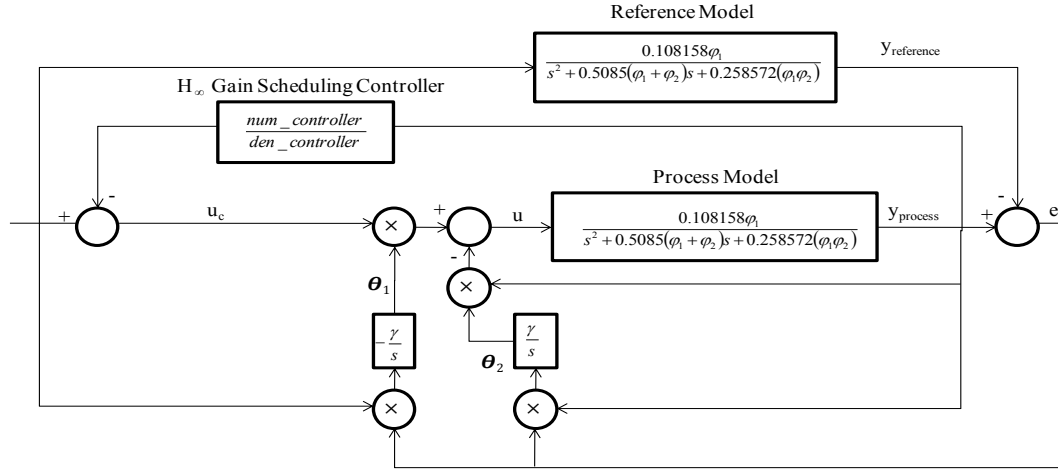


Figure 82.  $H_\infty$  Gain Scheduling – MRAC LPV Controller based on Lyapunov theory.

#### 4.5.1 Experiments and Results

This section explains the different experiments realized in the Coupled-Tank System using the MRAC- $H_\infty$ GS-LPV Controller (MRAC- $H_\infty$ GS-LPV) based on the MIT rule and based on the Lyapunov theory. Two different types of faults were simulated in the implemented schemes: abrupt and gradual faults. The next figures show the results of the experiments realized using this scheme.

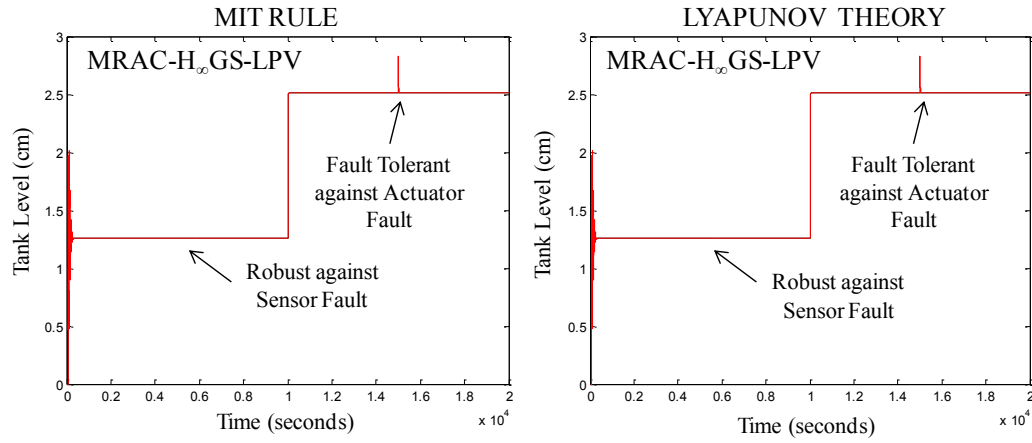
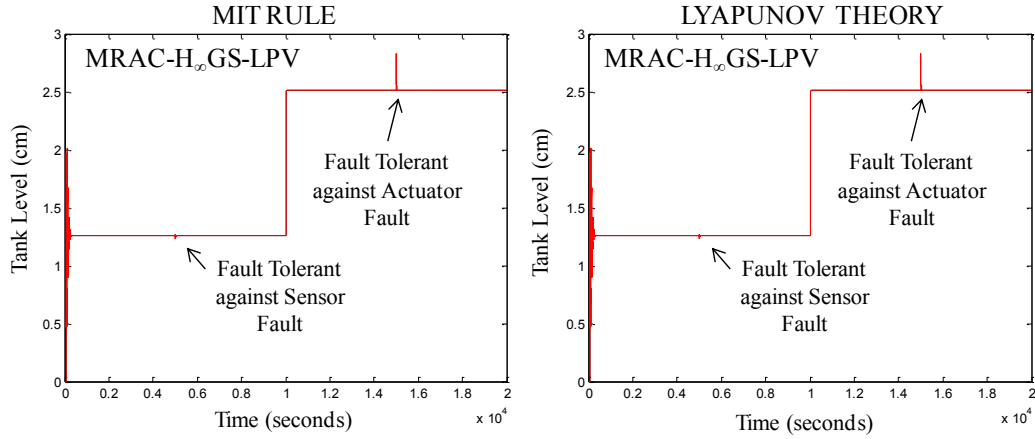


Figure 83. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of 3.3% and an abrupt-actuator fault of 20% for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$ .

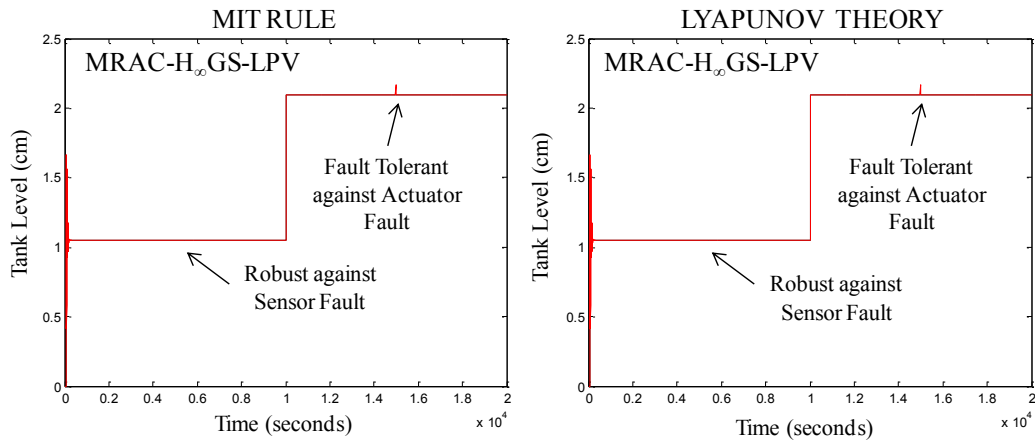
In Figure 83, the MRAC- $H_\infty$ GS-LPV Controllers based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , an abrupt-sensor fault of 3.3% was introduced at time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. It can be

observed that the MRAC- $H_\infty$ GS-LPV was robust against the sensor fault, could tolerate the change in the operating point and was fault tolerant against the actuator fault for both the MIT rule and the Lyapunov theory methods.



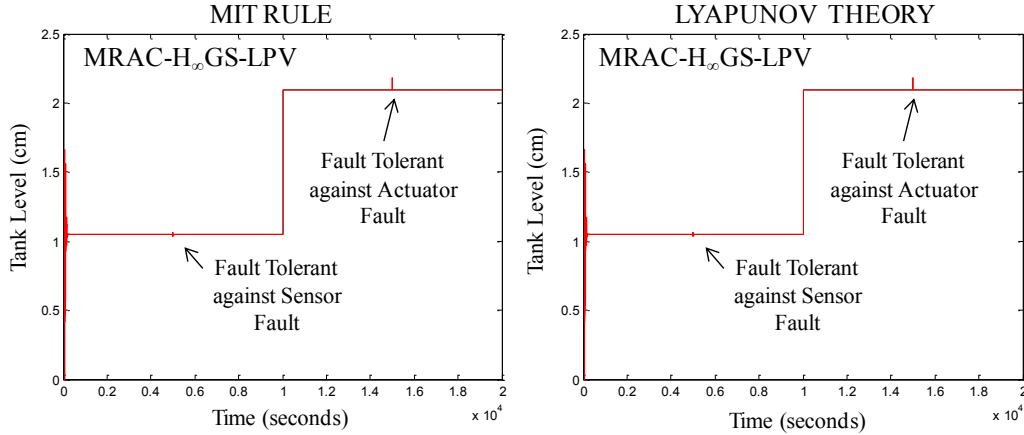
**Figure 84. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with an abrupt-sensor fault of 160% and an abrupt-actuator fault of 20% for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$ .**

In Figure 84, the MRAC- $H_\infty$ GS-LPV Controllers based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , an abrupt-sensor fault of 160% was introduced in time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC- $H_\infty$ GS-LPV was fault tolerant against the sensor and actuator fault and could tolerate the change in the operating point. The above applies for both methodologies (MIT rule and Lyapunov theory).



**Figure 85. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 3.3% and a gradual-actuator fault of 20% for the operating points  $\varphi_1=0.6$  and  $\varphi_2=0.6$ .**

In Figure 85, the MRAC- $H_\infty$ GS-LPV Controllers based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.6$ , a gradual-sensor fault of 3.3% was introduced at time 5000 seconds and an gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC- $H_\infty$ GS-LPV was robust against the sensor fault, could tolerate the change in the operating point and was fault tolerant against the actuator fault for both the MIT rule and the Lyapunov theory methods.



**Figure 86. Comparison between the MRAC- $H_\infty$ GS-LPV Controllers based on the MIT rule and based on the Lyapunov theory with a gradual-sensor fault of 160% and a gradual-actuator fault of 20% for the operating points  $\varphi_1=0.6$  and  $\varphi_2=0.6$ .**

In Figure 86, the MRAC- $H_\infty$ GS-LPV controllers based on the MIT rule and on the Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.6$ , a gradual-sensor fault of 160% was introduced in time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure it can be observed that the MRAC- $H_\infty$ GS-LPV was fault tolerant against the sensor and actuator fault and could tolerate the change in the operating point for both methodologies (MIT rule and Lyapunov theory).

#### 4.5.2 Comparison between the MRAC-LPV and the MRAC- $H_\infty$ GS-LPV based on MIT and based on Lyapunov theory

For each of the two proposed schemes based on the MIT rule and on the Lyapunov theory: MRAC-LPV and MRAC- $H_\infty$ GS-LPV the summary of the experiments results are explained in Table 19 and Table 20. These results explain the range in which the methodologies are robust, fault tolerant or degraded against the fault.

In addition, the Mean Square Error (MSE) was calculated for all the experiments. The results are shown in Table 21. It is important to mention that the results were the same for abrupt and gradual faults.

**Table 19. Results of experiments of the MRAC-LPV and MRAC-H<sub>∞</sub>GS-LPV methodologies based on MIT rule.**

Methodology	Sensor Faults		Actuator Faults	
	Abrupt Faults $0 < f < 3.3\% \rightarrow \text{FT}$ $f > 3.3\% \rightarrow \text{D}$	Gradual Faults $+/-0 < f < +/-3.3\% \rightarrow \text{FT}$ $f > +/- 3.3\% \rightarrow \text{D}$	Abrupt Faults $0 < f < 20\% \rightarrow \text{FT}$ $f > 20\% \rightarrow \text{D}$	Gradual Faults $+/-0 < f < +/- 20\% \rightarrow \text{FT}$ $f > +/- 20\% \rightarrow \text{D}$
MRAC-LPV				
MRAC-H <sub>∞</sub> GS-LPV	FT	FT	$0 < f < 20\% \rightarrow \text{FT}$ $f > 20\% \rightarrow \text{D}$	$+/-0 < f < +/- 20\% \rightarrow \text{FT}$ $f > +/- 20\% \rightarrow \text{D}$

FT = Fault Tolerant, D = Degraded

**Table 20. Results of experiments of the MRAC-LPV and MRAC-H<sub>∞</sub>GS-LPV methodologies based on Lyapunov theory.**

Methodology	Sensor Faults		Actuator Faults	
	Abrupt Faults $0 < f < 3.3\% \rightarrow \text{FT}$ $f > 3.3\% \rightarrow \text{D}$	Gradual Faults $+/-0 < f < +/- 3.3\% \rightarrow \text{FT}$ $f > +/- 3.3\% \rightarrow \text{D}$	Abrupt Faults $0 < f < 20\% \rightarrow \text{FT}$ $f > 20\% \rightarrow \text{D}$	Gradual Faults $+/-0 < f < +/- 20\% \rightarrow \text{FT}$ $f > +/- 20\% \rightarrow \text{D}$
MRAC-LPV				
MRAC-H <sub>∞</sub> GS-LPV	FT	FT	$0 < f < 20\% \rightarrow \text{FT}$ $f > 20\% \rightarrow \text{D}$	$+/-0 < f < +/- 20\% \rightarrow \text{FT}$ $f > +/- 20\% \rightarrow \text{D}$

**Table 21. MSE comparison between the MRAC-LPV and MRAC-H<sub>∞</sub>GS-LPV MIT and Lyapunov based design.**

Operating Point $\varphi_1=0.3$ and $\varphi_2=0.5$		
Methodology	MRAC MIT based design Sensor Faults (3.3%) and Actuator Faults (20%)	MRAC Lyapunov based design Sensor Faults (3.3%) and Actuator Faults (20%)
	0.004683719	0.004806042
MRAC-LPV		
MRAC-H <sub>∞</sub> GS-LPV	0.001259889	0.000996639
Methodology	MRAC MIT based design Sensor Faults (160%) and Actuator Faults (20%)	MRAC Lyapunov based design Sensor Faults (160%) and Actuator Faults (20%)
	9.834450837	9.834345325
MRAC-LPV		
MRAC-H <sub>∞</sub> GS-LPV	0.001216185	0.000942122
Operating Point $\varphi_1=0.6$ and $\varphi_2=0.6$		
Methodology	MRAC MIT based design Sensor Faults (3.3%) and Actuator Faults (20%)	MRAC Lyapunov based design Sensor Faults (3.3%) and Actuator Faults (20%)
	0.002133405	0.002114925
MRAC-LPV		
MRAC-H <sub>∞</sub> GS-LPV	0.000613752	0.000480325
Methodology	MRAC MIT based design Sensor Faults (160%) and Actuator Faults (20%)	MRAC Lyapunov based design Sensor Faults (160%) and Actuator Faults (20%)
	6.83165252	6.831596171
MRAC-LPV		
MRAC-H <sub>∞</sub> GS-LPV	0.000614215	0.000480346

In almost all the results of Table 21, it can be seen that the MSE from the schemes using the MRAC based on the Lyapunov theory is lower than the schemes using the MRAC based on MIT, because the Lyapunov theory adds stability to the closed-loop system.



#### 4.6 Experiments using the MRAC MIT and MRAC Lyapunov based design with the nonlinear model of the system

The experiments presented in the above section, are realized using the as the Process Model the LPV model of the Coupled-Tank system. In this section the experiments are implemented using the nonlinear model of the system, in order to present a more realistic simulation. The nonlinear model of the Coupled-Tank system is represented trough the following equations:

$$\dot{h}_1(t) = -a_1/A_1 \sqrt{2g} \sqrt{h_1(t)} + k_p/A_1 u(t) \quad (151)$$

$$\dot{h}_2(t) = a_1/A_2 \sqrt{2g} \sqrt{h_1(t)} - a_2/A_2 \sqrt{2g} \sqrt{h_2(t)} \quad (152)$$

$$y(t) = h_2(t) \quad (153)$$

The variables definition of the above equations was presented in Table 9 (Chapter 4). The representation of the MRAC-LPV MIT controller, the MRAC-LPV Lyapunov Controller, the  $H_\infty$  Gain Scheduling – MRAC LPV Controller based on MIT controller and the  $H_\infty$  Gain Scheduling – MRAC LPV controller based on Lyapunov are represented from Figures 87 to 90.

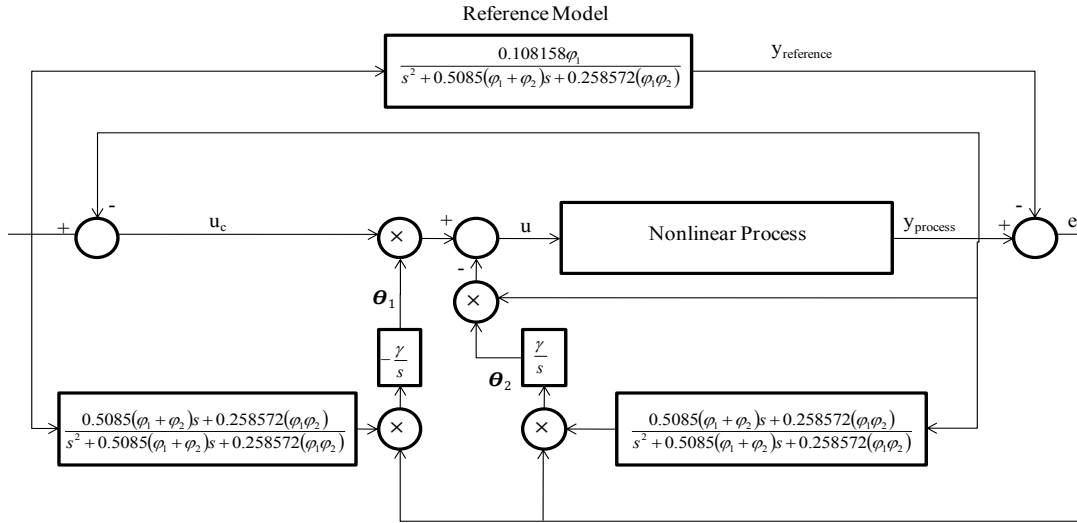
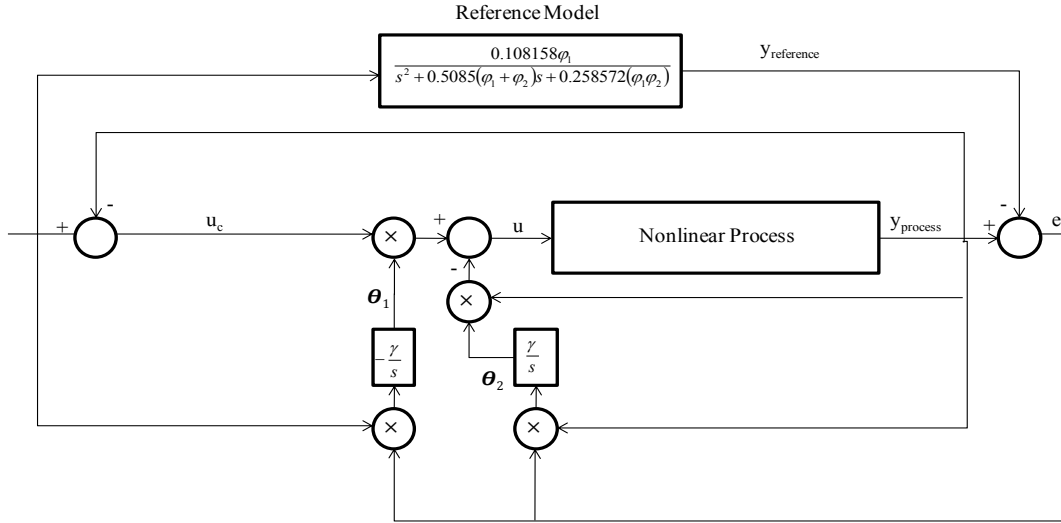
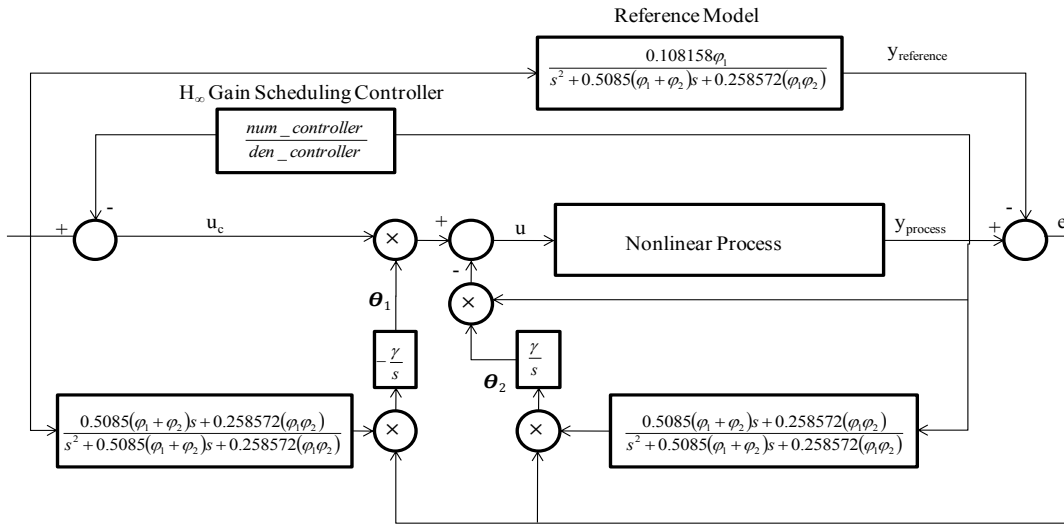


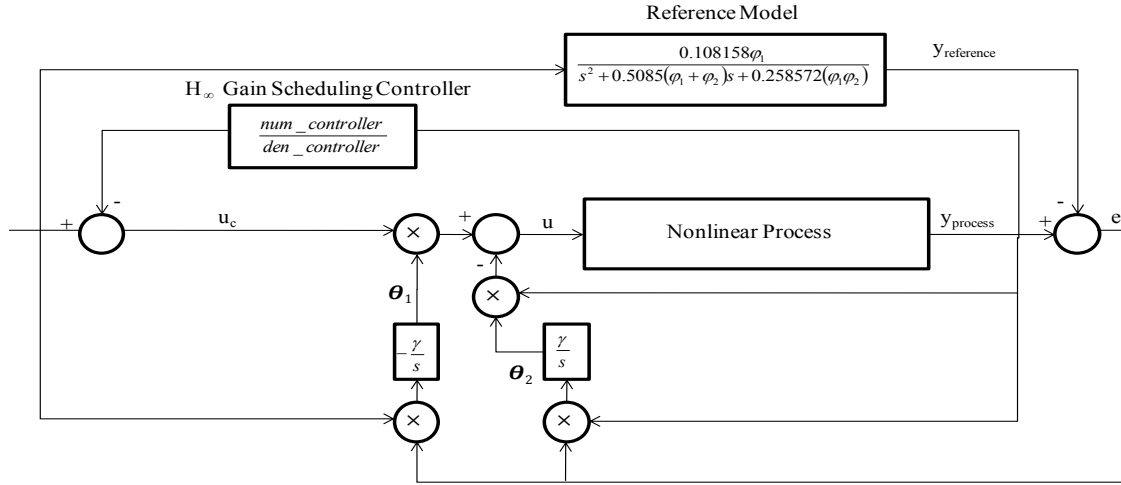
Figure 87. Nonlinear Process MRAC-LPV Controller based on MIT rule.



**Figure 88. Nonlinear Process MRAC-LPV Controller based on Lyapunov theory.**

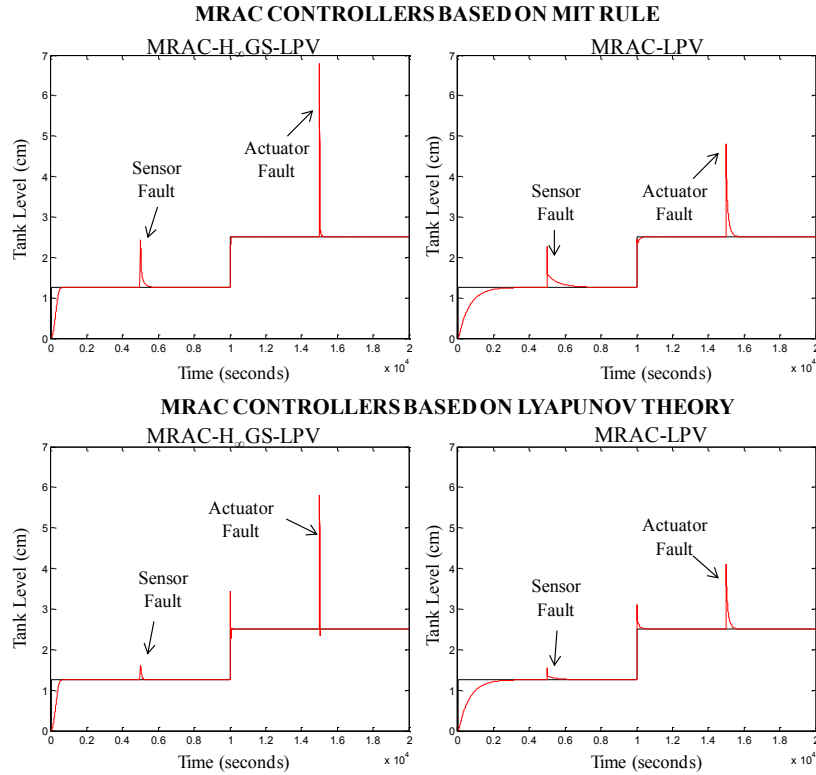


**Figure 89. Nonlinear Process MRAC- $H_\infty$  GS-LPV Controller based on MIT rule.**



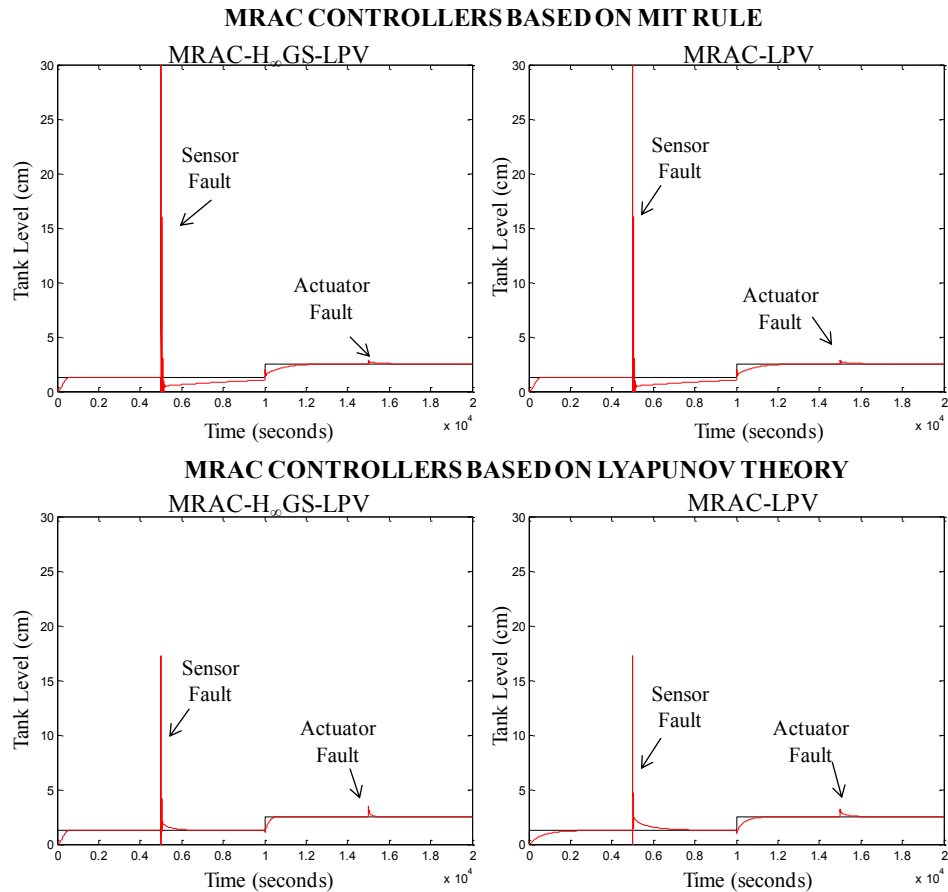
**Figure 90. Nonlinear Process MRAC-H<sub>∞</sub>GS-LPV Controller based on Lyapunov theory.**

When the controllers from Figure 87 to Figure 90 were used with the LPV model the value of  $\gamma$  used was 10000. On the other hand, when the nonlinear model is used instead of the LPV model a value of  $\gamma$  smaller gave better results because the chattering disappears (See Appendix C). The selected value of  $\gamma$  was 0.003. The next figures show the results of the nonlinear model implementation:



**Figure 91. Comparison between the Nonlinear Process MRAC-H<sub>∞</sub>GS-LPV and the Nonlinear Process MRAC-LPV Controllers with an abrupt-sensor fault of 3.3% and an abrupt-actuator fault of 20% for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$ .**

In Figure 91, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , an abrupt-sensor fault of 3.3% was introduced at time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the four different controllers could accommodate the fault and tolerate the change in the operating point. But in the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC- $H_\infty$ GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes.

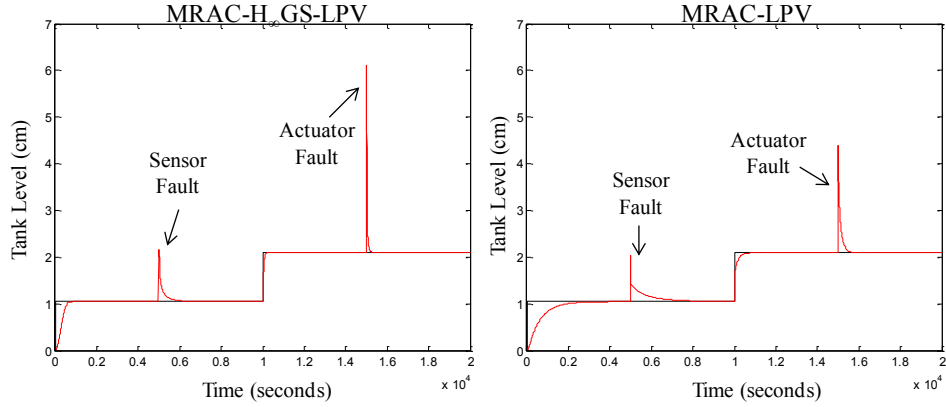


**Figure 92. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with an abrupt-sensor fault of 166% and an abrupt-actuator fault of magnitude 20% for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$ .**

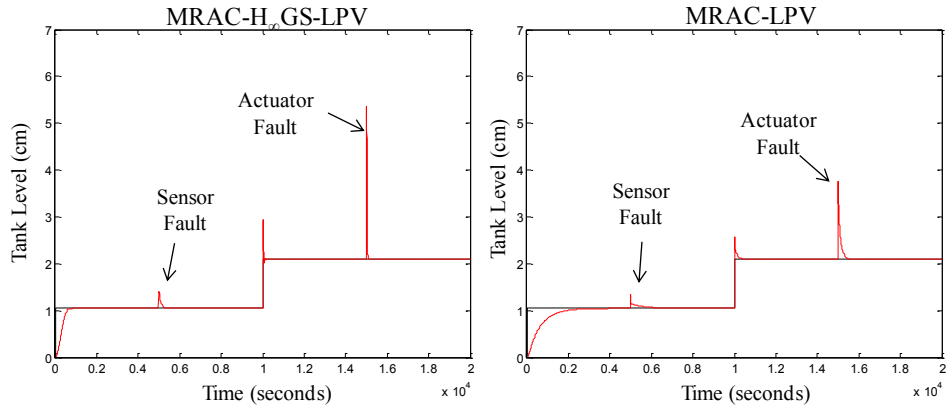
In Figure 92, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , an abrupt-sensor fault of 166% was introduced in time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV were unfeasible

against the sensor fault because the tank level limit is 30 cm and these controllers were beyond this limit. On the other hand, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV based on Lyapunov theory were able to accommodate the sensor and the actuator faults. But the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory showed a faster adaptation performance in comparison with the MRAC-LPV.

#### MRAC CONTROLLERS BASED ON MIT RULE

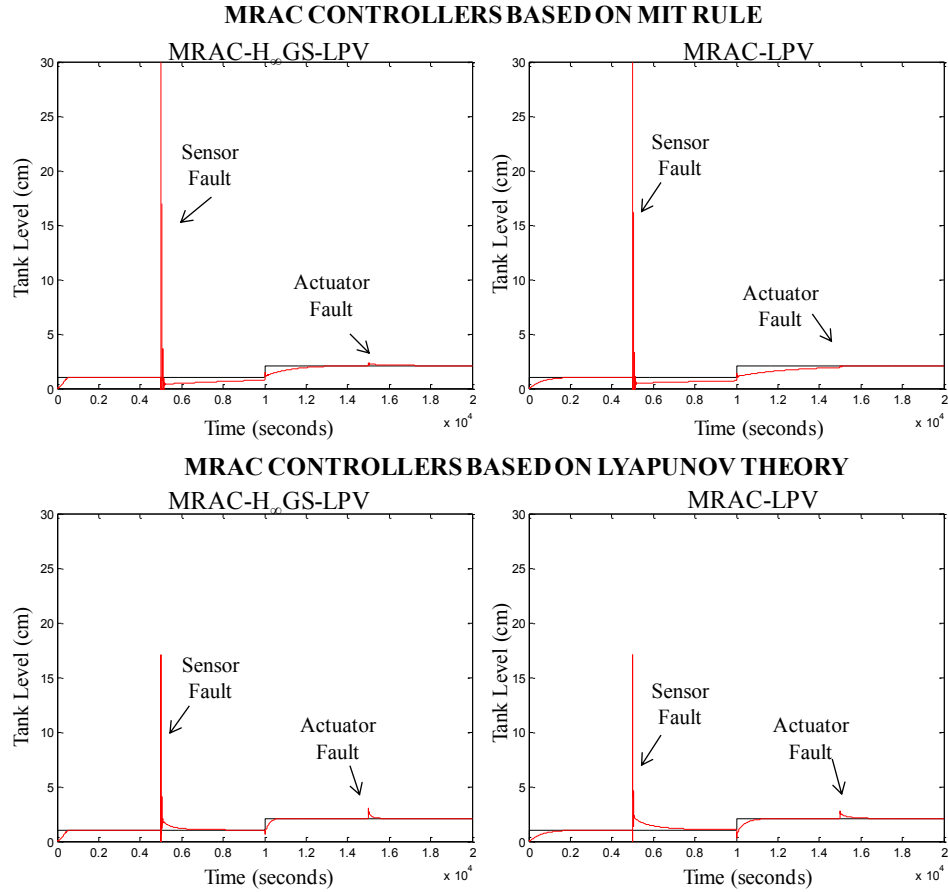


#### MRAC CONTROLLERS BASED ON LYAPUNOV THEORY



**Figure 93. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with a gradual-sensor fault of 3.3% and a gradual-actuator fault of 20% for the operating points  $\phi_1=0.6$  and  $\phi_2=0.6$  .**

In Figure 93, for gradual additive faults, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\phi_1=0.6$  and  $\phi_2=0.6$ , a gradual-sensor fault of 3.3% was introduced at time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the four different controllers could accommodate the fault and tolerate the change in the operating point. But in schemes based on the MIT rule the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC- $H_\infty$ GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes.



**Figure 94. Comparison between the Nonlinear Process MRAC-H<sub>∞</sub>GS-LPV and the Nonlinear Process MRAC-LPV Controllers with a gradual-sensor fault of 166% and a gradual-actuator fault of 20% for the operating points  $\phi_1=0.6$  and  $\phi_2=0.6$ .**

In Figure 94, for gradual additive faults, the MRAC-H<sub>∞</sub>GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\phi_1=0.6$  and  $\phi_2=0.6$ , a gradual-sensor fault of 166% was introduced at time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this Figure, it can be observed that the MRAC-H<sub>∞</sub>GS-LPV and the MRAC-LPV were unfeasible against the sensor fault because the tank level limit is 30 cm and these controllers were beyond this limit. On the other hand, the MRAC-H<sub>∞</sub>GS-LPV and the MRAC-LPV based on Lyapunov theory were able to accommodate the sensor and the actuator faults. But the MRAC-H<sub>∞</sub>GS-LPV based on Lyapunov theory showed a faster adaptation performance in comparison with the MRAC-LPV.

In addition, to compare the Nonlinear Process controller based on the MIT rule and the Nonlinear Process controller based on Lyapunov theory the Mean Square Error (MSE) was calculated for all the experiments. The results are shown in Table 22.

**Table 22. MSE Results of the comparison between the Nonlinear MRAC-LPV and the Nonlinear MRAC-H<sub>∞</sub>GS-LPV MIT and Lyapunov based design.**

Operating Point $\phi_1=0.3$ and $\phi_2=0.5$		
Methodology	MRAC MIT based design	MRAC Lyapunov based design
	Sensor Faults (3.3%) and Actuator Faults (20%)	Sensor Faults (3.3%) and Actuator Faults (20%)
MRAC-LPV	0.0379	0.0325
MRAC-H <sub>∞</sub> GS-LPV	0.0300	0.0244
Methodology	Sensor Faults (166%) and Actuator Faults (20%)	Sensor Faults (166%) and Actuator Faults (20%)
MRAC-LPV	1.0810	0.1324
MRAC-H <sub>∞</sub> GS-LPV	1.1672	0.1178
Operating Point $\phi_1=0.6$ and $\phi_2=0.6$		
Methodology	MRAC MIT based design	MRAC Lyapunov based design
	Sensor Faults (3.3%) and Actuator Faults (20%)	Sensor Faults (3.3%) and Actuator Faults (20%)
MRAC-LPV	0.0330	0.0253
MRAC-H <sub>∞</sub> GS-LPV	0.0278	0.0210
Methodology	Sensor Faults (166%) and Actuator Faults (20%)	Sensor Faults (166%) and Actuator Faults (20%)
MRAC-LPV	1.0799	0.1348
MRAC-H <sub>∞</sub> GS-LPV	1.1316	0.1193

## 4.7 Experiments and Results using the Coupled-Tank System as testbed implementing Multiplicative Faults

In addition to the above experiments multiplicative faults were tested. This type of faults is represented as a degradation of the nominal system. For example, actuator multiplicative fault is represented as follows:

$$u_f = \alpha u \quad (154)$$

where  $u_f$  represent the system input with the actuator fault,  $\alpha$  represents the degradation percentage of the actuator, and  $u$  is the nominal system input. The above type of faults were implemented in each of the following proposed schemes based on the MIT rule and on the Lyapunov theory: MRAC-LPV Controller and  $H_\infty$  Gain Scheduling MRAC-LPV Controller (MRAC- $H_\infty$ GS-LPV) for LPV systems, MRAC-LPV Controller and  $H_\infty$  Gain Scheduling MRAC-LPV Controller for nonlinear systems, MRAC-LPV Controller for LPV systems and MRAC-LPV Controller for nonlinear.

### 4.7.1 Multiplicative Faults applied in the LPV System

First, the multiplicative faults were tested in the LPV system. The results of these experiments are shown in Table 23. These results explain if the methodologies are robust, fault tolerant or degraded against the simulated fault and also demonstrate the Mean Square Error (MSE).

**Table 23. Results of experiments using multiplicative sensor (5000 seconds) and actuator (15000 seconds) faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for LPV systems.**

Methodology	MSE Sensor and Actuator Faults			
	0 = 100%	0.1 = 90%	0.5 = 50%	0.95 = 5%
MRAC-LPV based on MIT rule	FT→SF	FT→SF	FT→SF	FT→SF
	D→AF MSE=1.57169	D→AF MSE=1.27322	D→AF MSE=0.39341	D→AF MSE=0.00425
MRAC-LPV based on Lyapunov theory	FT→SF	FT→SF	FT→SF	FT→SF
	D→AF MSE=1.57128	D→AF MSE=1.27283	D→AF MSE=0.39308	D→AF MSE=0.00415
MRAC- $H_\infty$ GS-LPV based on MIT rule	R→SF	R→SF	R→SF	R→SF
	D→AF MSE=1.57158	D→AF MSE=1.27321	D→AF MSE=0.39381	D→AF MSE=0.00514
MRAC- $H_\infty$ GS-LPV based on Lyapunov theory	R→SF	R→SF	R→SF	R→SF
	D→AF MSE=1.57130	D→AF MSE=1.27293	D→AF MSE=0.39353	D→AF MSE=0.00486

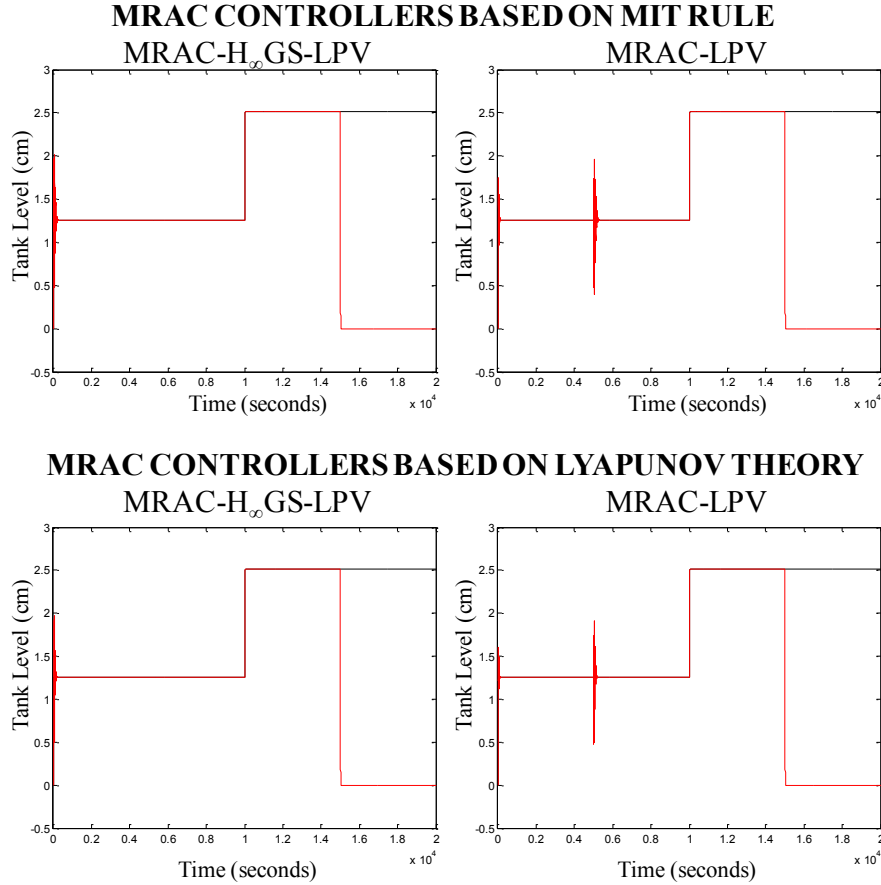
AF=Actuator Fault, D=Degraded, FT=Fault Tolerant, R=Robust, SF=Sensor Fault

In Table 23, it is observed that none of the implemented methodologies was absolutely fault tolerant to the combination of sensor fault (5000 seconds) and actuator fault (15000 seconds). For example, the MRAC-LPV based on Lyapunov theory was Fault Tolerant for the sensor fault and was degraded for the actuator fault. On the other hand the MRAC- $H_\infty$ GS-LPV based on MIT rule and based on Lyapunov theory are robust against the sensor fault and degraded against the actuator fault. The next figures represent the

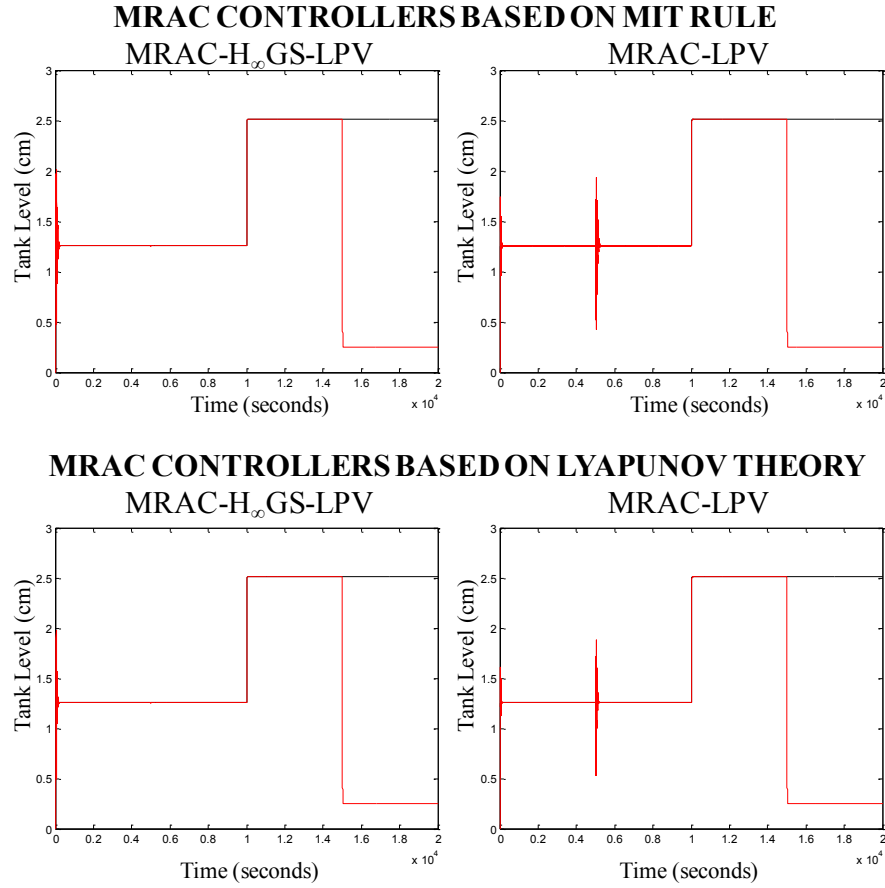


implementation of the above experiments. In addition single multiplicative faults in sensors and actuators were also tested, see Appendix B.

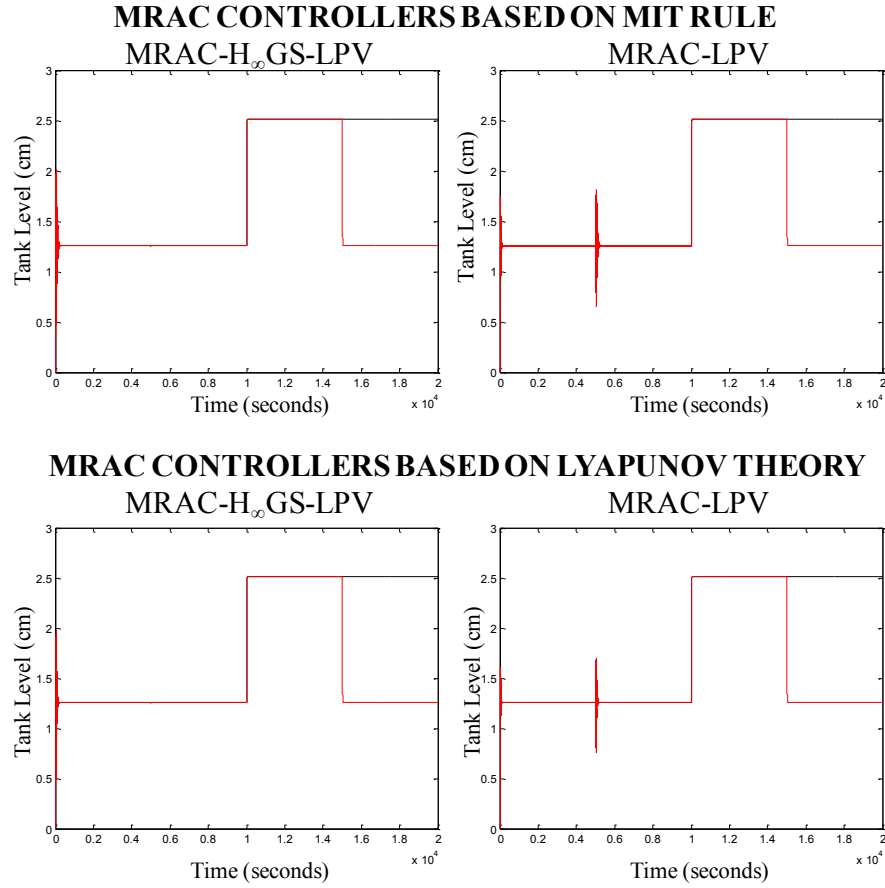
From Figure 95 to Figure 98, it can be observe that for the combination of multiplicative faults in sensors at time 5000 and multiplicative faults in actuators at time 15000 of 100%, 90%, 50% and 5%, the MRAC- $H_\infty$ GS-LPV controller was robust against the multiplicative faults in sensors and became degraded after the multiplicative faults in actuator. On the other hand, the MRAC-LPV scheme was fault tolerant to the multiplicative faults in sensor and also became degraded after the occurrence of the multiplicative faults in actuator. The above apply for the methodologies based on the MIT rule and the methodologies based on the Lyapunov theory. Also, it can be observe that both controllers (MRAC- $H_\infty$ GS-LPV and MRAC-LPV) were able to manage the change in the operating point at 10000 seconds.



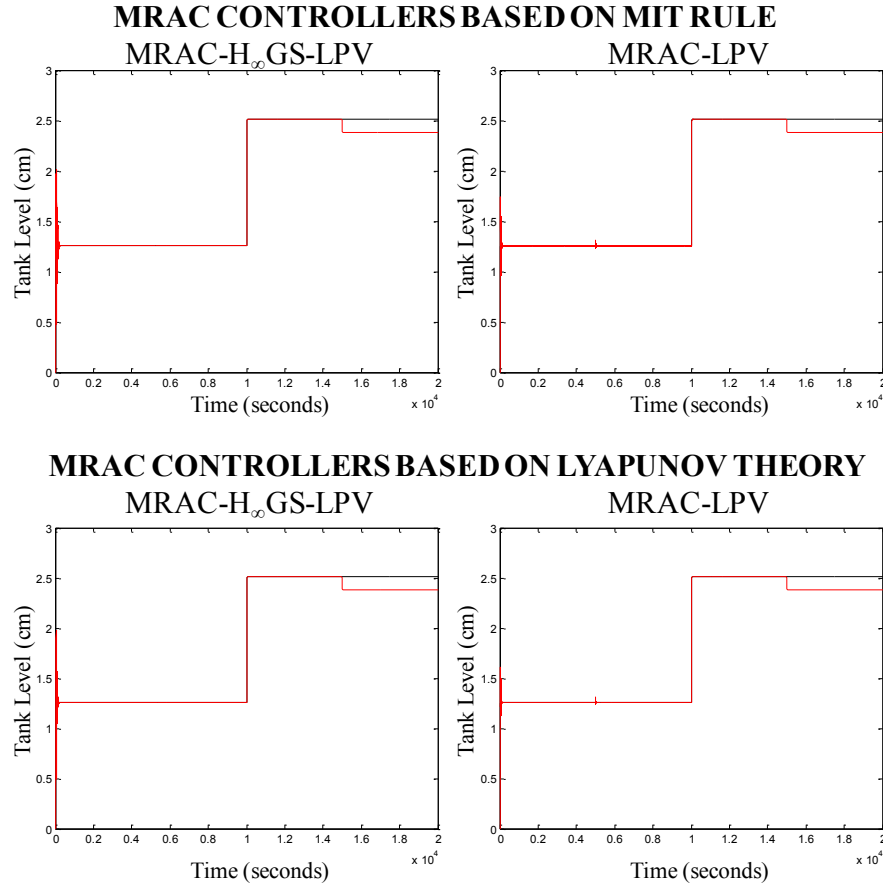
**Figure 95.** Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 100%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.



**Figure 96.** Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.



**Figure 97. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 50%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.**



**Figure 98.** Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

#### 4.7.2 Multiplicative Faults applied in the Nonlinear System

After testing the multiplicative faults in the LPV system, these types of faults were tested using the nonlinear system instead of the LPV system. The results of these experiments are shown in Table 24. These results explain if the methodologies are fault tolerant or degraded and also show the Mean Square Error (MSE). In addition, from Figure 99 to Figure 102 the experiments showed in Table 24 are represented.

Table 24. Results of experiments using multiplicative sensor (5000 seconds) and actuator (15000 seconds) faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for nonlinear systems.

Methodology	MSE Sensor and Actuator Faults			
	100%	90%	50%	5%
MRAC-LPV based on MIT rule	Degraded MSE=3.5676	Fault Tolerant MSE=0.5057	Fault Tolerant MSE=0.0379	Fault Tolerant MSE=0.0281
MRAC-LPV based on Lyapunov theory	Degraded MSE=3.5675	Fault Tolerant MSE=0.5055	Fault Tolerant MSE=0.0321	Fault Tolerant MSE=0.0278
MRAC- $H_\infty$ GS-LPV based on MIT rule	Degraded MSE=3.5559	Fault Tolerant MSE=0.3447	Fault Tolerant MSE=0.0318	Fault Tolerant MSE=0.0179
MRAC- $H_\infty$ GS-LPV based on Lyapunov theory	Degraded MSE=3.5551	Fault Tolerant MSE=0.3442	Fault Tolerant MSE=0.0302	Fault Tolerant MSE=0.0171

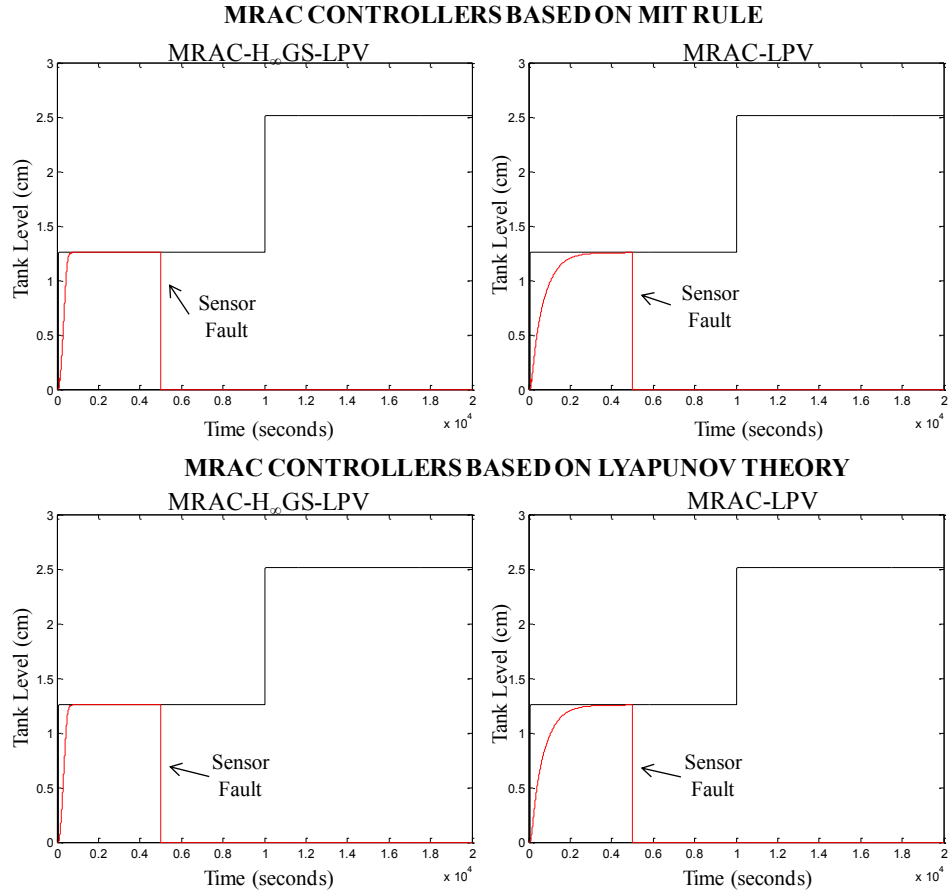
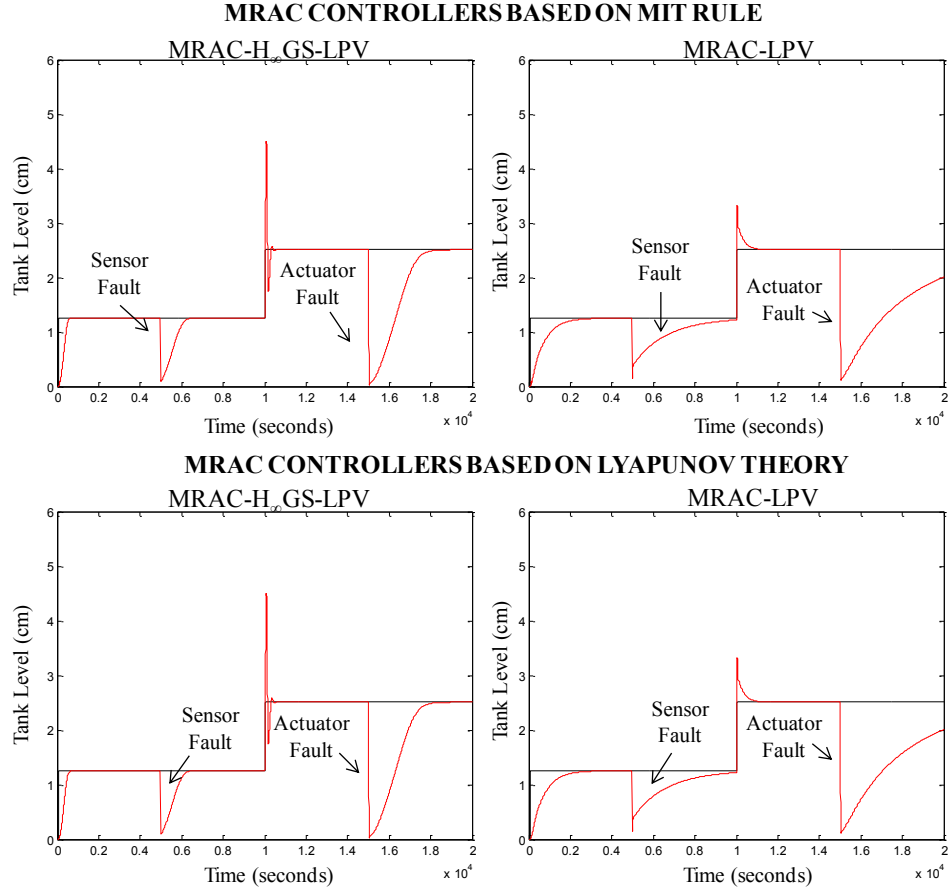


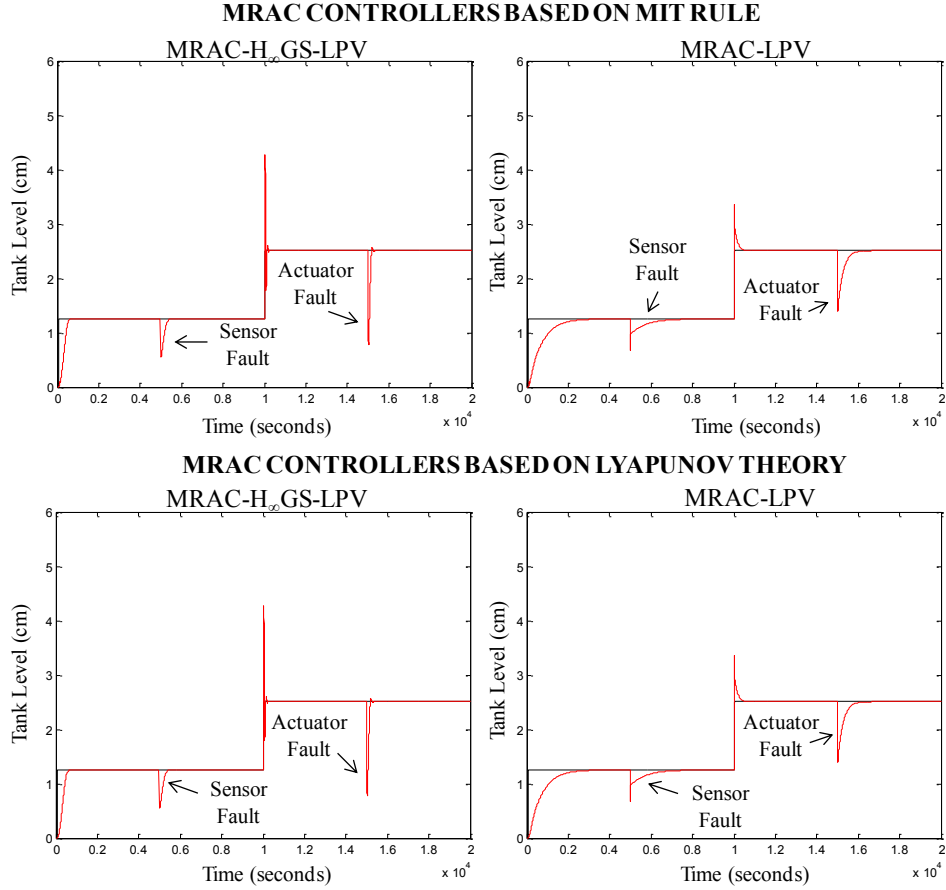
Figure 99. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 100%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.

In Figure 99, for the combination of multiplicative sensor fault of 100% at 5000 seconds and multiplicative actuator fault of 100% at 15000 seconds, it can be observe that the four controller became degraded after the occurrence of the multiplicative sensor fault and non of the controllers were able to return to the desired tank level.



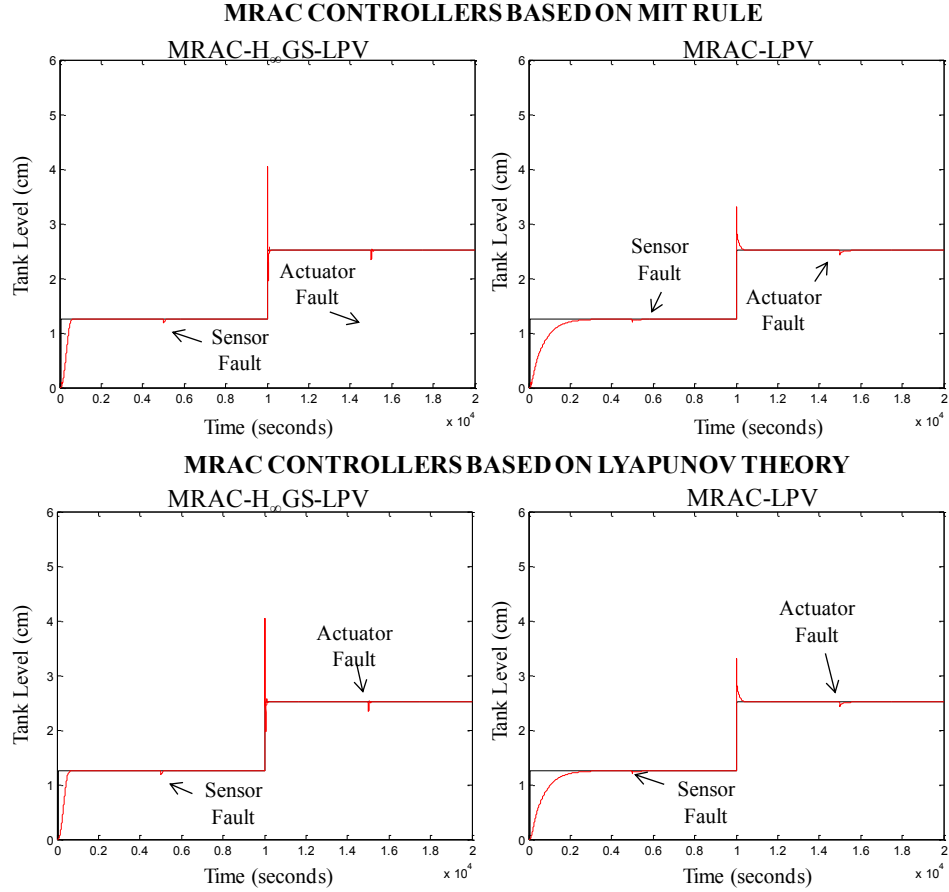
**Figure 100. Comparison between the system output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure 100, for the combination of multiplicative sensor fault of 90% at 5000 seconds and multiplicative actuator fault of 90% at 15000 seconds, it can be observe that MRAC- $H_{\infty}$ GS-LPV based on the MIT rule and based on Lyapunov theory were able to accommodate both types of faults and could tolerate the change in the operating point. On the other hand, the MARC-LPV based on the MIT rule and on Lyapunov theory controllers were able to accommodate the multiplicative sensor fault but could not accommodate in time the multiplicative actuator fault. The  $H_{\infty}$ GS helps the MRAC to achieve a faster adaptation mechanism.



**Figure 101. Comparison between the system output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 50%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure 101, for the combination of multiplicative sensor fault of 50% at 5000 seconds and multiplicative actuator fault of 50% at 15000 seconds, it can be observed that the four different controllers could accommodate the faults (multiplicative sensor and multiplicative actuator) and tolerate the change in the operating point. But in the schemes based on the MIT, the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC-H<sub>∞</sub>GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes.



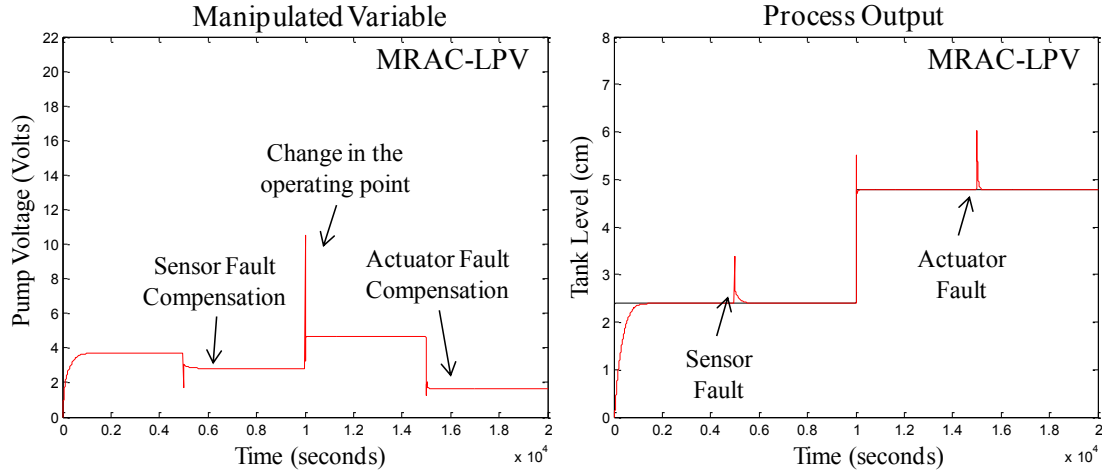
**Figure 102. Comparison between the system output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure 102, for the combination of multiplicative sensor fault of magnitude 0.95 at 5000 second and multiplicative actuator fault of 5% at 15000 seconds, it can be observed that the four different controllers could accommodate the faults (multiplicative sensor and multiplicative actuator) and tolerate the change in the operating point. But in the schemes based on the MIT, the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC-H<sub>∞</sub>GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes.

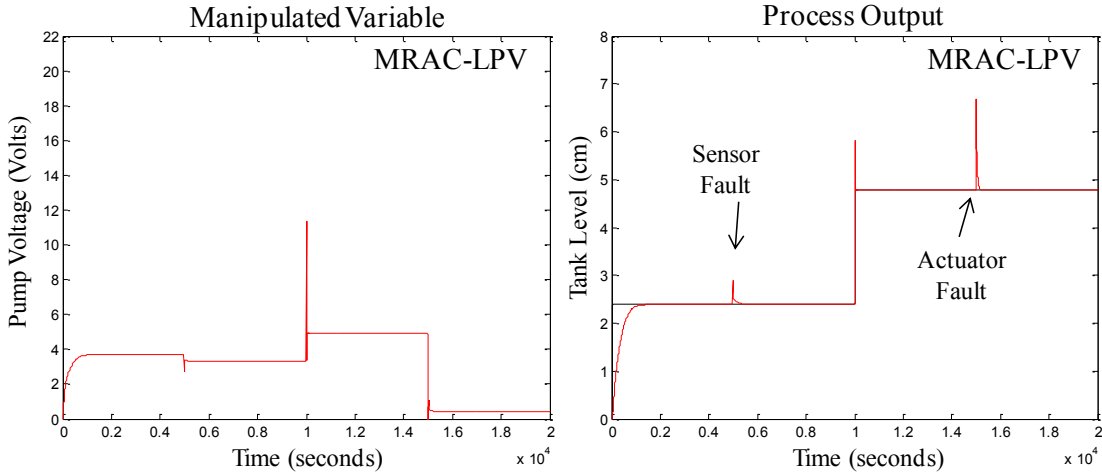


#### 4.8 Manipulated Variable Analysis for the Coupled Tank System

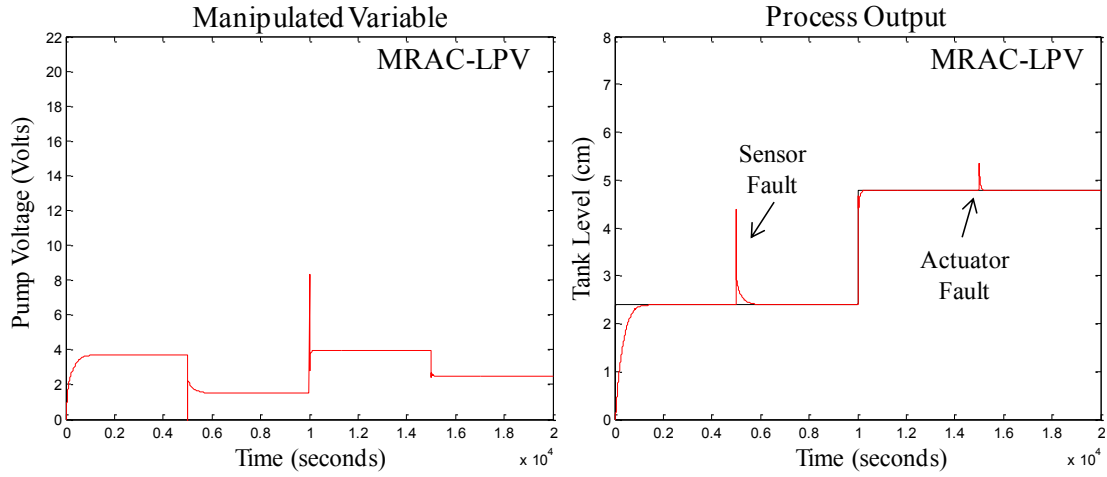
The next figures show different experiments that were realized to analyze the manipulated variable. These experiments were carried on in the operating point  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and different types and sizes of faults were applied.



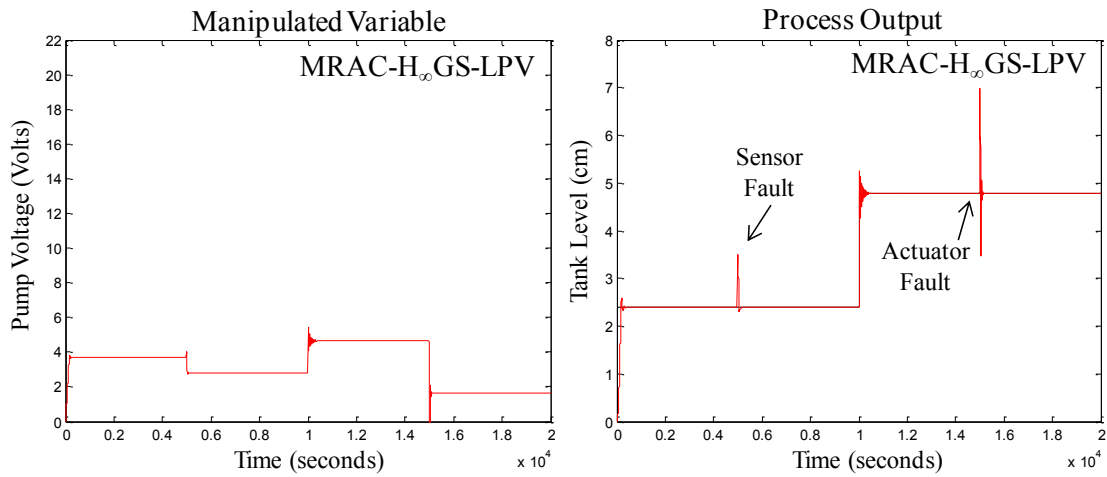
**Figure 103.** Comparison between the manipulated variable and the system output of the MRAC-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) and an additive actuator fault (15000 seconds) of 10% in the operating points  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and a change in the operating point at time 10000 seconds for the nonlinear system.



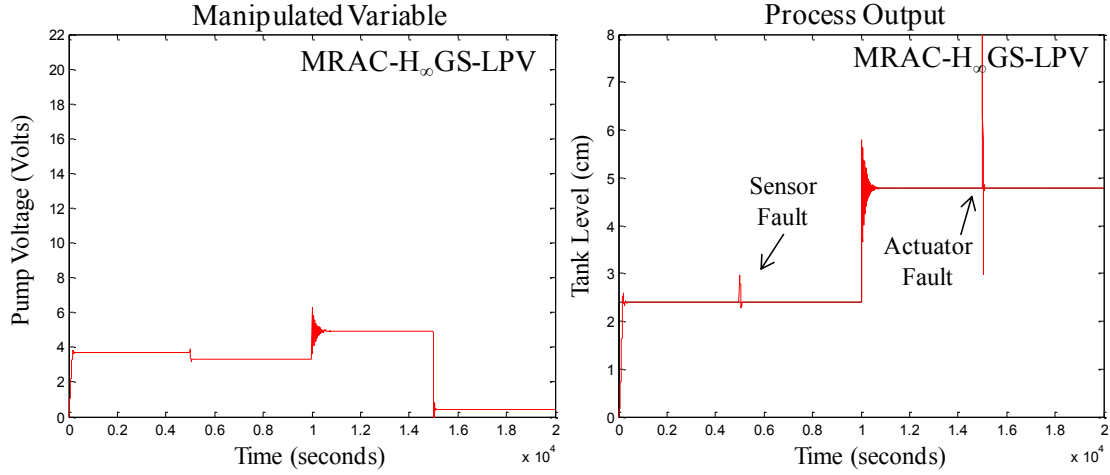
**Figure 104.** Comparison between the manipulated variable and the system output of the MRAC-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 5% and an additive actuator fault (15000 seconds) of 15% in the operating points  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and a change in the operating point at time 10000 seconds for the nonlinear system.



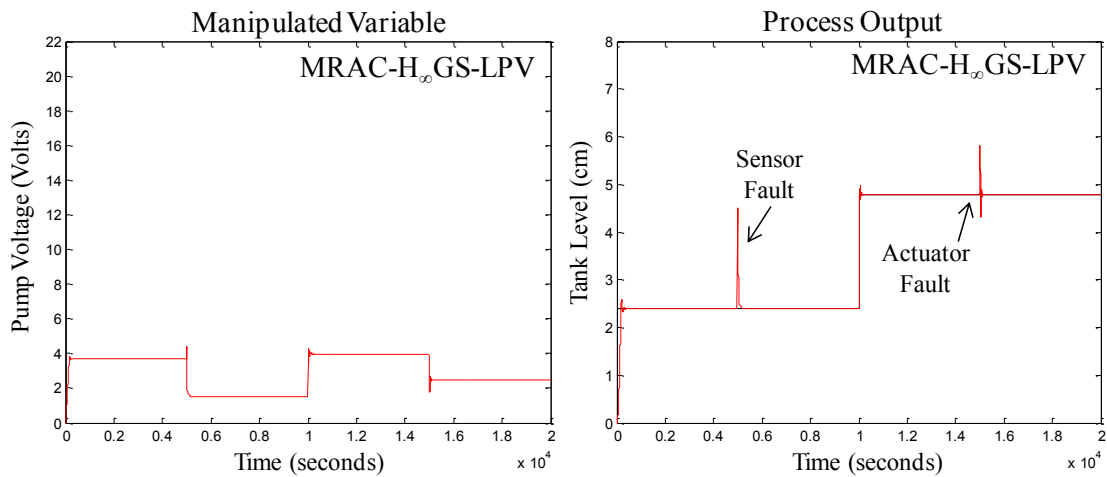
**Figure 105.** Comparison between the manipulated variable and the system output of the MRAC-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 20% and an additive actuator fault (15000 seconds) of 5% in the operating points  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and a change in the operating point at time 10000 seconds for the nonlinear system.



**Figure 106.** Comparison between the manipulated variable and the system output of the MRAC-H<sub>∞</sub> GS-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) and an additive actuator fault (15000 seconds) of 10% in the operating points  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and a change in the operating point at time 10000 seconds for the nonlinear system.



**Figure 107.** Comparison between the manipulated variable and the system output of the MRAC- $H_\infty$ GS-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 5% and an additive actuator fault (15000 seconds) of 15% in the operating points  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and a change in the operating point at time 10000 seconds for the nonlinear system.



**Figure 108.** Comparison between the manipulated variable and the system output of the MRAC- $H_\infty$ GS-LPV schemes based on Lyapunov theory, applying an additive sensor fault (5000 seconds) of 20% and an additive actuator fault (15000 seconds) of 5% in the operating points  $\varphi_1=0.35$  and  $\varphi_2=0.35$  and a change in the operating point at time 10000 seconds for the nonlinear system.

In the above figures the behavior of the manipulated variable under different experiments can be observed. In all the above experiments when the system starts the pump voltage increases to increase the water level, when a positive fault occurs the pump voltage decreases in order to take the tank level to the desired value. Also, when the operating point change, the pump voltage increases to increase the tank level. And finally, when a positive actuator faults appears the pump voltage decreases to decrease the tank level and return it to the desired value.

## 4.9 Comparison with Similar Approaches

Some other MRAC approaches in the last years have been developed. All of them are different from the one proposed in this research. Cho et al. (1990) published a method for FTC systems using a pole assignment controller and a MRAC controller to guarantee the system performance in the presence of a fault. This scheme is different from our work because it did not use any artificial intelligence method such as ANN or GA optimizer. Jia & Jingping (1997) introduced a MRAC based on a PID controller and a GA, in which the GA was used to optimize the PID parameters. This scheme was used in a Continuous Stirred Tank Reactor system (CSTR), and an Artificial Neural Network trained by the GA was used to estimate the state value of the CSTR. The differences between this scheme and the one presented in this research is that the Artificial Neural Network in Jia & Jingping (1997) is used as an estimator of the plant and in this research the Artificial Neural Network is used as a trajectory controller to follow the ideal system trajectory (normal operation mode). Ahmed (2000) presented an Artificial Neural Network based state feedback MRAC for a type of nonlinear plants. This methodology uses a time varying pseudo-linear feedback control in which the gain of the state feedback is generated from the ANN output. This scheme is different from the proposed structure presented in this research, first because it did not uses a GA to optimize a PID controller and also because the ANN is used just to approximate the controller parameters not as a trajectory controller. Zhu et al. (2000) presented an MRAC application with an aged actuator in order to identify vulnerable devices or control parameters on stability. This scheme did not use any artificial intelligence method (ANN or GA) and neither uses a PID controller. Thanapalan et al. (2006) introduces a MRAC and quaternion based adaptive attitude control (QAAC) in order to determinate and isolates faults. This scheme did not use an ANN controller and neither uses a GA optimizer to obtain the best PID controller parameters. Zhang and Li (2006) proposed a single neuron PID-MRAC based on Radial Basis Function (RBF) Neural Network on-line identification. In this approach, the RBF Artificial Neural Network is used to identify the system on-line in order to built the reference model and develop the controller parameters self-learning employing a single neuron controller. This method did not use a GA to optimize the PID controller; also, the PID controller is not used as a feedforward controller, and the RBF ANN was used to build the reference model. In our case, the reference model was constructed using real data system identification from an Industrial Heat Exchanger and the ANN controller is used as a trajectory control. Bayati (2008) presented a MRAC based on the PID-GA combination. The GA is used to obtain the best PID parameters. The difference between the new proposed structure and this method is that it did not use an Artificial Neural Network controller. Hongjie & Bo (2008) showed a MRAC controller based on an on-line Artificial Neural Network and a traditional PID controller used for servo system tracking control. In this scheme, the Artificial Neural Network controller was implemented to reduce the unknown model dynamics, the disturbances and parameter variations. The Artificial Neural Network weights and the MRAC parameters are updated using Lyapunov stability theory. The differences from this scheme and the one proposed in this research are that the Hongjie & Bo (2008) scheme did not use a GA to optimize the PID controller. Moreover, the controller is not a complete PID since it is just a Proportional Controller and the ANN is used to reduce the unknown model dynamics and it is not

used as a trajectory control. Demetriou et al. (2009) developed a theoretical framework of the MRAC design, adaptive parameter identification and automated fault detection of a positive real infinite dimensional system in which for each design problem the well-posedness and stability were investigated. Tan et al. (2011) presented a discrete-time multiple-model MRAC scheme for adaptive actuator failure compensation, this scheme has the capacity to achieve faster and accurate compensation of failure uncertainties. This scheme is different than the one proposed in this thesis because it does not combine the MRAC with other methods such as ANN, robust control, nonlinear control or LPV systems.

In the case of Robust  $H_\infty$  controller, although there are some publications where the  $H_\infty$  technique has been combined with other schemes (Hwang & Chen, 1998; Lian et al., 2002; Yu, 2004; Miyasato, 2007; Miyasato, 2008), to the best of our knowledge there are no reports concerning the combination of MRAC with  $H_\infty$ .

For the MRAC controller for LPV systems (or LPV controls) just a few studies had been developed. In Hsu (1990), a Variable Structure MRAC controller (VS-MRAC) was proposed. The benefit of this type of controller is a nicer transient behavior, a disturbance rejection capability, nonlinearities or parameter variations insensitivity and robustness against unmodeled dynamics. The difference between a Variable Structure (VS) system and an LPV system is that VS case is based on switching functions that creates a sliding surface. If the surface is accomplishing the switching function maintain the trajectory on the surface, the desired system dynamics is achieved. In Hsu et al. (1994), a VS-MRAC control was also developed. This structure uses only input and output data. Also, the controller showed a high-gain stability property which guarantees the elimination of the chattering when linear zones are induced. In Yan et al. (2006), an output feedback VS-MRAC controller was developed. This methodology is based on a classic VS-MRAC structure with a high gain switching mechanism to adapt the VS control signal. The advantages of this methodology are to guarantee a pre-specified steady-state and transient performance specifications for tracking error, the high frequency gain sign is not required a priori, the reference model of the MRAC controller does not need to be strictly positive real and the plant input disturbance can be rejected completely.

Miyasato (2006) presented a Model Reference Adaptive Controller for polytopic LPV systems in fixed polytopes denoted by convex hulls of extreme systems (uncertainties of system parameters). In this methodology, the control input is represented by a weighted sum of the extreme systems control signals and the weights are regulated adaptively. In addition, stabilizing signals are aggregated to manage the time-varying components effect in uncertain processes and to stabilize the plants. These signals are developed as a solution of nonlinear  $H_\infty$  control problem for a virtual system. This methodology is different from the one proposed in this work (MRAC- $H_\infty$  Gain Scheduling-LPV) because the LPV system is also a polytopic system but the control input is composed of the feedforward update rule ( $\theta_1$ ), the feedback update rule ( $\theta_2$ ) and the output of the  $H_\infty$  Gain Scheduling controller. Also, the  $H_\infty$  control problem is solved using the  $H_\infty$  Gain Scheduling LPV method and not a simple  $H_\infty$  control. In Miyasato (2008), to guarantee stability just a Lyapunov positive definite function is established and in our work to guarantee stability the MIT rule and the Lyapunov theory method are used.

In Abdullah & Zribi (2008), a model reference control for LPV systems was developed. The used LPV system parameters can be measured on-line and are bounded. The development of this type of control was divided into parts: the first part was to design the matrix coefficients of the controller. To obtain these coefficients singular value decomposition (SVD) was employed. The second part was the design of the state feedback controller gain; in order to acquire this gain a LMIs methodology was used. One of the main differences between the methodology in Abdullah & Zribi (2008) and the one proposed in this work is that the model reference control used in Abdullah & Zribi (2008) is not a model reference adaptive control. In addition, the control law parameters in our case can be obtained through the feedforward update rule ( $\theta_1$ ), the feedback update rule ( $\theta_2$ ) and the output of the  $H_\infty$  Gain Scheduling LPV controller and not with SVD or LMIs. Another important difference is that Abdullah & Zribi (2008) do not use  $H_\infty$  Gain Scheduling LPV Control. Finally, Montes de Oca et al., (2009) proposed a FTC design using LPV admissible model matching; this strategy is an active technique and requires the detection and estimation of the fault by a FDI scheme, then the controller can be redesigned. The faults are expresses as changes in the system dynamics and also are considered as scheduling variables in the LPV model to allow the controller reconfiguration. The main different with the schemes proposed in this thesis are the following: First, the schemes proposed in this thesis are passive because are designed off-line. Second, the different presented schemes (MRAC, MRAC-ANN, MRAC-LPV, MRAC- $H_\infty$ GS-LPV,etc) do not need and FDI block because the MRAC has the ability to detect and start to accommodate the fault by itself and third the scheme proposed by Montes de Oca et al. (2009) does not combine any technique as the ones proposed in this thesis.

**Table 25. Summary of comparison with similar approaches.**

Author	Methodology	Application	Advantages	Disadvantages	Differences
Cho et al. (1990)	Pole assignment + MRAC	Simple 2 <sup>nd</sup> order plant	Ensures the performance of set-point tracking	The failure is only in one of the two controllers.	Do not use any AI method.
Hsu (1990)	VS-MRAC	Plants with arbitrary relative degree	Nicer transient behavior, disturbance rejection capability, robustness	Only additive changes were proved.	The VS is based on switching functions that creates a sliding surface.
Hsu et al. (1994)	VS-MRAC	Plants with arbitrary relative degree	High-gain stability is achieved	High frequency noise must be compensated.	The VS is based on switching functions that creates a sliding. Surface
Jia & Jingping (1997)	MRAC + PID optimized by GA + ANN	CSTR system	Satisfying Control Effects	Long Computational Time due to the use of the GA.	The ANN was used as an estimator of the plant. The GA trained the ANN.
Ahmed (2000)	ANN + MRAC	Nonlinear Plants	Good tracking performance + fast	Only local stability was proved.	The ANN is used to approximate the

			convergence		controller parameters.
Zhu et al. (2000)	MRAC + aged actuator	Reactor Regulating System (Nuclear Plant)	Successful dealing against the worst tested situation	Limited simulation results. Lack of formal stability analysis.	Do not use any combination with the MRAC.
Miyasato (2006)	MRAC + $H_\infty$ + LPV	Continuous System	The adaptive control systems are uniformly bounded, and that the tracking error converges to an arbitrary small residual region.	Only mathematical procedure was showed (no simulations or experiments yet).	The $H_\infty$ is not a Gain Scheduling Controller.
Thanapalan et al. (2006)	MRAC + Quaternion based adaptive attitude control	Satellite Formation Flying	Increase operational safety	The limitations of system reconfiguration due to signal constraints are not established yet.	The MRAC is not combined with any proposed scheme of this thesis.
Yan et al. (2006)	Output Feedback VS-MRAC	Second order systems	Guarantee a pre-specified steady-state and transient performance specifications for tracking error	Amplitude of the ultimate switching control remains constant and can be eventually large / Unmodeled dynamic are not tested	Based on switching schemes.
Zhang & Li (2006)	PID-MRAC + RBF ANN	Second order system	High control accuracy and good dynamic performance	The stability analysis was not mention at all.	The RBF-ANN is used to identify the system.
Abdullah & Zribi (2008)	Model Reference Control	Coupled-Tank System	Simulation and experimental results indicate that the proposed scheme works well.	Only changes in reference models are tested.	Uses SVD and a state feedback controller gain.
Bayati (2008)	MRAC + PID-GA	Power System	Oscillations Mitigation	The stability analysis was not mention at all.	Different MRAC combinations except MRAC + PID.
Hongjie & Bo (2008)	MRAC + ANN + Proportional Controller	Servo System	Reduction of plant sensitivity to parameter variation and disturbance	The ANN controller is used in one part of the system.	The ANN structure inside the MRAC. The use of only a Proportional controller.
Demetriou et al. (2009)	MRAC	Positive real infinite dimensional	Well-posedness and stability analysis	Only additive faults were tested.	The MRAC is not combined with any

		system			proposed scheme of this thesis.
Montes de Oca et al. (2009)	AMM	Two-degree of Freedom Helicopter	Allow the controller design to be defined by a set of admissible faults.	The active controller requires the fault estimation.	Completely different structure - Need a FDI block.
Sang & Tao (2009)	MRAC	SISO and Multivariable MRAC systems	Robustness of the desired closed-loop performance of stability and asymptotic tracking	In SISO MRAC the performance is not achieved for large reduction in actuator effectiveness.	The MRAC is not combined with any proposed scheme of this thesis.
Tan et al. (2011)	Discrete-MRAC	Aircraft flight control	Faster and accurate compensation of failure uncertainties	Continuous-time system with actuator failures are not presented yet (under study).	The MRAC is not combined with other methods.



# **CHAPTER 5**

## **STABILITY**

### **ANALYSIS**

## 5 Stability Analysis

### 5.1 LPV Stability Analysis

To analyze the stability of the LPV systems the Quadratic Lyapunov Stability method is used (Gahinet et al., 1994). This method helps to analyze linear time-varying systems, such has:

$$E(t)\dot{x}(t) = A(t)x(t), \quad x(0) = x_0 \quad (155)$$

To ensure the stability of the above system, a sufficient condition for asymptotic stability is the existence of a positive-definite quadratic Lyapunov function:

$$V(x) = x^T P x \quad (156)$$

and

$$dV(x(t))/dt < 0 \quad (157)$$

throughout the state trajectories. This means that:

$$Q := P^{-1} \quad (158)$$

$$A(t)QE(t)^T + E(t)QA(t)^T < 0 \quad (159)$$

for all the times  $t$ .

Evaluating the quadratic stability is not tractable in general since equation (159) places an infinite number of constraints on  $Q$ . Therefore, equation (159) is reduced to a finite set of LMI constraints in the next cases:

1.  $A(t)$  and  $E(t)$  are fixed affine functions of time-varying parameters  $p_1(t), \dots, p_n(t)$ .

$$A(t) = A_0 + p_1(t)A_1 + \dots + p_n(t)A_n \quad (160)$$

$$E(t) = E_0 + p_1(t)E_1 + \dots + p_n(t)E_n \quad (161)$$

Equation (160) and (161) represent an affine parameter-dependent model.

2.  $A(t) + jE(t)$  ranges in a fixed polytope of matrices

$$A(t) = \alpha_1(t)A_1 + \dots + \alpha_n(t)A_n \quad (162)$$

$$E(t) = \alpha_1(t)E_1 + \dots + \alpha_n(t)E_n \quad (163)$$

with  $\alpha_i(t) \geq 0$  and  $\sum_{i=1}^n \alpha_i(t) = 1$ . The above is referred as a polytopic model.

The first case is for those system in which the state-space equations depend affinely on time-varying parameters, and the second case is for time-varying systems modeled by an envelope of LTI systems. In conclusion, a Quadratic Lyapunov function will guarantee stability for arbitrarily fast time variations.

The LMI conditions for quadratic stability are (Barmish, 1983; Boyd & Yang, 1989; Horisberger & Belanger, 1976):

For affine models, consider the parameter-dependent model

$$\begin{aligned} E(\mathbf{p})\dot{\mathbf{x}} &= A(\mathbf{p})\mathbf{x} \\ A(\mathbf{t}) &= A_0 + \mathbf{p}_1(\mathbf{t})A_1 + \dots + \mathbf{p}_n(\mathbf{t})A_n \\ E(\mathbf{t}) &= E_0 + \mathbf{p}_1(\mathbf{t})E_1 + \dots + \mathbf{p}_n(\mathbf{t})E_n \end{aligned} \quad (164)$$

where

$$\mathbf{p}_i(\mathbf{t}) \in [\underline{\mathbf{p}}_i, \bar{\mathbf{p}}_i] \quad (165)$$

and

$$V_s = \left\{ (\omega_1, \dots, \omega_n) : \omega_i \in [\underline{\mathbf{p}}_i, \bar{\mathbf{p}}_i] \right\} \quad (166)$$

represent the set of corners of the parameter box. The dynamical system (equation 164) is quadratically stable if there exist symmetric matrices  $Q$  and  $\{M_i\}_{i=1}^n$  in such a form that

$$A(\omega)QE(\omega)^T + E(\omega)QA(\omega)^T + \sum_i \omega_i^2 < 0 \text{ for all } \omega \in V_s \quad (167)$$

$$A_iPE_i^T + E_iPA_i^T + M_i \geq 0 \text{ for } i=1, \dots, n \quad (168)$$

$$M_i \geq 0 \quad (169)$$

$$Q > I \quad (170)$$

On the other hand, for polytopic models, the polytopic system:

$$\begin{aligned} E(\mathbf{t})\dot{\mathbf{x}}(\mathbf{t}) &= A(\mathbf{t})\mathbf{x}(\mathbf{t}) \\ A(\mathbf{t}) + jE(\mathbf{t}) &\in \text{Co}\{A_1 + jE_1, \dots, A_n + jE_n\} \end{aligned} \quad (171)$$

is quadratically stable if there exist a symmetric matrix  $Q$  and scalars  $t_{ij} = t_{ji}$  such that

$$A_iQE_j^T + E_jQA_i^T + A_jQE_i^T + E_iQA_j^T < 2t_{ij}I \text{ for } i, j \in \{1, \dots, n\} \quad (172)$$

$$Q > I \quad (173)$$

$$\begin{bmatrix} t_{11} & \dots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{1n} & \dots & t_{nn} \end{bmatrix} < 0 \quad (174)$$

All the above LMI conditions are necessary and sufficient for quadratic stability when: In the affine case, no parameter  $p_i$  enters both  $A(\mathbf{t})$  and  $E(\mathbf{t})$ , which means,  $A_i=0$  or  $E_i=0$  for all  $i$ . This implies that equations (168) and (169) can be suppressed and is enough to verify equation (159) at the corners  $\omega$  of the parameter box. On the other hand, for the polytopic case, either  $A(\mathbf{t})$  or  $E(\mathbf{t})$  is constant and is enough to compute equations (165) and (166) for  $t_{ij} = 0$ .

### 5.1.1 LPV Plant Open Loop Stability Analysis

Consider the following plant:

$$\frac{0.108158\varphi_1}{s^2+0.5085(\varphi_1+\varphi_2)s+0.258572(\varphi_1\varphi_2)} \quad (175)$$

where:

$$0.1 < \varphi_1 < 0.6 \quad \text{and} \quad 0.1 < \varphi_2 < 0.6$$

The stability will be measured using the resultant plants of the above system: 4 plants using the extremes values of  $\varphi_1$  and  $\varphi_2$  were obtained (see Table 25).

**Table 26. Resulting plants at the corners of the parameter box.**

Plants	Laplace Transfer Function	State Space Transfer Function
<b>Plant 1</b> ( $\varphi_1=0.1$ and $\varphi_2=0.1$ ):	$G_1 = \frac{0.01058}{s^2+0.1017s+0.002586}$	$A = \begin{bmatrix} -0.05085 & 0 \\ 0.05085 & -0.05085 \end{bmatrix}$ $B = \begin{bmatrix} 0.2127 \\ 0 \end{bmatrix}$ $C = [0 \quad 1]$ $D = [0]$
<b>Plant 2</b> ( $\varphi_1=0.1$ and $\varphi_2=0.6$ ):	$G_2 = \frac{0.01082}{s^2+0.3559s+0.01551}$	$A = \begin{bmatrix} -0.05085 & 0 \\ 0.05085 & -0.3051 \end{bmatrix}$ $B = \begin{bmatrix} 0.2127 \\ 0 \end{bmatrix}$ $C = [0 \quad 1]$ $D = [0]$
<b>Plant 3</b> ( $\varphi_1=0.6$ and $\varphi_2=0.1$ ):	$G_3 = \frac{0.06489}{s^2+0.3559s+0.01551}$	$A = \begin{bmatrix} -0.3051 & 0 \\ 0.3051 & -0.05085 \end{bmatrix}$ $B = \begin{bmatrix} 0.2127 \\ 0 \end{bmatrix}$ $C = [0 \quad 1]$ $D = [0]$
<b>Plant 4</b> ( $\varphi_1=0.6$ and $\varphi_2=0.6$ ):	$G_4 = \frac{0.06489}{s^2+0.6102s+0.09309}$	$A = \begin{bmatrix} -0.3051 & 0 \\ 0.3051 & -0.3051 \end{bmatrix}$ $B = \begin{bmatrix} 0.2127 \\ 0 \end{bmatrix}$ $C = [0 \quad 1]$ $D = [0]$

To calculate the quadratic stability of the above plants, the following Matlab® code was used.

- First, the four state space matrices A are defined:

```
>> A1=[-0.05085,0;0.05085,-0.05085];
```

```
>> A2=[-0.05085,0;0.05085,-0.3051];
```

```
>> A3=[-0.3051,0;0.3051,-0.05085];
```

```
>> A4=[-0.3051,0;0.3051,-0.3051];
```

- Second, the polytopic system is defined as:

```
>> s1 = ltisys(A1)
```

$$s1 = \begin{bmatrix} -0.05085 & 0 & 2 \\ 0.05085 & -0.05085 & 0 \\ 0 & 0 & \infty \end{bmatrix}$$

```
>> s2 = ltisys(A2)
```

$$s2 = \begin{bmatrix} -0.05085 & 0 & 2 \\ 0.05085 & -0.3051 & 0 \\ 0 & 0 & \infty \end{bmatrix}$$

>> s3 = ltisys(A3)

$$s3 = \begin{bmatrix} -0.3051 & 0 & 2 \\ 0.3051 & -0.05085 & 0 \\ 0 & 0 & \infty \end{bmatrix}$$

>> s4 = ltisys(A4)

$$s4 = \begin{bmatrix} -0.3051 & 0 & 2 \\ 0.3051 & -0.3051 & 0 \\ 0 & 0 & \infty \end{bmatrix}$$

>> polsys = psys([s1 s2 s3 s4])

$$\begin{bmatrix} -\infty & 0 & -0.05085 & 0 & 2 & 0 & -0.05085 & 0 & 2 & 0 & -0.3051 & 0 & 2 & 0 & -0.3051 & 0 & 2 \\ 1 & 0 & 0.05085 & -0.05085 & 0 & 0 & 0.05085 & -0.3051 & 0 & 0 & 0.3051 & -0.05085 & 0 & 0 & 0.3051 & -0.3051 & 0 \\ 4 & 0 & 0 & 0 & -\infty & 0 & 0 & 0 & -\infty & 0 & 0 & 0 & -\infty & 0 & 0 & 0 & -\infty \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Third, the quadratic stability is calculated using:

>> [tmin,P] = quadstab(polsys)

Solver for LMI feasibility problems  $L(x) < R(x)$

This solver minimizes  $t$  subject to  $L(x) < R(x) + t*I$

The best value of  $t$  should be negative for feasibility

Iteration : Best value of  $t$  so far

1 0.045331

2 -0.012013

Result: best value of  $t$ : -0.012013

f-radius saturation: 0.000% of  $R = 1.00e+008$

This system is quadratically stable

tmin = -0.012013412684107

The value of the matrix  $P$  is

$$P = \begin{bmatrix} 0.95945 & 0.04672 \\ 0.04672 & 0.62847 \end{bmatrix}$$

Denoting the LMI system represented by equations (173) and (174) by  $A(x) < 0$ , the function quadstab evaluates its feasibility by minimizing  $\tau$  subject to  $A(x) < \tau$  and returns the global minimum “tmin” of this problem. Therefore the system is quadratically stable if and only if  $tmin < 0$ .

### 5.1.2 Maximizing the Quadratic Stability Region of the LVP Plant

In order to maximize the stability region of the LPV plant, the following theory is used:

- First E matrix need to be constant and each parameter  $p_i(t)$  ranges in an interval  $p_i(t) \in [\underline{p}_i, \bar{p}_i]$ .
- Second, the center and radius of each interval is represented by  $u_i = \frac{1}{2}(\underline{p}_i + \bar{p}_i)$  and  $\delta_i = \frac{1}{2}(\bar{p}_i - \underline{p}_i)$ .
- Third, the maximal quadratic stability region is defined as the largest portion of the parameter box in which the quadratic stability can be established. The above means the largest dilatation factor such that the system is quadratically stable whenever  $p_i(t) \in [u_i - \theta\delta_i, u_i + \theta\delta_i]$ .

The computation of the above theory is the following. The LPV system is transformed in an LTI form:

```
>> s0=ltisys([0 0;0 0],[0.2127;0],[0 1], [0])
```

$$s_0 = \begin{bmatrix} 0 & 0 & 0.2127 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\infty \end{bmatrix}$$

```
>> s1=ltisys([-0.5085 0;0.5085 0],[0;0],[0 0], [0],0) %Phi1 component
```

$$s_1 = \begin{bmatrix} -0.5085 - 1i & 0 & 0 & 2 \\ 0.5085 & 0 - 1i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\infty \end{bmatrix}$$

```
>> s2=ltisys([0 0;0 -0.5085],[0;0],[0 0], [0],0) %Phi2 component
```

$$s_2 = \begin{bmatrix} 0 - 1i & 0 & 0 & 2 \\ 0 & -0.5085 - 1i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\infty \end{bmatrix}$$

```
>> Phi1min=.1; Phi1max=.6;
```

```
>> Phi2min=.1; Phi2max=.6;
```

```
>> pv=pvec('box', [Phi1min Phi1max; Phi2min Phi2max])
```

$$pv = \begin{bmatrix} 1 & 0.1 & 0.6 & 0 & 0 \\ 2 & 0.1 & 0.6 & 0 & 0 \end{bmatrix}$$

```
>> affsys = psys(pv,[s0 s1 s2])
```

$$\begin{bmatrix} -\infty & 0 & 0 & 0 & 0.2127 & 2 & 0 & -0.5085-1i & 0 & 0 & 2 & 0 & 0 & -1i & 0 & 0 & 2 & 1 & 0.1 & 0.6 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0505 & 0 & -1i & 0 & 0 & 0 & 0 & -0.5085-1i & 0 & 0 & 2 & 0.1 & 0.6 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & -\infty & 0 & 0 & 0 & 0 & -\infty & 0 & 0 & 0 & 0 & 0 & -\infty & 0 & 0 & 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ -10 & 0 \end{bmatrix}$$

To calculate the maximal stability margin of the parameter box, the next command is used:

```
>> [marg,P] = quadstab(affsys,[1 0 0])
```

Solver for generalized eigenvalue minimization

Iterations : Best objective value so far

1	
2	
3	
4	
5	309.375000
6	146.808105
7	101.297593
***	new lower bound: -4926.595947
8	22.509389
***	new lower bound: -1117.669819
9	4.647148
10	3.206532
***	new lower bound: -556.511336
11	2.212507
***	new lower bound: -66.758202
12	2.190382
***	new lower bound: -32.272847
13	2.168478
14	1.614029
***	new lower bound: -2.015697
15	1.597889
16	1.259115
***	new lower bound: -0.161549
17	1.224548
18	1.147979
***	new lower bound: -0.119306
19	1.128178
20	0.885332
***	new lower bound: 0.516723
21	0.885332
22	0.816218
23	0.755383
***	new lower bound: 0.524422
24	0.755383

25	0.740948
***	new lower bound: 0.643162
26	0.739420
27	0.733404
***	new lower bound: 0.692649
28	0.732767
29	0.726499
30	0.724912
***	new lower bound: 0.693690
31	0.724912
32	0.724424
33	0.721063
***	new lower bound: 0.709444
34	0.720699
***	new lower bound: 0.709807

Result: feasible solution

best value of t: 0.720699

guaranteed absolute accuracy: 1.09e-002

f-radius saturation: 0.000% of R = 1.00e+008

Termination due to SLOW PROGRESS:

the gen. eigenvalue t decreased by less than  
1.000% during the last 5 iterations.

Quadratic stability established on 138.7541% of the prescribed parameter box  
marg = 1.3875

$$P = \begin{bmatrix} 663.0128 & 22.0194 \\ 22.0194 & 42.7137 \end{bmatrix}$$

The above means that the system is quadratically stable on 138.7541% of the parameter box. If  $\text{marg} \geq 1$  the system is quadratically stable in the entire specified parameter box.



## 5.2 Stability Assumptions for the MRAC controller

The objective of the Model Reference Adaptive Controller is achieved if the input of the plant “u” is selected so that the closed-loop transfer function from the reference “r” to the plant output “y<sub>p</sub>” has stable poles and is equal to the transfer function of the reference model. When the above is accomplished the plant output matches the output of the reference model exponentially fast.

$$G_r = \frac{CG}{1+CG} \quad (176)$$

- The plant model is strictly proper and minimum phase (i.e. have stable zeros).
- The design of the controller requires the knowledge of the coefficients of the plant transfer function  $G(s)$ .
- The reference model has the same degree as the corresponding plant polynomial.
- The reference model is stable and minimum phase.
- The input signal of the controller “u<sub>c</sub>” must be persistently exciting to achieve the desired value of the system (reference model).

If the plant is represented by:

$$\frac{Y(s)}{U(s)} = K_p \frac{B_p(s)}{A_p(s)} \quad (177)$$

- $A_p$  and  $B_p$  are coprime Hurwitz polynomials.
- The roots of  $B_p$  are the open left-half s-plane.
- The sign of  $K_p$  is known and assumed to be positive.
- The reference model is a strictly positive real transfer function.
- $A_r$  and  $B_r$  ( $A$  and  $B$  from the reference model) are a controllable pair.
- The reference model is monic and stable.
- The reference model is Hurwitz polynomial (i.e. its zeros have strictly negative real parts).
- The persistency excitation (PE) of the input signals in the adaptive loop guarantees the exponential stability of the unperturbed error system and eventually the local stability of the closed-loop time variant plant.

### 5.2.1 Lyapunov Stability Theorem for the Design of the MRAC Controller

As mentioned in Chapter 2, the Lyapunov theory in the design of an MRAC controller was introduced because the MIT rule does not guarantee the stability of the closed-loop system. To design an MRAC controller using Lyapunov theory, the first step is to derive a differential equation for the error that contains the adaptation parameters. Then, a Lyapunov function and an adaptation mechanism need to be established to reduce the error to zero. The Lyapunov derivative function  $dV/dt$  is usually negative

semidefinite. Therefore, to determine the parameter convergence is necessary to establish persistently excitation and uniform observability on the system and the reference signal (Astrom & Wittenmark, 1995).

The Lyapunov stability theorem establishes the following: If there exists a function  $V: \mathbb{R}^n \rightarrow \mathbb{R}$  being positive definite and its derivative:

$$dV/dt = \partial V^T / \partial x \, dx/dt = \partial V^T / \partial x \, f(x) = -W(x) \quad (178)$$

is negative semidefinite, then the solution  $x(t) \rightarrow 0$  to

$$dx/dt = f(x) \quad f(0) = 0 \quad (179)$$

is stable. If  $dV/dt$  is negative definite the solution will be asymptotically stable.  $V$  denotes the Lyapunov function for the system. If:

$$dV/dt < 0 \quad \text{and} \quad V(x) \rightarrow \infty \quad \text{when} \quad \|x\| \rightarrow \infty \quad (180)$$

the solution is globally asymptotically stable.

Therefore, the following procedure was realized:

Process model:

$$\ddot{y}_{process} + a_1 \dot{y}_{process} + a_0 y_{process} = bu \quad (181)$$

Reference model:

$$\ddot{y}_{reference} + a_{1r} \dot{y}_{reference} + a_{0r} y_{reference} = b_r u_c \quad (182)$$

Control law:

$$u = \theta_1 u_c - \theta_2 y_{process} \quad (183)$$

Error:

$$e = y_{process} - y_{reference} \quad (184)$$

Then, the error dynamics is represented by:

$$(185)$$

$$\dot{e} = \dot{y}_{process} - \dot{y}_{reference} = \frac{1}{a_1} [bu - \ddot{y}_{process} - a_0 y_{process}] - \frac{1}{a_{1r}} [b_r u_c - \ddot{y}_{reference} - a_{0r} y_{reference}]$$

To simplify the mathematical notation  $y_{reference} = y_r$  and  $y_{process} = y_p$ .

$$\dot{e} = \dot{y}_p - \dot{y}_r = \frac{1}{a_1} [bu - \ddot{y}_p - a_0 y_p] - \frac{1}{a_{1r}} [b_r u_c - \ddot{y}_r - a_{0r} y_r] \quad (186)$$

Substituting  $y_r = y_p - e$  and  $\ddot{y}_r = \ddot{y}_p - \ddot{e}$  from equation 22, equation 23 is obtained:

$$\dot{e} = \frac{1}{a_1} bu - \frac{1}{a_1} \ddot{y}_p - \frac{a_0}{a_1} y_p - \frac{1}{a_{1r}} b_r u_c + \frac{1}{a_{1r}} \ddot{y}_p - \frac{1}{a_{1r}} \ddot{e} + \frac{a_{0r}}{a_{1r}} y_p - \frac{a_{0r}}{a_{1r}} e \quad (187)$$

Replacing  $u = \theta_1 u_c - \theta_2 y_p$  in the above equation and placing the error terms in the left side of the equation, equation 188 is obtained:

$$\frac{1}{a_{1r}} \ddot{e} + \dot{e} + \frac{a_{0r}}{a_{1r}} e = \frac{1}{a_1} b \theta_1 u_c - \frac{1}{a_{1r}} b_r u_c - \frac{1}{a_1} b \theta_2 y_p - \frac{a_0}{a_1} y_p + \frac{a_{0r}}{a_{1r}} y_p - \frac{1}{a_1} \ddot{y}_p + \frac{1}{a_{1r}} \ddot{y}_p \quad (188)$$

The control objective is that the Process Model must be equal to the Reference Model ( $a_1 = a_{1r}$ ,  $a_0 = a_{0r}$ , and  $b = b_r$ ), then:

$$\frac{1}{a_{1r}} \ddot{e} + \dot{e} + \frac{a_{0r}}{a_{1r}} e = \frac{1}{a_{1r}} b_r \theta_1 u_c - \frac{1}{a_{1r}} b_r u_c - \frac{1}{a_{1r}} b_r \theta_2 y_p - \frac{a_{0r}}{a_{1r}} y_p + \frac{a_{0r}}{a_{1r}} y_p - \frac{1}{a_{1r}} \ddot{y}_p + \frac{1}{a_{1r}} \ddot{y}_p \quad (189)$$

$$\frac{1}{a_{1r}}\ddot{e} + \dot{e} + \frac{a_{0r}}{a_{1r}}e = \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p \quad (190)$$

$$\frac{1}{a_{1r}}\frac{d^2e}{dt^2} + \frac{de}{dt} + \frac{a_{0r}}{a_{1r}}e = \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p \quad (191)$$

$$\frac{de}{dt} = -\frac{1}{a_{1r}}\frac{d^2e}{dt^2} - \frac{a_{0r}}{a_{1r}}e + \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p \quad (192)$$

The proposed Lyapunov function is quadratic in tracking error and controller parameter estimation error since it is expected that the adaptation mechanism will drive both types of errors to zero (tracking error and error in the controller parameters estimation). From the equation error dynamics (see equation 192) the proposed Lyapunov function is:

$$V(e, \theta_1, \theta_2) = \frac{1}{2}\left(a_{1r}e^2 + \frac{1}{\gamma b_r}(b_r\theta_1 - b_r)^2 + \frac{1}{\gamma b_r}(b_r\theta_2)^2\right) \quad (193)$$

where  $b_r, \gamma$  and  $a_{1r} > 0$ .

Equation 193 will be zero when the error is zero and the controller parameters are equal to the desired values. The above Lyapunov function is valid if the derivative of this function is negative. Thus, the derivative of equation 194 is:

$$\dot{V} = a_{1r}e\frac{de}{dt} + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (194)$$

Substituting equation 192 in the above equation, and rearranging the similar terms, equation 195 is obtained.

$$(195)$$

$$\dot{V} = a_{1r}e\left(-\frac{1}{a_{1r}}\frac{d^2e}{dt^2} - \frac{a_{0r}}{a_{1r}}e + \frac{1}{a_{1r}}(b_r\theta_1 - b_r)u_c - \frac{1}{a_{1r}}(b_r\theta_2)y_p\right) + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (196)$$

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 + (b_r\theta_1 - b_r)u_c e - (b_r\theta_2)y_p e + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (196)$$

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 + (b_r\theta_1 - b_r)u_c e + \frac{1}{\gamma}(b_r\theta_1 - b_r)\frac{d\theta_1}{dt} - (b_r\theta_2)y_p e + \frac{1}{\gamma}(b_r\theta_2)\frac{d\theta_2}{dt} \quad (197)$$

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 + \frac{1}{\gamma}(b_r\theta_1 - b_r)\left(\frac{d\theta_1}{dt} + \gamma u_c e\right) + \frac{1}{\gamma}(b_r\theta_2)\left(\frac{d\theta_2}{dt} - \gamma y_p e\right) \quad (198)$$

If the adaptation parameters are updated as:

$$\frac{d\theta_1}{dt} = -\gamma u_c e \quad (199)$$

$$\frac{d\theta_2}{dt} = \gamma y_p e \quad (200)$$

Then

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 \quad (201)$$

It can be seen that equation 37 is negative semidefinite which implies  $V(t) \leq V(0)$ . This ensures that  $e$ ,  $\theta_1$  and  $\theta_2$  are bounded. Since  $a_{1r} > 0$ ,  $a_{0r} > 0$  and  $u_c$  is bounded then  $y_r$  is bounded and therefore  $y_p = e + y_r$  is bounded as well. From the boundedness and convergence set theorem it can be concluded that the error  $e$  will go to zero (Astrom & Wittenmark, 1995). Since  $V(e, \theta_1, \theta_2)$  is positive definite and

$$\dot{V} = -e\frac{d^2e}{dt^2} - a_{0r}e^2 < 0 \quad (202)$$

Then according to the theorem of Lyapunov uniform asymptotic stability of non-autonomous systems, the equilibrium point  $x = 0$  is uniformly stable.

**Theorem of Lyapunov uniform asymptotic stability of non-autonomous systems**

Let  $x = 0$  be an equilibrium point of a system described by  $\dot{x} = f(x, t)$  and  $u \subset R^n$  a domain containing it. Let  $V: u \times [0, \infty) \rightarrow R$  be a continuously differentiable function that satisfies

$$\omega_1(x) \leq V(x, t) \leq \omega_2(x) \quad (203)$$

$$\dot{V}(x, t) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x, t) \leq -\omega_3 \quad (204)$$

For all  $t \geq t_0$  and  $x \in u$ , where  $\omega_1(x)$ ,  $\omega_2(x)$  and  $\omega_3(x)$  are continuous positive definite functions of  $u$ .

Then  $x = 0$  is uniformly asymptotically stable and  $V$  is called a Lyapunov function. Furthermore, if  $\omega_3(x) = 0$ , then  $x = 0$  is uniformly stable.

*Corollary:*

Suppose that the assumptions of the above theorem hold for all  $x \in R^n$  and  $\omega_1(x) \rightarrow \infty$  for  $\|x\| \rightarrow \infty$ , then  $x = 0$  is globally uniformly asymptotically stable (Astrom & Wittenmark, 1995).

**Theorem of boundedness and convergence set**

Let  $D = \{x \in u \mid \|x\| < \gamma\}$  and suppose that  $f(x, t)$  is locally Lipschitz on  $D \times [0, \infty)$ . Let  $V$  a continuously differentiable function such that

$$\alpha_1(\|x\|) \leq V(x, t) \leq \alpha_2(\|x\|) \quad (205)$$

And

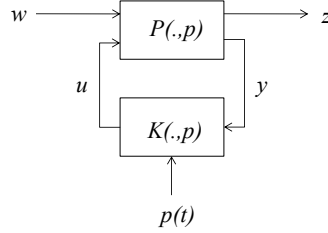
$$\frac{dv}{dt} = \frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} f(x, t) \leq -\omega(x) \leq 0 \quad (206)$$

$\forall t \geq 0, \forall x \in D$ , where  $\alpha_1$  and  $\alpha_2$  are class  $K$  functions defined on  $[0, \infty)$  and  $\omega(x)$  is continuously on  $D$ .

Furthermore, it is assumed that  $\frac{dv}{dt}$  is uniformly continuous in  $t$ . Then, all solutions to  $\frac{dx}{dt} = f(x, t)$  with  $\|x(t_0) \leq \alpha_2^{-1}(\alpha_1(\gamma))\|$  are bounded and satisfy  $\omega(x(t)) \rightarrow 0$  as  $t \rightarrow \infty$  (Astrom & Wittenmark, 1995).

### 5.3 Stability Assumptions for $H_\infty$ Gain scheduling controller

As mentioned in Chapter 4, the designed gain scheduled controller  $K(.,p)$  (see Figure 109) must satisfy the vertex property and the closed-loop system should be stable for all admissible parameters trajectories  $p(t)$  (Apkarian & Gahinet, 1995; Apkarian et al., 1995, Becker & Packard, 1994; Packard, 1994).



**Figure 109.  $H_\infty$  Gain Scheduling Controller representation.**

Where  $K(.,p)$  represents the  $H_\infty$  Gain Scheduling Controller,  $P(.,p)$  is the parameter dependent plant, and  $p(t)$  represents a time-varying vector of physical parameters.

The above is implicit in the design of the  $H_\infty$  Gain Scheduling Controller, because first to enforce the performance and robustness requirements, the following loop-shaping criterion was used:

$$\left\| \frac{W_1 S}{W_2 K S} \right\|_\infty < 1 \quad (207)$$

Where  $S = (I + GK)^{-1}$  and the  $H_\infty$  norm is in terms of input/output Random Mean Square gain which is the largest input/output gain over all bounded inputs  $u(t)$  (Willems, 1971; Anderson & Vongpanitler, 1973; Vidyasagar, 1992). Then, for loop-shaping purposes an augmented plant associated with the above criterion was created using the following commands (see section 4.5):

```
[pdP,r]=sconnect('r','y=K','K:r-y','G:K',pdG);
Paug=smult(pdP,sdiag(w1,w2));
```

Where  $pdP$  and  $Paug$  represent polytopic models. Therefore, the  $H_\infty$  Gain Scheduling Controller was computed with the next command:

```
[gopt,pdK]=hinfgs(Paug,r,1e-2,1e-4);
```

The above computation gave a value of  $\gamma_{opt} < 1$  ( $gopt=0.0024$ ), which means that the performance specification are achievable, and  $pdK$  returns the polytopic description of the controller with the  $\gamma_{opt} < 1$  performance. Therefore, the closed-loop system of the designed gain scheduled controller is stable for all the admissible parameters trajectories  $p(t)$ .

The stability analysis of the whole control system (MRAC- $H_\infty$ GS-LPV) could not be computed due to the complexity of the closed loop system. For this reason, the stability was proved by design, which means that the proposed controller is integrated by an MRAC based on Lyapunov theory and a  $H_\infty$  Gain Scheduling Controller that are already stable.

# **CHAPTER 6**

## **CONCLUSIONS**

## 6 Conclusions

### 6.1 Industrial Heat Exchanger

#### 6.1.1 Comparison between MRAC-ANN-PID, MRAC- $H_\infty$ , MRAC-SMC, MRAC-ANN, MRAC-PID and MRAC

In general, the MRAC-ANN-PID methodology showed the best results because it was robust against sensor and imperceptible against actuator faults of 5%, 15% and 25% with a very low error between the reference model and the process model. This method was the best scheme because is a combination of two types of controllers: the PID controller optimized by the GA with the best parameters to handle the fault, and the ANN that was trained to follow the desired system trajectory no matter the fault size. In addition, the MRAC- $H_\infty$  approach was robust against certain types of faults (5% and 15% of abrupt and gradual sensor faults) and it was fault tolerant to the rest of fault types (25% of abrupt and gradual sensor faults and 5%, 15% and 25% of abrupt and gradual actuator faults). Also, the MRAC-SMC resulted in a good FTC scheme because it was robust against sensor faults (5%, 15% and 25%) because has the lower MSE, but it was fault tolerant for some actuator faults (5% and 15%) and was degraded for actuators faults of 25%. The other three approaches (MRAC-ANN, MRAC-PID and MRAC) were degraded for abrupt and gradual actuator faults, and just one of them (MRAC-ANN) was robust against abrupt and gradual sensor faults. Also, the manipulated variable was plotted and in this figures it can be observed how the system compensate the fault.

### 6.2 Coupled-Tank System

#### 6.2.1 Comparison between MRAC-ANN4OP-LPV, MRAC-4OP-LPV and MRAC- $H_\infty$ 4OP-LPV based on the MIT rule and on Lyapunov theory

In the evaluation of the first operating point  $\varphi_1=0.1$  and  $\varphi_2=0.1$ , when the abrupt sensor fault of 23.3% was applied the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory was robust, the MRAC- $H_\infty$ 4OP-LPV and MRAC-4OP-LPV based on MIT rule and based on Lyapunov theory resulted to be fault tolerant (the fault was corrected immediately). In addition when the abrupt actuator fault of 1% was applied the MRAC-ANN4OP-LPV based on the MIT rule and on Lyapunov theory could not accommodate the faults (the system remained oscillating), the MRAC- $H_\infty$ 4OP-LPV based on Lyapunov theory and on MIT rule was fault tolerant (the fault was corrected in less than 2500 seconds). And also, the MRAC-4OP-LPV based on MIT rule was fault tolerant, the fault was corrected after 15000 second in the scheme based on the MIT rule and 12500 second in the scheme based on Lyapunov theory.

In the evaluation of the third operating point  $\varphi_1=0.6$  and  $\varphi_2=0.1$ , when the gradual sensor fault of 10% was applied the MRAC-ANN4OP-LPV based on the MIT rule and based on Lyapunov theory was robust, the MRAC- $H_\infty$ 4OP-LPV based on MIT rule and on Lyapunov theory was fault tolerant (the fault was accommodated immediately) and the MRAC-4OP-LPV based on MIT rule and based on Lyapunov theory

resulted to be degraded. In addition when the gradual actuator fault of 1% was applied the MRAC- $H_\infty$ 4OP-LPV based on MIT rule and on Lyapunov theory was fault tolerant (the fault was accommodated in less than 4000 seconds), the MRAC-ANN4OP-LPV based on the MIT rule and on Lyapunov theory could not accommodate the fault and the MRAC-4OP-LPV based on MIT rule could not accommodate the fault (the system remained oscillating) and the MRAC-4OP-LPV based on Lyapunov theory was fault tolerant (the fault was accommodated in 2000 seconds).

In general, for sensor faults the MRAC-ANN4OP-LPV methodology showed the best results because it was fault tolerant against the applied sensor faults no matter the value of the operating point between the operating ranges. This method resulted the best scheme because is a combination of two type of controllers, one is a Model Reference Adaptive Controller (MRAC) and the other one is an Artificial Neural Network designed to follow the ideal trajectory (non-faulty trajectory). Both controllers were designed to work in the 4 operating points of an LPV system giving them the possibility of control each of these operating points. On the other hand, for actuator faults the MRAC- $H_\infty$ 4OP-LPV was the best scheme because it was fault tolerant to the applied faults and also could accommodate the faults faster than the MRAC-4OP-LPV scheme.

## **6.2.2 Comparison between MRAC- $H_\infty$ GS-LPV and MRAC-LPV based on the MIT rule and on Lyapunov theory**

### **6.2.2.1 Additive Faults**

For the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$  with an additive abrupt sensor fault of 3.3% (5000 seconds) and an abrupt actuator fault of 20% (15000 seconds), the MRAC- $H_\infty$ GS-LPV based on MIT rule and on Lyapunov theory was robust to the sensor fault, was fault tolerant to the actuator fault and could tolerate the change in the operating point at 10000 seconds. On the other hand, the MRAC-LPV based on MIT rule and on Lyapunov theory was fault tolerant and could tolerate the change in the operating point at 10000 seconds. Then, if the additive abrupt sensor fault was changed to 166% (5000 seconds) and the abrupt actuator fault remains with 20% (15000 seconds) the MRAC- $H_\infty$ GS-LPV based on MIT and on Lyapunov theory was fault tolerant to the sensor and to the actuator fault and the MRAC-LPV based on MIT rule and on Lyapunov theory became degraded.

For the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.6$  with an additive gradual sensor fault of 3.3% (5000 seconds) and gradual actuator fault of 20% (15000 seconds) the MRAC- $H_\infty$ GS-LPV based on MIT rule and on Lyapunov theory was robust to the sensor fault, was fault tolerant to the actuator fault and could tolerate the change in the operating point at 10000 seconds. In addition, the MRAC-LPV based on MIT rule and on Lyapunov theory was fault tolerant and could tolerate the change in the operating point at 10000 seconds. Then, if the additive abrupt sensor fault was changed to 166% (5000 seconds) and the abrupt actuator fault remains with 20% (15000 seconds) the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV based on MIT rule and on Lyapunov theory were fault tolerant and could tolerate the change in the operating point at 10000 seconds.

In general the MRAC- $H_\infty$ GS-LPV showed better results because is a combination of two type of



LPV controllers, one is a Model Reference Adaptive Controller (MRAC) and the other one is a  $H_\infty$  Gain Scheduling Controller, both controllers were designed for an LPV system giving them the possibility of controlling any desired operating point between the operation range of the dependent variables ( $\varphi_1$  and  $\varphi_2$ ).

#### 6.2.2.2 Multiplicative Faults

It was observed that in the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$ , with multiplicative sensors faults of 100%, 90%, 50% and 5% at time 5000 seconds and multiplicative actuator faults of 100%, 90%, 50% and 5% at time 15000 seconds and a change in the operating point at time 10000 seconds, the MRAC- $H_\infty$ GS-LPV scheme was robust against the multiplicative sensor fault but the system output became degraded after the occurrence of the multiplicative actuator faults. On the other hand, the MRAC-LPV scheme was fault tolerant to multiplicative sensor faults and also the system output remained with an offset after the occurrence of the multiplicative actuator fault. The above applies for all the tested faults magnitudes and for the schemes based on the MIT rule and based on Lyapunov theory.

### 6.2.3 Comparison between the MRAC controllers using the nonlinear model of the system

#### 6.2.3.1 Additive Faults

The value of  $\gamma$  used was 0.003 because a smaller value of  $\gamma$  decreases the chattering in the system output (see Appendix C). For the operating point  $\varphi_1=0.3$  and  $\varphi_2=0.5$  with an additive abrupt sensor fault of 3.3% (5000 seconds) and an abrupt actuator fault of 20% (15000 seconds), the four different controllers could accommodate the fault and tolerate the change in the operating point. But in the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC- $H_\infty$ GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes.

Then, if the additive abrupt sensor fault was changed to a 166% (5000 seconds) and the abrupt actuator fault remains with a 20% (15000 seconds) the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV were unfeasible against the sensor fault because the tank level limit is 30 cm and these controllers were beyond this limit. On the other hand, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV based on Lyapunov theory were able to accommodate the sensor and the actuator faults. But the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory showed a faster adaptation performance in comparison with the MRAC-LPV.

For the operating point  $\varphi_1=0.6$  and  $\varphi_2=0.6$  with an additive gradual sensor fault of 3.3% (5000 seconds) and gradual actuator fault of 20% (15000 seconds) the four different controllers could accommodate the fault and tolerate the change in the operating point. But in schemes based on the MIT rule the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC- $H_\infty$ GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes. Then, if the additive gradual sensor fault was changed to 166% (5000 seconds) and the gradual actuator fault remains in 20% (15000 seconds) the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV were unfeasible against the sensor fault because the tank level limit is 30 cm and these controllers were beyond this limit. On the other hand, the MRAC- $H_\infty$ GS-LPV and the

MRAC-LPV based on Lyapunov theory were able to accommodate the sensor and the actuator faults. But the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory showed a faster adaptation performance in comparison with the MRAC-LPV.

In general, the MRAC- $H_\infty$ GS-LPV controller based on Lyapunov theory was the best scheme because was able to accommodate the fault in less time than the other schemes and also the fault impact was lower in comparison with the other schemes.

### 6.2.3.1 Multiplicative Faults

For the combination of multiplicative sensor fault of 100% at 5000 seconds and multiplicative actuator fault of 100% at 15000 seconds, it can be observe that the four controller became degraded after the occurrence of the multiplicative sensor fault and non of the controllers were able to return to the desired tank level. If the of multiplicative changed to a multiplicative sensor fault of 90% at 5000 seconds and multiplicative actuator fault of 90% at 15000 seconds, it can be observe that MRAC- $H_\infty$ GS-LPV based on the MIT rule and based on Lyapunov theory were able to accommodate both types of faults and could tolerate the change in the operating point. On the other hand, the MARC-LPV based on the MIT rule and on Lyapunov theory controllers were able to accommodate the multiplicative sensor fault but could not accommodate in time the multiplicative actuator fault. The  $H_\infty$ GS helps the MRAC to achieve a faster adaptation mechanism. If the multiplicative fault decreases to a 50%, it can be observed that the four different controllers could accommodate the faults (multiplicative sensor and multiplicative actuator) and tolerate the change in the operating point. But in the schemes based on the MIT, the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC- $H_\infty$ GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes. Finally, for the combination of multiplicative sensor fault of 5% at 5000 second and multiplicative actuator fault of 5% at 15000 seconds, it can be observed that the four different controllers could accommodate the faults (multiplicative sensor and multiplicative actuator) and tolerate the change in the operating point. But in the schemes based on the MIT, the impact of the fault is bigger than the schemes using Lyapunov theory. Also, the schemes using the MRAC- $H_\infty$ GS-LPV controller showed a faster adaptation than the MRAC-LPV schemes.

In general, the controllers presented in this research:

- Allow the system availability in spite of the presence of a fault, for example the MRAC-ANN-PID was robust against the tested sensor faults and the actuator faults were imperceptible due to the combination of an AI and an Adaptive control technique. And the MRAC- $H_\infty$ GS-LPV was robust and fault tolerant against sensor faults and fault tolerant for actuator faults.
- The implemented controllers in most cases were able to accommodate a fault between certain fault magnitude thresholds.
- The systems used to test the control schemes (Industrial Heat Exchanger and Coupled-Tank System) had stable plants because all the poles were in the left-half  $s$  plane.

- For the Industrial Heat Exchanger, the MRAC-PID-ANN methodology showed the best results because was robust against the tested sensor faults and the actuator faults were imperceptible with a very low error between the reference model and the process model; this method resulted the best scheme because is a combination of two type of controllers: the PID controller optimized by the GA with the best parameters to handle the fault, and the ANN that was trained to follow the desired system trajectory and its control structure adds robustness to the system.
- The computational time to train the ANN and to obtain the PID parameters using the GA was excessively long (48 hrs for the GA); this was the main reason for not incorporating these schemes in the second part of the investigation.
- For additive faults, the MRAC- $H_\infty$ GS-LPV controller based on Lyapunov theory was able to deal with the tested abrupt and gradual faults in actuators and sensors of nonlinear processes represented by LPV models and by the nonlinear process, and could accommodate the tested faults for any operating point between the operating ranges of the LPV system.
- For multiplicative faults, the MRAC- $H_\infty$ GS-LPV controller based on Lyapunov theory was in general the best controller, because was the one that could handle the faults more properly because is a combination of an MRAC design with the Lyapunov theory which adds closed loop stability to the system and also the controller has an  $H_\infty$  gain scheduling controller that was design to be stable for the operating points between the operating range.
- The schemes tested using the nonlinear system presented some chattering due to the nonlinearities of the system, but the schemes based on Lyapunov theory and specifically the scheme that combines MRAC with  $H_\infty$  gains scheduling control based on Lyapunov theory were able to deal and decrease the chattering. The chattering decreases if the  $\gamma$  in the MRAC controller decreases.
- The Lyapunov theory implemented to design the MRAC controllers guarantees closed-loop stability.
- The MRAC- $H_\infty$ GS-LPV was demonstrated to be stable by design because the MRAC based on Lyapunov theory controller guarantees closed-loop stability and the  $H_\infty$  Gain Scheduling controller was also designed to be stable along the specifies parameter trajectories.
- The combination of the MRAC controller with other schemes becomes an Active + Passive FTC because the MRAC accommodates the fault on-line but the combinational controllers (for example:  $H_\infty$ ) are passive controller because were designed off-line.
- The multiplicative faults in comparison with the additive faults applied in the LPV system of the Coupled-Tank system had more impact in the case of actuator faults, because after the occurrence of the fault the system became degraded and could not accommodate the fault. On the other hand, the multiplicative faults in comparison with the additive faults applied in the nonlinear system of the Coupled-Tank system had more impact in the case of 100% of fault because after the sensor fault the system was degraded and could not accommodate the fault, in the other fault cases the controller was able to accommodate the fault.

## 7 Future Work

As a continuity of this thesis the following future work will be realized:

- Quantify the values of the fault that can be tolerated by the FTC scheme, using an LPV Gain Scheduling methodology in order to establish an accepted fault range (Patton & Klinkhieo, 2010).
- A new FTC is proposed for future work (see Figure 110). This scheme consists of an FDI block that uses DPCA + Contribution Plots (Tudón et al., 2010; Tudón et al., 2011). After the identification and isolation of the fault an MRAC will be chosen from a bank of multiple MRAC model depending on the value of the fault (Tan et al. 2011). Each MRAC model has a different reference model according to the fault magnitude and according to the fault affectionation in the system. To save computational effort the adaptation mechanism will be active just after the detection and quantification of the fault. In the FDI block, the DPCA detect the fault and the Contribution Plots isolate the fault. Figure 111 shows the procedure of these methodologies (Tudón et al., 2010, Tudón et al., 2011).

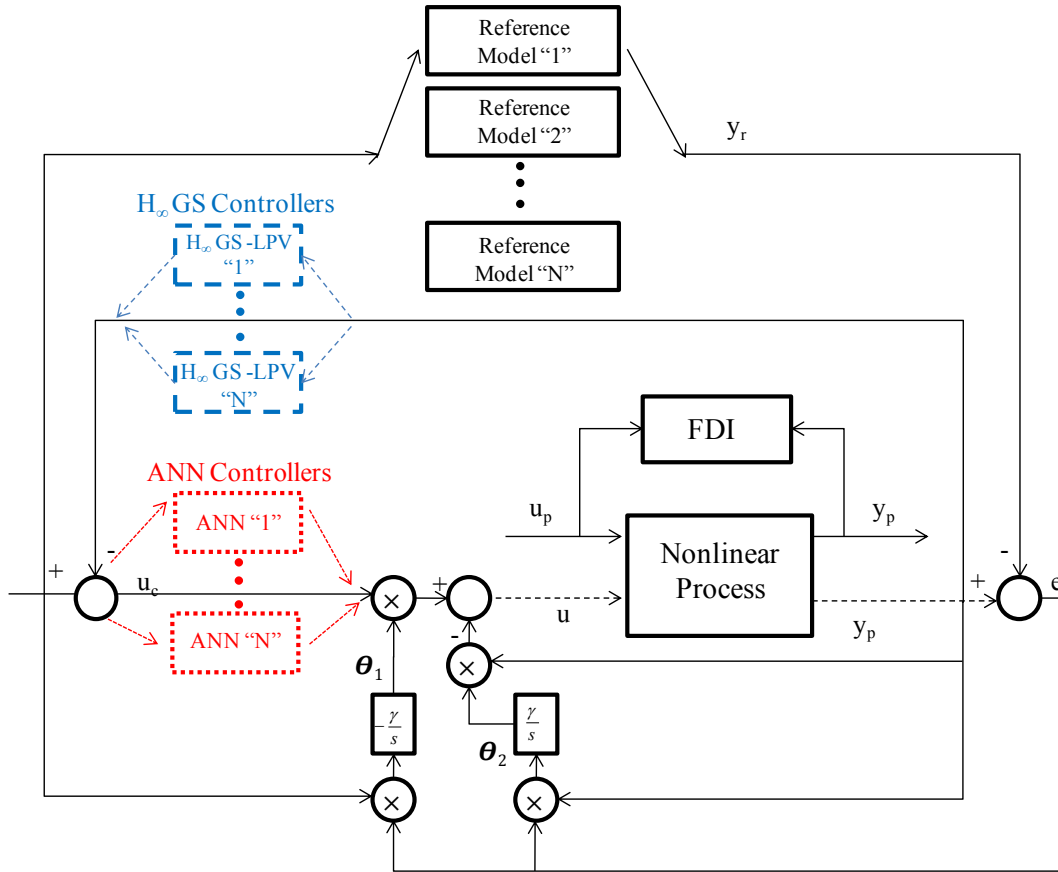
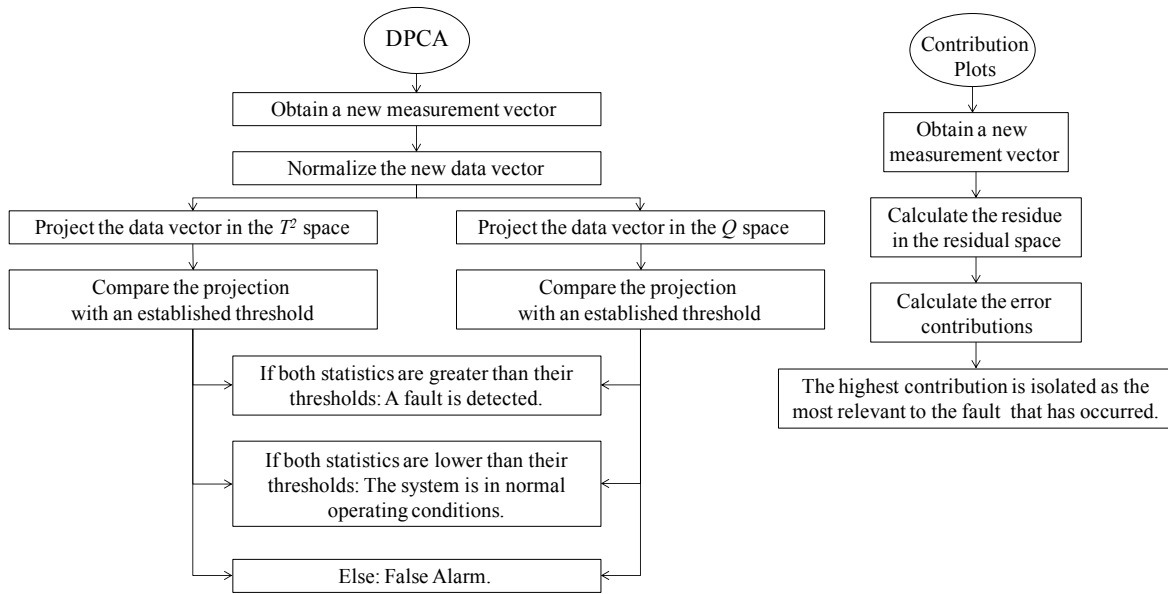


Figure 110. FDI + MRAC FTC Scheme.



**Figure 111. DPCA and Contribution Plots methodologies (Tudón et al. 2010).**

- To improve the fault tolerant capabilities of the MRAC schemes that use an ANN, it is recommended to prove this controller with other ANN architectures. The ANN architecture used in this investigation is a two-layer feed forward neural network.
- Test the LPV control schemes in a real physical experiment to corroborate the fault tolerant capabilities of the proposed schemes.
- Implement the proposed schemes in different types of systems to corroborate and compare the results of this thesis.
- Design an analytic methodology to establish the fault tolerant threshold.
- Establish the global stability.

## 8 References

1. Abdullah, A. & Zribi, M. (2009). Model reference control of LPV systems. *Journal of the Franklin Institute*, Elsevier, 346, 854-871.
2. Ahmed, M. (2000). Neural Net based MRAC for a Class of Nonlinear Plants. *Neural Networks*, 13(1), 111-124.
3. Alves, M., Nobrega, E. & Yoneyama, T. (2009). Adaptive Neural Control for a Tolerant Fault System. *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 137-142.
4. Anderson, B. & Vongpanitlerd. (1973). *Network Analysis*. Englewood Cliffs: Prentice-Hall.
5. Apkarian, P. (1999). *Coupled Water Tank Experiments Manual*. Canada: Quanser Consulting Inc.
6. Apkarian, P. & Gahinet, P. (1995). A Convex Characterization of Gain-Scheduled  $H_\infty$  Controllers. *IEEE Transactions on Automatic Control*, 40(5), 853-864.
7. Apkarian, P., Gahinet, P. & Becker, G. (1995). Self-scheduled  $H_\infty$  Control of Linear Parameter-varying Systems: a Design Example. *Journal of Automatica*, 21(9), 1251-1261.
8. Astrom, K. & Wittenmark, B. (1995). *Adaptive Control. 2<sup>nd</sup> Edition*. Addison-Wesley Publishing Company.
9. Bamieh, B. & Giarré, L. (2001). LPV Models: Identification for Gain Scheduling Control. In *European Control Conference*, Porto, Italy.
10. Bastani, F. & Chen, I. (1988). The role of artificial intelligence in fault-tolerant process-control systems. In *Proceedings of the 1st Int Conf on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* (pp. 1049-1058). New York, United States: ACM.
11. Beainy, F., Mau, A. & Commuri, S. (2009). Unmanned Aerial Vehicles Operational Requirements and Fault-Tolerant Robust Control in Level Flight. In *17<sup>th</sup> Mediterranean Conference on Control & Automation* (pp. 700-705). Thessaloniki, Greece: IEEE.
12. Becker, G. & Packard, P. (1994). Robust Performance of Linear Parametrically Varying Systems Using Parametrically Dependent Linear Feedback. *Systems and Control Letters*, 23, 205-215.
13. Bhattacharyya, S., Chapellat, H. & Keel, L. (1995). *Robust Control – The Parametric Approach*. Prentice Hall.
14. Blanke, M., Izadi-Zamanabadi, R., Bogh, R. & Lunau, Z. (1997). Fault tolerant control systems—A holistic view. *Control Engineering Practice*, 5(5), 693-702.
15. Blanke, M., Staroswiecki, M. & Wu, N. E. (2001). Concepts and methods in fault-tolerant control. In *Proceedings of the 2001 American Control Conference* (pp. 2606-2620). Arlington, Virginia, United States: IEEE.
16. Blanke, M., Kinnaert, M., Lunze, J. & Staroswiecki, M. (2003). *Diagnosis and Fault-Tolerant Control*. Berlin, Germany: Springer-Verlag.
17. Blondel, V. (1994). *Simultaneous Stabilization of Linear Systems*. Heidelberg, Germany: Springer Verlag.

18. Bosche, J., El Hajjaji, A. & Rabhi, A. (2009). Actuator Fault-tolerant control for vehicle dynamics. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 1003-1008). Barcelona, Spain: IFAC.
19. Caglayan, A., Allen, S. & Wehmuller, K. (1988). Evaluation of a second generation reconfiguration strategy for aircraft flight control systems subjected to actuator failure/surface damage. In *Proceedings of the 1988 National Aerospace and Electronics Conference* (pp. 520-529). Dayton, Ohio, United States: IEEE.
20. Chen, L., Patton, R. & Klinkhio, S. (2010). An LPV Pole-placement approach to friction compensation as an FTC. In *Conference on Control and Fault Tolerant Systems* (pp. 100-105). Nice, France: IEEE.
21. Cho, Y., Kim, K. & Bien, Z. (1990). Fault Tolerant Control using a Redundant Adaptive Controller. In *Twenty-ninth Conference on Decision and Control* (pp. 1467-1478). Honolulu, Hawaii, United States: IEEE.
22. Cruz-Reynoso, R. (2011). *Evaluación Experimental de Esquemas Adaptativos de Control Tolerante a Fallas Basados en el Modelo de Referencia*. Tesis de maestría. Tecnológico de Monterrey, Monterrey, México.
23. De Lira, S., Puig, V. & Quevedo, J. (2009). LPV Model-Based Fault Diagnosis Using Relative Fault Sensitivity Signature Approach in a PEM Fuel Cell. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 528-533). Barcelona, Spain: SAFEPROCESS.
24. Demetriou, A., Ito, K. & Smith, R. (2009). Adaptive techniques for the MRAC. Adaptive parameter identification, and on-line fault monitoring and accommodation for a class of positive real infinite dimensional systems. *International Journal of Adaptive Control and Signal Processing*, 23, 193-215.
25. Dong, Q., Zhong, M. & Ding, S. (2009). On Active Fault Tolerant Control for a Class of Time-Delay Systems. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 882-886). Barcelona, Spain: SAFEPROCESS.
26. Dong, X., Wang, H. & Xu, Q. (2006). Study on New Control Method for Boiler Combustion System. In *Proceedings of the 6<sup>th</sup> World Congress on Intelligent Control and Automation* (pp. 2944-2948). Dalian, China: IEEE.
27. Eterno, J., Looze, D., Weiss, J. & Willsky, A. (1985). Design Issues for Fault-Tolerant Restructurable Aircraft Control, In *Proceedings of 24th Conference on Decision and Control* (pp. 900-905). Fort Lauderdale, Florida, United States: IEEE.
28. Farrell, J., Berger, T. & Appleby, B. (1993). Using learning techniques to accommodate unanticipated faults. *IEEE Control Systems Magazine*, 13(3), 40-49.
29. Fodor, D. & Toth, R. (2004). Speed Sensorless Linear Parameter Variant  $H_\infty$  Control of the Induction Motor. In *43<sup>rd</sup> IEEE Conference on Decision and Control* (pp. 4435-4440). Bahamas: IEEE.
30. Forsythe, G. E., Malcolm, M. A. & Moler, C. B. (1977). *Computer Methods for Mathematical Computations*. New Jersey, United States: Prentice- Hall.
31. Fortuna, L. (2007). *Soft Sensors Monitoring and Control of Industrial Processes*. Springer.

32. Fradkov, A., Andrievsky, B. & Peaucelle, D. (2008). Adaptive Control Design and Experiments for LAAS “Helicopter” Benchmark. *European Journal of Control*, 4, 329-339.
33. Gao, Z. & Antsaklis, P. (1991). Stability of the pseudo-inverse method for reconfigurable control systems. *International Journal of Control*, 53(3), 717-729.
34. Giarré, L., Bauso, D., Falugi, P. & Bamieh, B. (2006). LPV model identification for gain scheduling control: An application to rotating stall and surge control problem. *Journal of Control engineering Practice*, 14(4), 351-361.
35. Gilbert, W., Henrion, D., Bernussou, J. & Boyer, D. (2008). Polynomial LPV synthesis applied to turbofan engines. *Journal of Control Engineering Practice*, 18(9), 1077-1083.
36. Ginter, V. & Pieper, J. (2009). Robust Gain Scheduled Control of a Hydrokinetic Turbine. Part 1: Design. In *Proceedings of the IEEE Electrical Power and Energy Conference* (pp. 1-6). Quebec, Canada: IEEE.
37. Glover, K. & McFarlane, D. (1989). Robust stabilization of normalized coprime factor plant descriptions with  $H_\infty$  bounded uncertainty. *IEEE Transactions on Automatic Control*, 34(8), 821-830.
38. Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Massachusetts, United States: Addison Wesley.
39. Gomaa, M. (2004). Fault tolerant control scheme based on multi-ann faulty models. In *ICEEC International Conference Electrical, Electronic and Computer Engineering* (pp. 329-332). IEEE.
40. Groot, M., Van de Wal, M., Scherer, G. & Bosgra, O. (2005). LPV control for a wafer stage: beyond the theoretical solution. *Journal of Control Engineering Practice*, 13, 231-245.
41. Hagan, M. & Menhaj, M. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989-993.
42. Hecker, S. (2006). Robust  $H_\infty$ -based vehicle steering control design. In *Proceeding of the IEEE International Conference on Control Applications* (pp. 1294-1299). Munich, Germany: IEEE.
43. Henry, D., Falcoz, A. & Zolghadri, A. (2009). Structured  $H_\infty/H_2$  LPV filters for fault diagnosis: Some new results. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Barcelona, Spain: SAFEPROCESS.
44. Hongjie, H. & Bo, Z. (2008). A New MRAC Method based on Neural Network for High-Precision Servo System. In *IEEE Vehicle Power and Propulsion Conference* (pp. 1-5). Harbin, China: IEEE.
45. Hsu, L. (1990). Variable structure model reference adaptive control (VS-MRAC) using only input and output measurements: the general case. *IEEE Transactions on Automatic Control*, 35 (11), 1238-1243.
46. Hsu, L., de Araújo, A. & Costa, R. (1994). Analysis and Design of I/O Based Variable Structure Adaptive Control. *IEEE Transactions on Automatic Control*, 39(1), 4-21.
47. Hwang, C. & Chen, B. (1988). Adaptive control of optimal model matching in  $H_\infty$ -norm space. *IEEE Proceeding of Control Theory and Applications*, 135(4), 295-301.



48. Isermann, R., Schwarz, R. & Stölzl, S. (2002). Fault-tolerant drive-by-wire systems. *IEEE Control Systems Magazine*, 22(5) 64-81.
49. Jia, L. & Jingping, J. (1997). The Model Reference Adaptive Control Based On the Genetic Algorithm. In *Proceedings of the International Conference on Neural Networks* (pp. 783-787). Houston, Texas, United States: IEEE.
50. Jiang, J. (1994). Design of reconfigurable control systems using eigenstructure assignments. *International Journal of Control*, 59(2), 395-410.
51. Guo, J., Liu, Y. & Tao, G. (2010). A multivariable MRAC Design Using State Feedback for Lenearized Aircraft Models with Damage. In *American Control Conference* (pp. 2095-2100). Baltimore, MD, United States: IEEE.
52. Guo, J., Tao, G. & Liu, Y. (2011). A Multivariable MRAC Design for Aircraft Systems under Failure and Damage Conditions. In *American Control Conference* (pp. 600-605). San Francisco, CA, United States: IEEE.
53. Kamalasan, S. & Ghandakly, A. (2011). A Neural Network Parallel Adaptive Controller for Fighter Aircraft Pitch-Rate Tracking. *IEEE Transactions on Instrumentation and Measurement*, 60(1), 258-267.
54. Karsai, G., Biswas, G., Narasimhan, S., Szemethy, T., Peceli, G., Simon, G. & Kovacsazy, T. (2002). Towards Fault-Adaptive Control of Complex Dynamic Systems, In: *Software- Enabled Control*, 347-368.
55. Khalil, H. (2002). *Nonlinear Systems*. 3<sup>rd</sup> Ed. Prentice Hall.
56. Korbicz, J., Koscielnny, J. & Kowalczyk, Z. (2004). *Fault diagnosis: models, artificial intelligence, applications*. Berlin, Germany: Springer-Verlag.
57. Kurihara, K., Matsumoto, S., Nishiuchi, N. & Masuda, K. (2009). Fault Tolerant Control Method for Gesture Reproduction Robot of Remote Person. In *International Conference on Computers & Industrial Engineering* (pp. 495-500). Troyes, France: IEEE.
58. Kwiatkowski, A. & Werner, H. (2005). LPV Control of a 2-DOF Robot Using Parameter Reduction. In *Proceeding of the 44<sup>th</sup> IEEE Conference on Decision and control, and the European Control Conference* (pp. 3369-3374). Seville, Spain: IEEE.
59. Le, V. & Safonov, M. (1992). Rational matrix GCD's and the design of squaring-down compensators: a state space theory. *IEEE Transactions on Automatic Control*, 37(3), 384-392.
60. Lee, S. & Park, J. (2007). Output feedback model predictive control for LPV system using parameter-dependent Lyapunov function. *Journal of Applied Mathematics and Computation*, 190, 671-767.
61. Li, Z. (2009). Fault Diagnosis and Fault Tolerant Control of Mobile Robot Based on Neural Networks. In *Proceeding of the Eight International Conference on Machine Learning and Cybernetics* (pp. 1077-1081). Baoding, China: IEEE.

62. Lian, K., Chiu, C. & Liu, P. (2002). Semi-Decentralized Adaptive Fuzzy Control for Cooperative Multirobot Systems with  $H_\infty$  Motion/Internal Force Tracking Performance. *IEEE Transaction on System, Man, and Cybernetics- Part B: Cybernetics*, 32(3), 269-280.
63. Liang, B. & Duan, G. (2004). Robust  $H_\infty$  fault-tolerant control for uncertain descriptor systems by dynamical compensators. *Journal of Control Theory and Applications*, 2(3), 288-292.
64. Liang, Y. & Marquez, H. (2008). Gain Scheduling Synchronization Method for Quadratic Chaotic System. *IEEE Transactions on circuits and systems – I: Regular Papers*, 55(4), 1097-1107.
65. Lin, F. (2007). *Robust Control Design - An Optimal Control Approach*. Wiley.
66. Lunze, J. & Richter, J. (2006). *Control reconfiguration: Survey of methods and open problems*. Bochum, Germany: ATP.
67. Luzar, M., Witczak, M., Puig, V. & Nejjar F. (2009). Development of a Fault-Tolerant Control with MATLAB and Its Application to the Twin-Rotor System. In *7th Workshop on Advanced Control and Diagnosis* (pp. 1-8).
68. Mahmoud, M., Jiang, J. & Zhang, Y. (2003). *Active fault tolerant control systems: Stochastic analysis and synthesis*. Berlin, Germany: Springer.
69. McFarlane, D. & Glover, K. (1989). *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*. Springer Verlag.
70. Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, Massachusetts, United States: MIT Press.
71. Miyasato, Y. (2006). Model Reference Adaptive Control of Polytopic LPV Systems – An Alternative Approach to Adaptive Control. In *Proceeding of the IEEE International Symposium on Intelligent Control* (pp. 2012-2017). Munich, Germany: IEEE.
72. Miyasato, Y. (2007). Model Reference Adaptive  $H_\infty$  for Distributed Parameter Systems of Hyperbolic Type by Finite Dimensional Controllers – construction with unbounded observation operator. In *Forty-sixth IEEE Conference on Decision and Control* (pp. 1338-1343). New Orleans, Louisiana, United States: IEEE.
73. Miyasato, Y. (2008). Model Reference Adaptive  $H_\infty$  Control for Flexible Arms by Finite Dimensional Controllers. In *Forty-seventh IEEE Conference on Decision and Control* (pp. 3257-3262). Cancun, Mexico: IEEE.
74. Montes de Oca, S., Puig, V., Theilliol, D. & Tornil-Sin, S. (2009). Fault-Tolerant Control Design using LPV Admissible Model Matching: Application to a Two-degree of Freedom Helicopter. In *17<sup>th</sup> Mediterranean Conference on Control & Automation* (pp. 522-527). Thessaloniki, Greece: IEEE.
75. Montes de Oca, S., Puig, V., Theilliol, D. & Tornil-Sin, S. (2009). Admissible Model Matching Fault Tolerant Control Design Based on LPV Fault Representation. *Int. J. of Appl. Math. Comput. Sci.*
76. Montes de Oca, S. & Puig, V. (2010). Fault-Tolerant Control Design using a Virtual Sensor for LPV Systems. In *Conference on Control and Fault Tolerant Systems* (pp. 88-93). Nice, France: IEEE.

77. Montes de Oca, S., Puig, V., Theilliol, D. & Tornil-Sin, S. (2010). Fault-Tolerant Control Design using LPV Admissible Model Matching with  $H_2/H_\infty$  Performance: Application to a Two-degree of Freedom Helicopter. *Conference on Control and Fault Tolerant Systems* (pp: 251-256). Nice, France: IEEE.
78. Nagrath, J. (2006). *Control Systems Engineering*. Indian Institute of Technology, Delhi, India: Anshan Ltd.
79. Neimann, H. & Stoustrup, J. (2005), Passive fault tolerant control of a double inverted pendulum - a case study. *Control Engineering Practice*, 13(8), 1047-1059.
80. Nguyen, H., Nadipuren, P., Walker, C. & Walker, E. (2002). *A First Course in Fuzzy and Neural Control*. United States: CRC Press Company.
81. Nieto, J., Castañon, J., Rabhi, A., El Hajjaji, A. & Morales-Menendez, R. (2009). Vehicle Fault Detection and Diagnosis combining AANN and ANFIS. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 1079-1084). Barcelona, Spain: SAFEPROCESS.
82. O'Dwyer, A. (2003). *Handbook of PI and PID Controller Tuning Rules*. Imperial College Press.
83. Packard, A. (1994). Gain Scheduling via Linear Fractional Transformations. *Syst. Contr. Letters*, 22, 79–92.
84. Pan, H., Wong, H., Kapila, V. & Queiroz, M. (2005). Experimental validation of a nonlinear backstepping liquid level controller for a state coupled two tank system. *Control Engineering Practice*, 13, 27-40.
85. Panagi, P. & Polycarpou, M. (2009). Decentralized Fault Accommodation of a Class of Interconnected Nonlinear Systems using an Adaptive Approximation Approach. In *17<sup>th</sup> Mediterranean Conference on Control & Automation* (pp. 546-551). Thessaloniki, Greece: IEEE.
86. Pashilkar, A., Sundararajan, N. & Saratchandran, P. (2006). A Fault-tolerant Neural Aided Controller for Aircraft Auto-landing. *Aerospace Science and Technology*, 10, 49-61.
87. Patan, K. & Korbicz, J. (2009). Fault detection and accommodation by means of neural networks. Application to the boiler unit. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 119-124). Barcelona, Spain: SAFEPROCESS.
88. Patton, R. J. (1997). Fault-tolerant control: The 1997 situation. *Proceedings of the 3rd IFAC symposium on fault detection, supervision and safety for technical processes* (pp. 1033–1055). Hull, United Kingdom: SAFEPROCESS.
89. Patton, R., Lopez-Toribio, C. & Uppal, F. (1999). Artificial intelligence approaches to fault diagnosis. *IEEE Condition Monitoring: Machinery, External Structures and Health*, 5/1–518.
90. Patton, R. & Klinkhieo, S. (2010). LPV fault estimation and FTC of a two-link manipulator. In *American Control Conference* (pp: 4647-4652). Baltimore, MD, United States: IEEE.
91. Perhinschi, M., Napolitano, M., Campa, G., Fravolini, M. & Seanor, B. (2007). Integration of Sensor and Actuator Failure Detection, Identification, and Accommodation Schemes within Fault Tolerant Control Laws. *Control and Intelligent Systems*, 35(4), 309-318.

92. Polycarpou, M. & Helmicki, A. (1995). Automated fault detection and accommodation: A learning systems approach. *IEEE Transactions on Systems*, 25(11), 1447-1458.
93. Polycarpou, M. & Vemuri, A. (1995). Learning methodology for failure detection and accommodation. *IEEE Control Systems Magazine*, 15(3), 16-24.
94. Polycarpou, M. (2001). Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach. *IEEE Transactions on Automatic Control*, 46(5), 736-742.
95. Polycarpou, M. & Helmicki, A. (1995). Automated fault detection and accommodation: A learning systems approach. *IEEE Transactions on Systems*, 25(11), 1447-1458.
96. Poudheh, M. (2008). An Adaptive Controller Based on Genetic Algorithm to Mitigate the Oscillations in Power System. *International Conference on Smart Manufacturing Application* (pp. 502-507), Gyeonggi-do, South Korea: IEEE.
97. Poussot-Vassal, C., Sename, O., Dugard, L., Gáspár, P., Szabó, Z. & Bokor, J. (2008). A new semi-active suspension control strategy through LPV technique. *Journal of Control Engineering Practice*, 16, 1519-1534.
98. Priddy, K. & Keller, P. (2005). *Artificial neural networks*. SPIE Press.
99. Robert, D., Sename, O. & Simon, D. (2007). A reduced polytopic LPV synthesis for a sampling varying controller: experimentation with a T inverted pendulum. In *European Control Conference ECC'07*. Kos, Greece.
100. Rodrigues, M., Theilliol, D., Aberkane, S. & Sauter, D. (2007). Fault Tolerant Control Design for Polytopic LPV Systems. *Int. J. Appl. Math. Comput. Sci.*, 17(1), 27-37.
101. Rumerhart, D., McClelland, J. & the PDP Research Group. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*. Cambridge, Massachusetts, United States: MIT Press.
102. Ruan, D. (1997). *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*. United States: Kluwer Academic Publishers.
103. Salcedo, J. & Martínez, M. (2006). Identificación de Modelos LPV para el Control de Sistemas No Lineales. *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*, 3(3), 92-107.
104. Sang, Q and Tao, G. (2009). Performance Robustness of MRAC under Reduction in Actuator Effectiveness. In *American Control Conference* (pp. 4506-4511). St. Louis, MO, United States: IEEE.
105. Schroder, P., Chipperfield, A., Fleming, P. & Grum, N. (1998). Fault tolerant control of active magnetic bearings. In *IEEE International Symposium on Industrial Electronics* (pp. 573-578). Pretoria, South Africa: IEEE.
106. Shamma, J. & Cloutier, J. (1993). Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations. *Journal of Guidance, Control and Dynamics*, 16(2), 256-263.
107. Skogestad, S. & Postlethwaite, I. (2005). *Multivariable Feedback Control. Analysis and Design*. Wiley Ed.

108. Staroswiecki, M. (2005). Fault tolerant control: The pseudo-inverse method revisited. In *Proceedings 16th IFAC World Congress* (pp. Th-E05-TO/2). Prague, Czech Republic: IFAC.
109. Steffen, T. (2005). *Control reconfiguration of dynamic systems: Linear approaches and structural tests*. Berlin, Germany: Springer.
110. Stengel, R. (1991). Intelligent Failure-Tolerant Control. *IEEE Control Systems Magazine*, 11(4), 14-23.
111. Sugawara, E., Fukushi, M. & Horiguchi, S. (2003). Fault Tolerant Multi-layer Neural Networks with GA Training. In *18th IEEE International Symposium on Defect and Fault Tolerance in VLSI systems* (pp. 328-335). Boston, Massachusetts, United States: IEEE.
112. Tan, C., Tao, G. & Qi, R. (2011). A Discrete-Time Multiple-Model MRAC Scheme for Adaptive Actuator Failure Compensation. In *Proceeding of 8<sup>th</sup> Asian Control Conference* (pp. 1276-1281). Kaohsiung, Taiwan: IEEE.
113. Tao, G. (1996). Inherent robustness of MRAC schemes. *System & Control Letters*, 29, 165-173.
114. Thanapalan, K., Veres, S., Rogers, E. & Gabriel S. (2006). Fault Tolerant Controller Design to Ensure Operational Safety in Satellite Formation Flying. In *Forty-fifth IEEE Conference on Decision & Control* (pp. 1562-1567). San Diego, California, United States: IEEE.
115. Tudón-Martínez, J., Morales-Menéndez, R. & Garza-Castañón, L. (2010). Fault Diagnosis in a Heat Exchanger using Process History Based-Methods. In *20<sup>th</sup> European Symposium on Computer Aided Process Engineering – ESCAPE20* (pp. 169-174). Ischia, Naples, Italy.
116. Tudón-Martínez, J., Morales-Menéndez, R., Ramírez-Mendoza, R., Garza-Castañón, L. & Vargas-Martínez, A. (2011). Fault Detection in a Heat Exchanger, Comparative Analysis between Dynamic Principal Component Analysis and Diagnostic Observers. *Revista Iberoamericana de Computación. Computación y Sistemas*, 14(3), 269-282.
117. Venkatasubramanian, V., Rengaswamy, R., Yin, K. & Kavuri, S. (2003a). A review of process fault detection and diagnosis. Part I. Quantitative modelbased methods. *Computers and Chemical Engineering*, 27(3), 293-311.
118. Venkatasubramanian, V., Rengaswamy, R. & Kavuri, S. (2003b). A review of process fault detection and diagnosis. Part II. Qualitative models and search strategies. *Computers and Chemical Engineering*, 27(3), 313-326.
119. Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. & Yin, K. (2003c). A review of process fault detection and diagnosis. Part III. Process history based methods. *Computers and Chemical Engineering*, 27(3), 327-346.
120. Vidyasagar, M. (1992). *Nonlinear System Analysis*. Englewood Cliffs: Prentice-Hall.
121. Wang, H. & Wang, Y. (1999). Neural-network-based fault-tolerant control of unknown nonlinear systems. *Control Theory and Applications*, 46(5), 389-398.
122. Wang, C. & Weiss, G. (2006). Self-Scheduled LPV Control of a Wind Driven Doubly-Fed Induction Generator. In *Proceeding of the 45<sup>th</sup> IEEE Conference on Decision & Control* (pp. 1246-1251). San Diego, CA, United States: IEEE.

123. Weidong, C. & Shaocheng, T. (2007). Fuzzy Adaptive Fault Tolerant Sliding Mode Control for SISO Nonlinear Systems. In *Second International Conference on Innovative Computing, Information and Control* (pp. 476-479). Kumamoto, Japan: ACM.
124. Whitaker, H., Yamron, J. & Kezer, A. (1958). *Design of Model Reference Adaptive Control Systems for Aircraft, Report R-164*. MIT Press.
125. Willems, J. (1971). Least-Squares Stationary Optimal Control and the Algebraic Riccati Equation. *IEEE Trans. Aut. Contr.*, 621 – 634.
126. Xiadong, H., Huang, H., Dexiao, X. & Wang, Z. (2008). Dynamic output feedback satisfactory fault-tolerant control with constraints of consistent indices. In *Seventh World Congress on Intelligent Control and Automation* (pp. 4241-4245). Chongqing, China: IEEE.
127. Xie, W. & Eisaka, T. (2008). Two-Degree-of-Freedom Controller Design for Linear Parameter-Varying Systems. *Asian Journal of Control*, 10(1), 115-120.
128. Yan, L., Hsu, L. & Xiuxia, S. (2006). A variable structure MRAC with expected transient and steady-state performance. *Journal of Automatica*, 42, 805-813.
129. Yang, S., Mid, E. & Mohamed, A. (2008). Modified Feedback Configuration for Sensor Fault Tolerant Control. In *10<sup>th</sup> International Conference on Control, Automation, Robotics and Vision* (pp. 2038-2043). Hanoi, Vietnam: IEEE.
130. Yang, G. & Ye, D. (2006). Adaptive fault-tolerant  $H_\infty$  control via state feedback for linear systems against actuator faults. In *Conference on Decision and Control* (pp. 3530-3535). San Diego, California, United States: IEEE.
131. Ye, N. (2003). *The Handbook of Data Mining: Human factors and ergonomics, Volume in the Human Factors and Ergonomics Series*. Lawrence Erlbaum Associates.
132. Yen, G. & De Lima, P. (2005). An Integrated Fault Tolerant Control Framework Using Adaptive Critic Design. In *International Joint Conference on Neural Networks* (pp. 2983-2988). IEEE.
133. Yen, G. & Ho, L. (2000). Fault Tolerant Control: An Intelligent Sliding Mode Control Strategy. In *Proceeding of the American Control Conference* (pp. 4204-4208). Chicago, Illinois, United States: IEEE.
134. Yu, D. L., Chang, T. & Yu, D. W. (2005). Fault Tolerant Control of Multivariable Processes Using Auto-Tuning PID Controller. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 35(1), 32-43.
135. Yu, D. L., Chang, T. & Yu, D. W. (2005). Adaptive Neural Model-Based Fault Tolerant Control for Multi-Variable Processes. *Engineering Applications of Artificial Intelligence*, 18, 393-411.
136. Yu, R., Breikin, T. & Wang, H. (2009). A Hybrid ANN-based Fault Tolerance Method and its Application. In *7<sup>th</sup> IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (pp. 378-383). Barcelona, Spain: SAFEPROCESS.
137. Yu, W. (2004).  $H_\infty$  Tracking-based adaptive fuzzy-neural control for MIMO uncertain robotic systems with time delays. *Fuzzy Sets and Systems*, 146, 375-401.

138. Zames, G. (1981). Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverse. *IEEE Transactions on Automatic Control*, 26(2), 301-320.
139. Zhang, M. & Li, W. (2006). Single Neuron PID Model Reference Adaptive Control Based on RBF Neural Network. In *Proceeding of the Fifth International Conference on Machine Learning and Cybernetics* (pp. 3021-3025). Dalian, China: IEEE.
140. Zhang, Y. & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Elsevier Annual Reviews in Control*, 32, 229-252.
141. Zhang, Z., Long, Z., She, L. & Chang, W. (2007). Fault-tolerant Control for Maglev Suspension System Based on Simultaneous Stabilization. In *Proceeding of the IEEE, International Conference on Automation and Logistics* (pp. 299-303). Jinan, China: IEEE.
142. Zhu, O., Kim, B., House, B. & Kim, K. J. (1999). An Adaptive Controller for Wolsong NGS Bulk Liquid Zone Control of RRS. In *Nuclear Science Symposium* (pp. 1699-1703). Seattle, Washington, United States: IEEE.

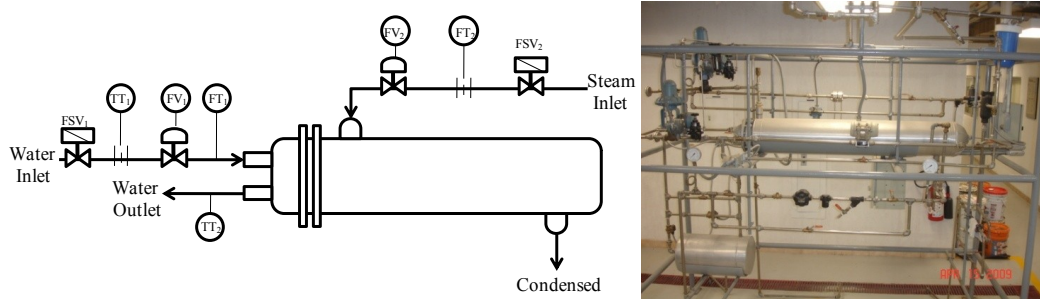
## Appendix A

Three Fault Tolerant Control (FTC) schemes based on Model Reference Adaptive Control (MRAC) for Single Input-Single Output (SISO) processes are compared in an experimental study. The first structure, named MRAC-MIT, is based on an MRAC design using the MIT rule, the second scheme, named MRAC-LYA, is based on an MRAC design using the Lyapunov theory, and the third scheme is a Variable Structure MRAC (VS-MRAC) modified with a Saturation Function and a PI controller (VS-MRAC-SAT-PI) (Cruz-Reynoso R, 2010). The three FTC controllers were implemented and tested first in an Industrial Heat Exchanger (simulation) and then in a Coupled-Tank System (real physical implementation). In order to compare the performance of these schemes, different types of additive faults (abrupt and gradual faults) implemented in sensors and actuators with different magnitudes were tested. Results showed the robustness of the three MRAC-based FTC schemes to different fault scenarios.

To test the different approaches proposed in this research, two different experiments were developed. First, the three schemes were tested in an Industrial Heat Exchanger, these experiments are all simulated. Second, another experiment using a Coupled-Tank system is chosen. This experiment was tested in simulation and in a real physical Coupled-Tank system.

### Industrial Heat Exchanger

The first system used as testbed (shown in Figure A.1) is a shell and tube Industrial Heat Exchanger that has two inputs: water and steam flows controlled by pneumatic valves ( $FSV_1$  and  $FSV_2$ , respectively). The water pass inside the tubes at room temperature and the steam pass through the tube walls in order to transfer heat to the water. In addition, the industrial heat exchanger has one output, in which the water temperature is measured by a thermistor ( $TT_2$ ). Variations in water and steam flows are determined by flow transmitters ( $FT_1$  and  $FT_2$ , respectively).



**Figure A. 1. Industrial Heat Exchanger used in the experiments.**

To obtain the continuous model of this process, an identification experiment was performed, where a Pseudo Random Binary Sequence (PRBS) was applied to water and steam valves, and variations in water temperature were recorded. With the data obtained in the PRBS test, the identification was achieved using Matlab®. The following model was obtained:



$$G_p = G_{steam} + (-G_{water}) \quad (208)$$

$$G_p = \frac{0.00002}{s^2+0.004299s+0.00002} + \frac{-0.000013}{s^2+0.007815s+0.00008} \quad (209)$$

$$T(s) = \frac{0.00002}{s^2+0.004299s+0.00002} F_{steam}(s) - \frac{0.000013}{s^2+0.007815s+0.00008} F_{water}(s) \quad (210)$$

$$G_{steam} = \frac{T(s)}{F_{steam}(s)} = \frac{0.00002}{s^2+0.004299s+0.00002} \quad (211)$$

$$G_{water} = \frac{T(s)}{F_{water}(s)} = \frac{-0.000013}{s^2+0.007815s+0.00008} \quad (212)$$

where  $G_p$  represents the Process Model,  $G_{steam}$  and  $G_{water}$  describes the steam and water model of the industrial heat exchanger, respectively.  $T(s)$  describes the Water Temperature at the exit and  $F_{steam}(s)$  and  $F_{water}(s)$  represent the steam and water flow, respectively.

#### *Model Reference Adaptive Controller for the Industrial Heat Exchanger*

In order to derive an MRAC for the Industrial Heat Exchanger, it is important to take in account the two second order systems: steam and water systems. With the background theory presented in Section II, the following equations were developed:

$$y_{steam\_process} = G_{steam} * u = \left( \frac{0.00002}{s^2+0.004299s+0.00002} \right) (\theta_1 u_c - \theta_2 y_{process}) = \left( \frac{0.00002\theta_1}{s^2+0.004299s+0.00002+0.00002\theta_2} \right) u_c \quad (213)$$

$$y_{water\_process} = G_{water} * u = \left( \frac{-0.000013}{s^2+0.007815s+0.00008} \right) (\theta_1 u_c - \theta_2 y_{process}) = \left( \frac{-0.000013\theta_1}{s^2+0.007815s+0.00008-0.000013\theta_2} \right) u_c \quad (214)$$

Using equations (213) and (214), the error can be redefined as:

$$e_{steam} = \left( \frac{0.00002\theta_1}{s^2+0.004299s+0.00002+0.00002\theta_2} \right) u_c - (G_{steam\_reference\_model} * u_c) \quad (215)$$

$$e_{water} = \left( \frac{-0.000013\theta_1}{s^2+0.007815s+0.00008-0.000013\theta_2} \right) u_c - (G_{water\_reference\_model} * u_c) \quad (216)$$

Therefore, the error partial derivatives with respect to the adaptive feedforward ( $\theta_1$ ) and adaptive feedback ( $\theta_2$ ) gain are specified as equation 217 for the steam process and equation 218 for the water process:

$$\partial e / \partial \theta_1 = \left( \frac{0.00002}{s^2+0.004299s+0.00002+0.00002\theta_2} \right) u_c \quad \text{and} \quad \partial e / \partial \theta_2 = - \left( \frac{0.00002\theta_1}{s^2+0.004299s+0.00002+0.00002\theta_2} \right) y_{process} \quad (217)$$

$$\partial e / \partial \theta_1 = \left( \frac{-0.000013}{s^2+0.007815s+0.00008-0.000013\theta_2} \right) u_c \quad \text{and} \quad \partial e / \partial \theta_2 = - \left( \frac{b_r \theta_1}{s^2+0.007815s+0.00008-0.000013\theta_2} \right) y_{process} \quad (218)$$

Consequently, the Process characteristic equation can be transformed into equations (219) and (220), because the MRAC system aim is to approximate the Process Model with the Reference Model.

$$steam \rightarrow s^2 + 0.004299s + 0.00002 + 0.00002\theta_2 \approx s^2 + 0.004299s + 0.00002 \quad (219)$$

$$water \rightarrow s^2 + 0.007815s + 0.00008 - 0.000013\theta_2 \approx s^2 + 0.007815s + 0.00008 \quad (220)$$

Finally, from equations (219) and (220), the error partial derivatives are transformed; and employing the MIT rule, the update rules for the adaptive feedforward ( $\theta_1$ ) and adaptive feedback ( $\theta_2$ ) gain are obtained as follows:

$$(221)$$

$$steam \rightarrow \frac{d\theta_1}{dt} = -\gamma \left( \frac{0.004299s+0.00002}{s^2+0.004299s+0.00002} u_c \right) e \text{ and } \frac{d\theta_2}{dt} = \gamma \left( \frac{0.004299s+0.00002}{s^2+0.004299s+0.00002} y_{process} \right) e \quad (222)$$

$$water \rightarrow \frac{d\theta_1}{dt} = -\gamma \left( \frac{0.007815s+0.00008}{s^2+0.007815s+0.00008} u_c \right) e \text{ and } \frac{d\theta_2}{dt} = \gamma \left( \frac{0.007815s+0.00008}{s^2+0.007815s+0.00008} y_{process} \right) e$$

On the other hand, for the MRAC controller based on Lyapunov theory for the Industrial Heat Exchanger no matter if is for the steam process or for the water process the difference is that the adaptive feed forward ( $\theta_1$ ) and the adaptive feedback ( $\theta_2$ ) update rules are:

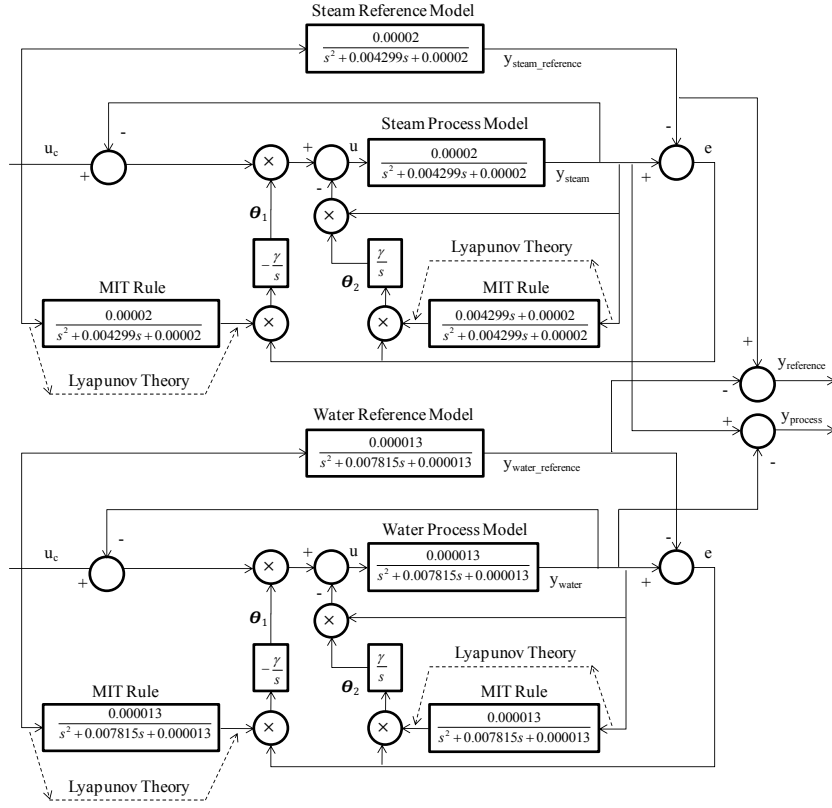
$$\frac{d\theta_1}{dt} = -\gamma u_c e \quad (223)$$

$$\frac{d\theta_2}{dt} = \gamma y_p e \quad (224)$$

With the above equations the controller of Figure A. 2 was implemented.

#### *VS-MRAC-SAT-PI for the Industrial Heat Exchanger*

In order to achieve the VS-MRAC-SAT-PI controller proposed in this research for the Industrial Heat Exchanger, first, a simple VS-MRAC controller was designed. This controller is implemented following the method presented in (Hsu, 1988) and (Hsu, 1990). The principal characteristic of this methodology is that it uses commutation function as sign functions. Then, a second structure was proposed (VS-MRAC-SAT). In this structure the original scheme is modified substituting the sign functions by saturation function. This modification is recommended to smooth the control signal and to avoid high frequency oscillations known as chattering. Finally, the third and last scheme, which is the one proposed in this research adds a Proportional Integer (PI) controller to the original VS-MRAC controller. This scheme uses sign functions to realize the commutation.



**Figure A. 2. Fault Tolerant MRAC Controller Structure.**

For the first structure (VS-MRAC), to obtain a reference model for the steam and water system with unitary gain and a settling time of 2500 seconds and 1600 seconds, respectively, the  $M(s)$  is defined as:

$$M_{steam}(s) = \frac{K}{(s+a_m)^2} = \frac{0.000007}{(s+0.002632)^2} \quad (225)$$

$$M_{water}(s) = \frac{K}{(s+a_m)^2} = \frac{0.000017}{(s+0.004112)^2} \quad (226)$$

Considering that  $a_m$  takes the value of 0.002632 for the steam system and a value of 0.004112 for the water system and that the design method indicates that  $\omega_0 \gg a_m$ , the value of  $\omega_0$  is

$$\omega_{0\_steam} = 5a_m = 5 * 0.002632 = 0.01316 \quad (227)$$

$$\omega_{0\_water} = 5a_m = 5 * 0.004112 = 0.02056 \quad (228)$$

Using  $\zeta=0.456$  for both systems (steam and water) and  $\omega_0=0.01316$  and  $\omega_0=0.02026$  for the steam and water systems, respectively, the following filter transfer function is obtained:

$$F^{-1}(s)_{steam} = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} = \frac{0.000173}{s^2 + 0.012002s + 0.000173} \quad (229)$$

$$F^{-1}(s)_{water} = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0} = \frac{0.000423}{s^2 + 0.018751s + 0.000423} \quad (230)$$

From the Industrial Heat Exchanger second order model, the values of  $a_I=0.004299$  and  $a_I=0.007815$  for the steam and water systems, respectively, are obtained. Therefore the polynomial  $L(s)$  is define by

$$L(s)_{steam} = s + a_1 = s + 0.004299 \quad (231)$$

$$L(s)_{water} = s + a_1 = s + 0.007815 \quad (232)$$

From the multiplication of the above equations by their respective reference model, the following equations are form:

$$ML_{steam} = \frac{0.000007s + (3.0093 \times 10^{-8})}{s^2 + 0.005264s + 0.000007} \quad (233)$$

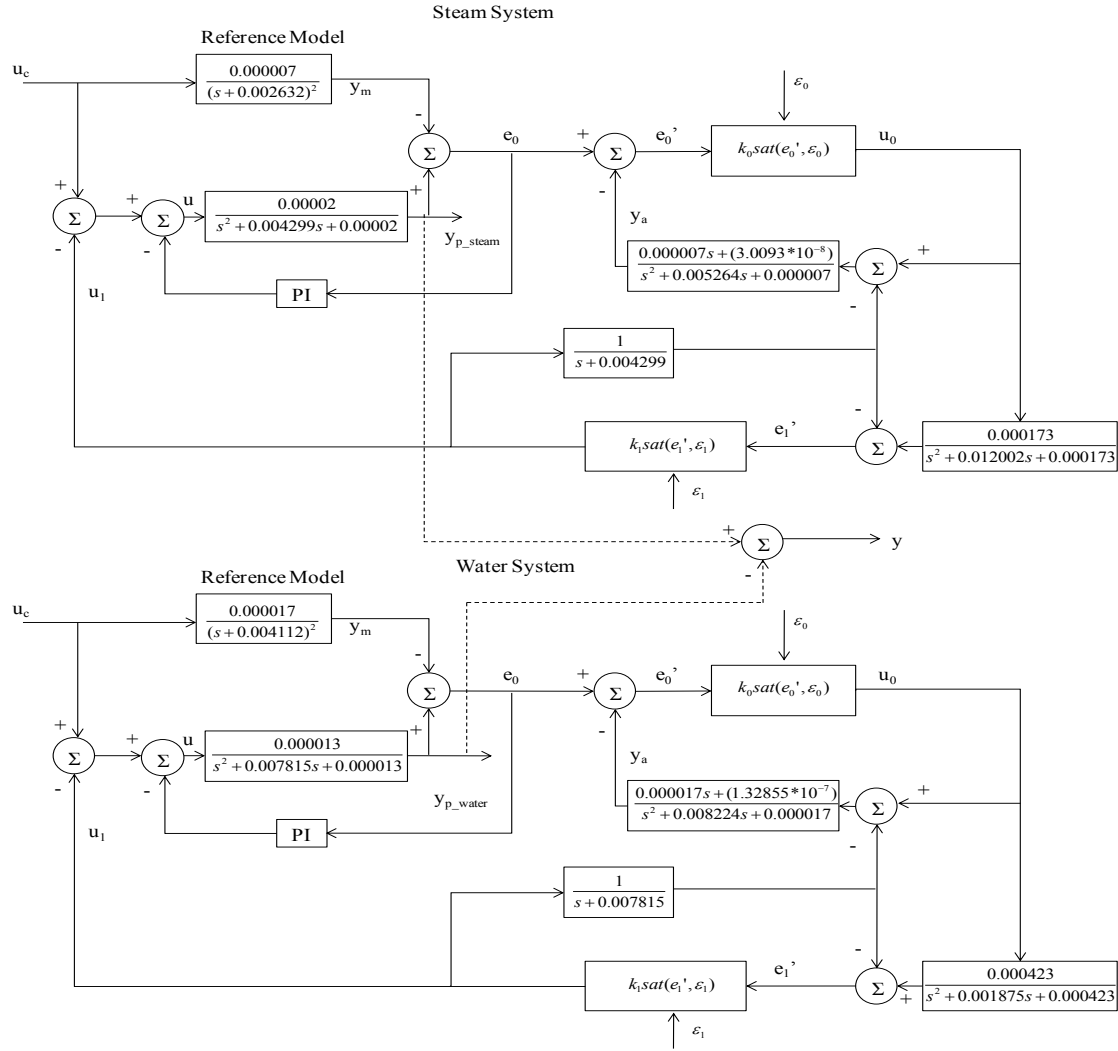
$$ML_{water} = \frac{0.000017s + (1.32855 \times 10^{-7})}{s^2 + 0.008224s + 0.000017} \quad (234)$$

The value of  $k_I$  is selected as 1, therefore the value of  $k_\theta$  is represented by

$$k_{0\_steam} = \frac{k_1}{a_m} = \frac{1}{0.002632} = 379.939 \quad (235)$$

$$k_{0\_water} = \frac{k_1}{a_m} = \frac{1}{0.004112} = 243.191 \quad (236)$$

Then, the sign of the classical VS-MRAC scheme is changed by a saturation function (VS-MRAC-SAT). The saturation function will behave as a sign function just when the absolute value of the input signal is higher that the design parameter value  $\varepsilon$ . If the input signal has an absolute value lower than  $\varepsilon$ , the output signal from the saturation function is equal to the input signal multiplied by the parameter  $k_\theta$  or  $k_I$ . The value of  $\varepsilon_\theta$  and  $\varepsilon_I$  was determined in an empiric form from the simulation of the system ( $\varepsilon_\theta=0.01$  and  $\varepsilon_I=2$  for the steam and water system). Finally, the proposed scheme presented in this research is a VS-MRAC with a Saturation function and a PI controller (VS-MRAC-SAT-PI). In this structure a PI controller is added to the VS-MRAC-SAT structure. The PI controller  $a_P$ ,  $c_P$ ,  $a_I$  and  $c_I$  parameters were selected in an empiric form through simulation ( $a_P=1$ ,  $c_P=320$ ,  $a_I=1.6$ ,  $c_I=0.99$  for the steam system and  $a_P=1$ ,  $c_P=320$ ,  $a_I=1.6$ ,  $c_I=0.99$  for the water system). Finally, the VS-MRAC-SAT-PI is represented by Figure A. 3.



**Figure A. 3. VS-MRAC-SAT-PI Structure for the Industrial Heat Exchanger.**

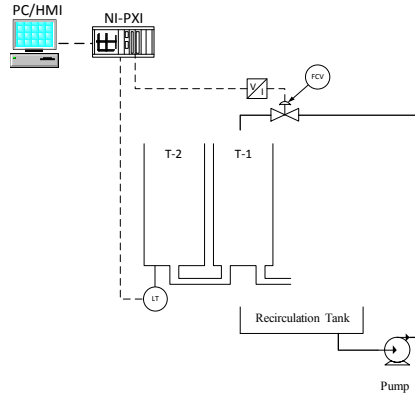
### Coupled-Tank System

A real Coupled-Tank system control station was used to design; implement and test the proposed controller (see Figure A.4).



**Figure A. 4. Physical Implementation of the Coupled-Tank System.**

In Figure A.5 a schematic diagram of the Coupled-Tank control station is represented. This system consist in a pair of coupled cylindrical tanks interconnected by the inferior part, both tanks have a height of 0.8 m and a diameter of 0.15 m. A pneumatic control valve (FCV) regulates the flow of water that feeds tank T-1, this valve is controlled by a current signal that varies from 4 to 20 mA. The water level is measured in tank T-2 using a level transmitter (LT) which delivers a current signal from 4 to 20 mA. The water discharge is located in Tank T-1. The sensor and control signals are generated and acquired using a data acquisition card NI-PXI from National Instruments. After processing, the control variable measures the percentage of valve opening and the process variable measures the percentage of the tank capacity. The implementation of the controllers is realized using Matlab® and Labview®. The control objective is to regulate the input flow in a way that the liquid level in Tank T-2 follows a reference signal. The control system must be able to follow the reference signal in spite of the presence of level sensor and control valve faults.



**Figure A. 5. Coupled-Tank System Control Scheme Schematic Diagram.**

To obtain a transfer model of the above scheme a PRBS test was applied to the Coupled-Tank system. From the results of the PRBS test and using the Identification Control Toolbox of Matlab®, the following second order transfer function was obtained.

$$G_p(s) = \frac{0.02422e^{-1.4s}}{s^2 + 0.3746s + 0.003637} \quad (237)$$

### Model Reference Adaptive Controller for the Coupled-Tank System

Two different MRAC were designed for the Coupled-Tank System, an MRAC based on the MIT rule and an MRAC based on the Lyapunov theory.

The reference model used in both controllers was chosen in order to have a maximum peak (MP) of overshoot of 20% and a settling time ( $t_s$ ) of 60 seconds. A reference model that achieves the above characteristic is:

$$G_r(s) = \frac{0.02138}{s^2 + 0.1333s + 0.02138} \quad (238)$$

Based on the theory explained in the Background theory section, the adaptation laws of the MRAC based on the MIT rule for the above reference model are represented by (see figure A. 6):

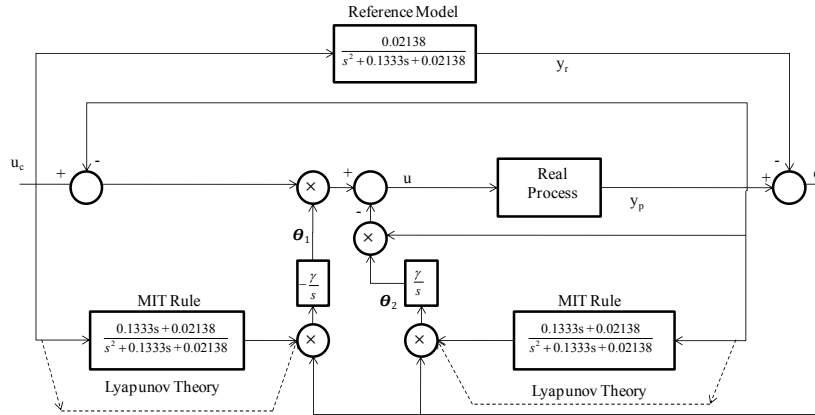
$$\frac{d\theta_1}{dt} = -\gamma \left( \frac{0.1333s + 0.02138}{s^2 + 0.007815s + 0.00008} u_c \right) e \quad (239)$$

$$\frac{d\theta_2}{dt} = \gamma \left( \frac{0.1333s + 0.02138}{s^2 + 0.007815s + 0.00008} y_{process} \right) e \quad (240)$$

On the other hand, the adaptation laws for the MRAC base don Lyapunov theory are given by (see figure A. 6):

$$\frac{d\theta_1}{dt} = -\gamma u_c e \quad (241)$$

$$\frac{d\theta_2}{dt} = \gamma y_p e \quad (242)$$



**Figure A. 6. Coupled-Tank MRAC scheme based on MIT rule and based on Lyapunov theory.**

In order to achieve the VS-MRAC-SAT-PI for the Coupled-Tank system, the following procedure was realized. For the first structure (VS-MRAC), to obtain a reference model with unitary gain and a settling time of 60 seconds  $M(S)$  is defined as:

$$M_{steam}(s) = \frac{K}{(s+a_m)^2} = \frac{0.04}{(s+0.2)^2} \quad (243)$$

Considering that  $a_m$  takes the value of 0.2 and that the design method indicates that  $\omega_0 > a_m$ , the value of  $\omega_0$  is

$$\omega_0 = 5a_m = 1 \quad (244)$$

Using  $\zeta=0.707$  and  $\omega_0=1$ , the following filter transfer function is obtained:

$$F^{-1}(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 + \omega_0} = \frac{1}{s^2 + 1.414s + 1} \quad (245)$$

From the Coupled-Tank system second order model, the value of  $a_I=0.3746$ , therefore the polynomial  $L(s)$  is define by

$$L(s)_{steam} = s + a_1 = s + 0.3746 \quad (246)$$

From the multiplication of the above equation by the reference model, the following equation is form:

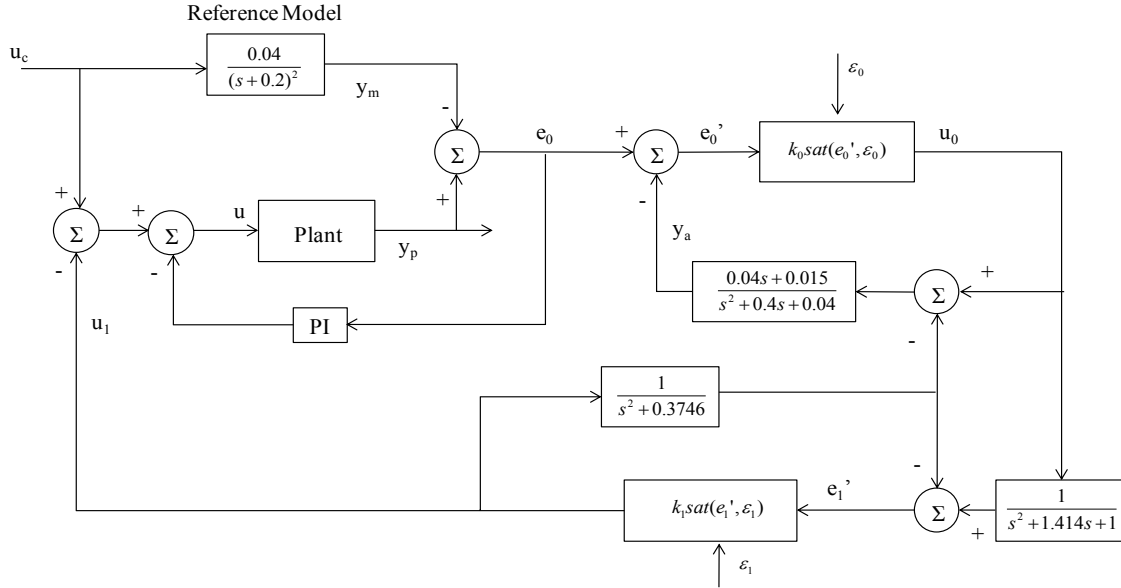
$$ML_{steam} = \frac{0.04s+0.015}{s^2+0.4s+0.04} \quad (247)$$

The value of  $k_I$  is selected as 20, therefore the value of  $k_0$  is represented by

$$k_0 = \frac{k_1}{a_m} = 100 \quad (248)$$

Then, the second structure (VS-MRAC-SAT) was developed. The value of  $\varepsilon_0$  and  $\varepsilon_I$  was determined in an empiric form from the simulation of the system ( $\varepsilon_0=0.01$  and  $\varepsilon_I=0.5$ ). And finally, the proposed scheme presented in this research (VS-MRAC-SAT-PI) was developed. The values of PI parameters ( $a_P$ ,  $c_P$ ,  $a_I$  and  $c_I$ ) were selected in an empiric form through simulation ( $a_P=10$ ,  $c_P=0.2$ ,  $a_I=20$  and  $c_I=0.3$ ) (see Figure A.7.).





**Figure A. 7.** VS-MRAC-SAT-PI Structure for the Coupled-Tank System.

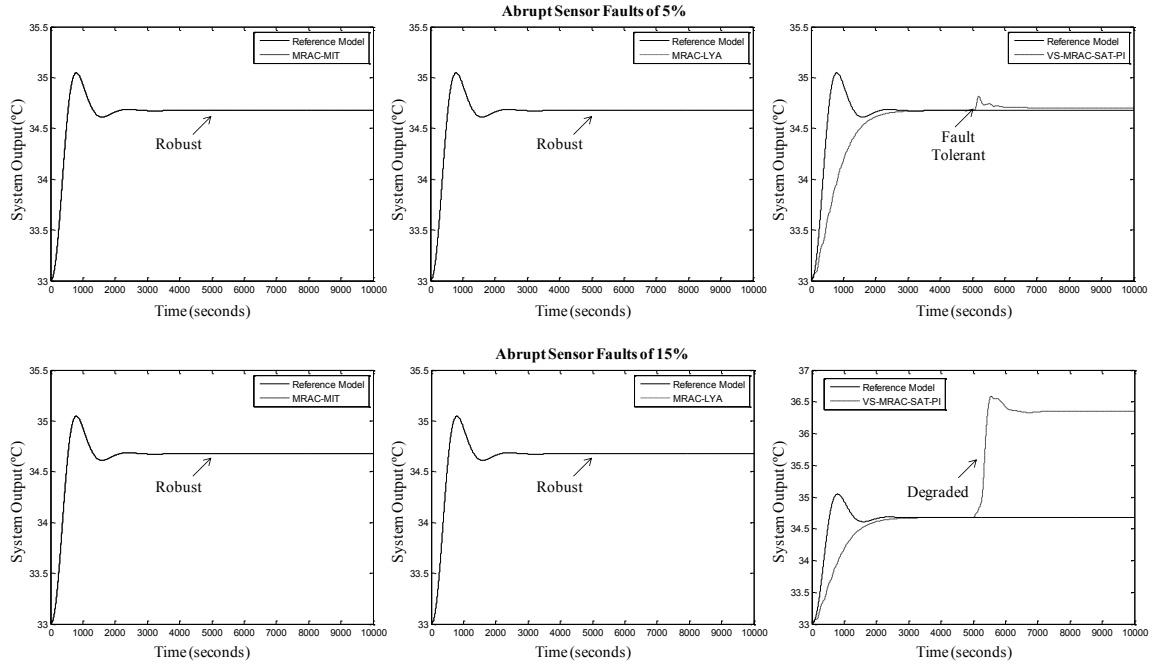
## Results

To test the above approaches, two different types of faults are introduced in the considered testbed case. The first type of fault is an additive abrupt fault and the second type of fault is an additive gradual fault. All types of faults are introduced in actuator and sensors. An additive fault will modify the quantity of the nominal value by the addition of a quantity  $f(t)$ . An abrupt additive fault in actuators represent, for instance, a pump stuck or in sensors a constant bias in measurements. A gradual additive fault could be a progressive loss of electrical power in pump or a drift in the sensor measurements.

As mentioned before, the above type of faults were tested in two different systems: a simulated Industrial Heat Exchanger and a real physical Coupled-Tank System. In each system, the three developed schemes were tested (MRAC based on MIT rule, MRAC based on Lyapunov theory and VS-MRAC-SAT-PI).

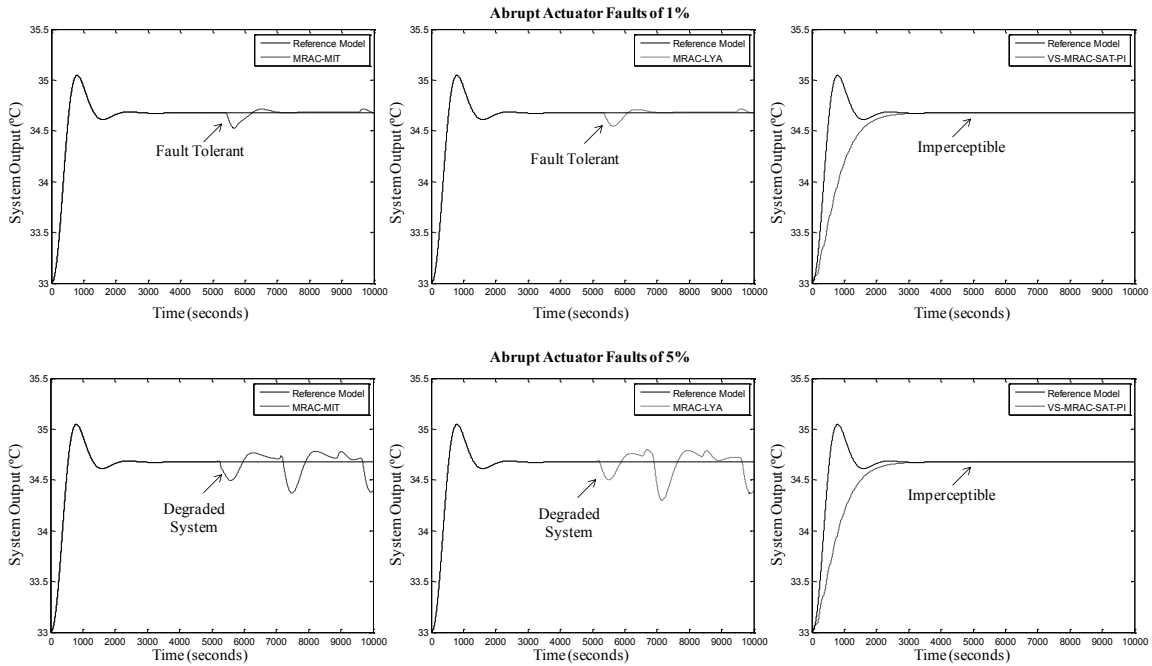
### *Simulation Results for the Industrial Heat Exchanger*

To test and compare the proposed schemes in the Industrial Heat Exchanger, the same operating conditions were applied for the three schemes (MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI). It is important to mention that the faults are introduced after the stabilization of the system. The total test time is of 15000 seconds, the sensor or the actuator faults (abrupt or gradual for both cases) were introduced at 5000 seconds. For sensor faults two different faults magnitudes were proved (5% and 15%). On the other hand, for actuator faults the magnitude of the faults were chosen as 1% and 5%. In addition the MSE error was calculated for each of the above experiments (see Table A-II). Figure A. 8 and Figure A. 9 shows the implementation of abrupt sensor and abrupt actuator faults, while Figure A. 10 and Figure A. 11 shows the implementation of gradual sensor and gradual actuator faults.



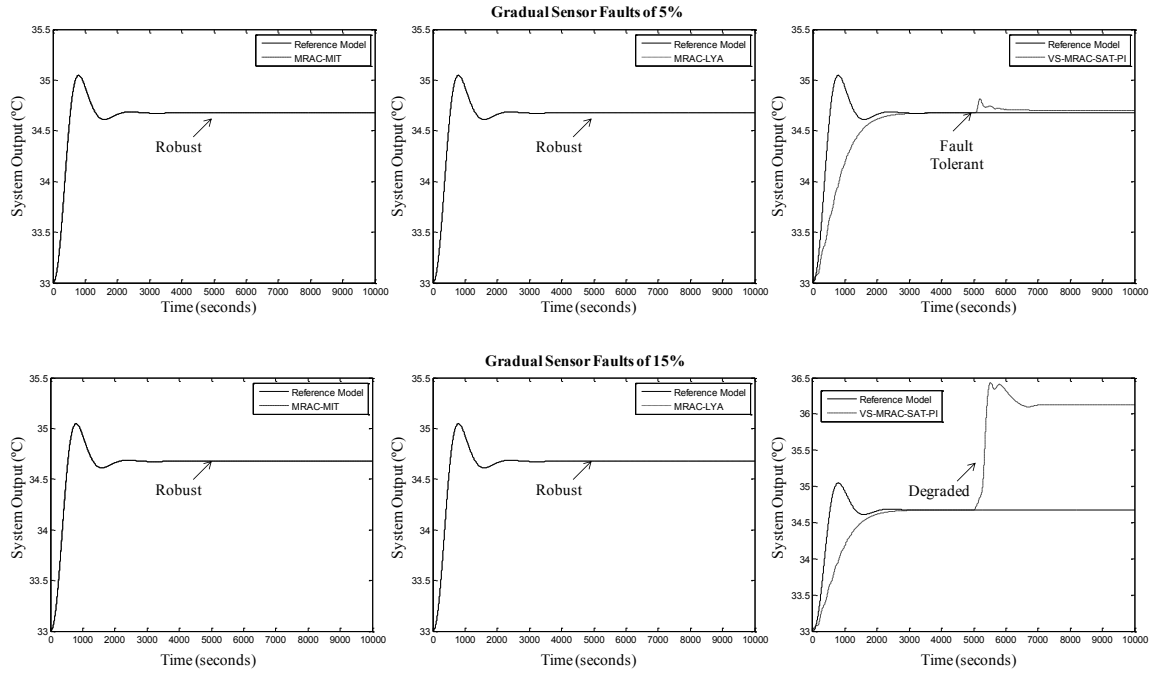
**Figure A. 8. Abrupt Sensor Faults of 5% and 15% tested in the MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI schemes.**

In Figure A. 8, it can be observed that the MRAC-MIT and the MRAC-LYA schemes were robust against abrupt sensor faults of 5% and 15%, while the VS-MRAC-SAT-PI scheme was fault tolerant for abrupt sensor faults of 5% but was degraded for abrupt sensor fault of 15%.

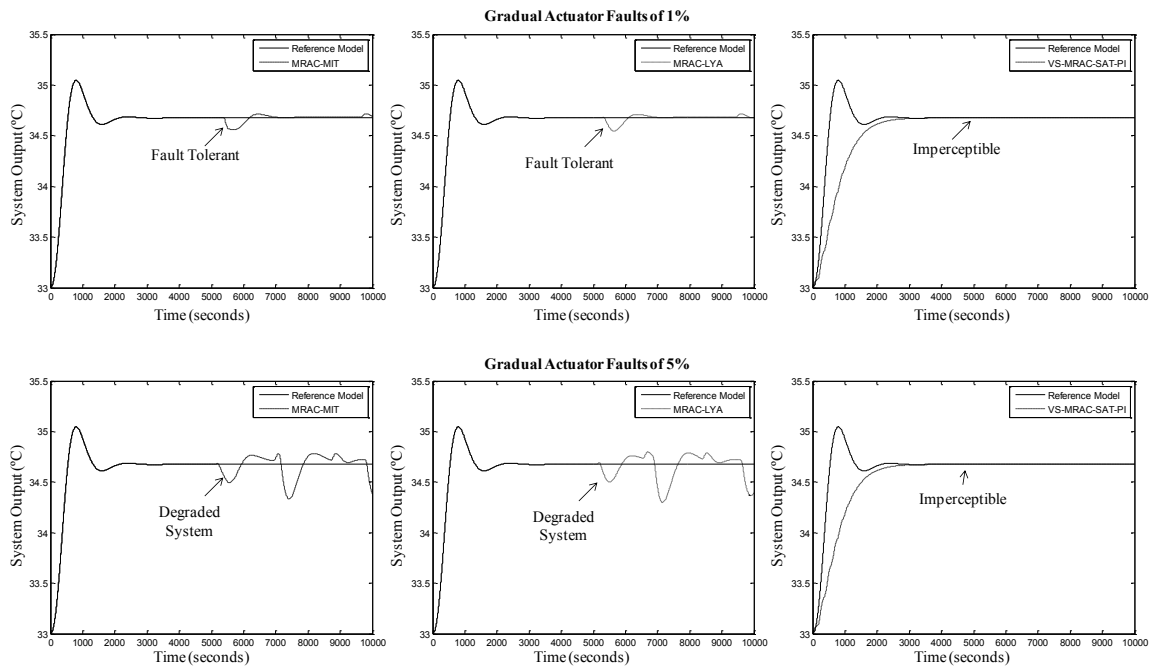


**Figure A. 9. Abrupt Actuator Faults of 1% and 5% tested in the MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI schemes.**

In Figure A. 9, it can be shown that the MRAC-MIT and the MRAC-LYA schemes became degraded after the occurrence of the abrupt actuator fault of 1% and 5%, while the fault of the VS-MRAC-SAT-PI scheme was imperceptible for both magnitudes (abrupt actuator faults of 1% and 5%).



**Figure A. 10. Gradual Sensor Faults of 5% and 15% tested in the MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI schemes.**



**FigureA. 11. Gradual Actuator Faults of 1% and 5% tested in the MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI schemes.**

The same results of abrupt faults (Figure A. 8 and Figure A. 9) were obtained testing the gradual faults (see Figure A. 10 and Figure A. 11).

The summary of the above results can be observed in Table II. And Table III presents the result of the MSE from the above tested types of faults.

Table A-II corroborate the results showed from Figure A. 8 to Figure A. 11 and the results of Table II, in which in general the best designed schemes for abrupt or gradual sensor faults were the MRAC-MIT and the MRAC-LYA schemes. And the best designed scheme for abrupt or gradual actuator faults was the VS-MRAC-SAT-PI scheme.

**Table A- I. Summary of results of Abrupt or Gradual Faults**

Proposed Schemes	Abrupt or Gradual Faults			
	Sensor Faults		Actuator Faults	
	5%	15%	1%	5%
<b>MRAC-MIT</b>	R	R	D	D
<b>MRAC-LYA</b>	R	R	D	D
<b>VS-MRAC-SAT-PI</b>	FT	D	R	R

D = Degraded System, FT = Fault Tolerant, R = Robust

**Table A- II. MSE of the different designed schemes**

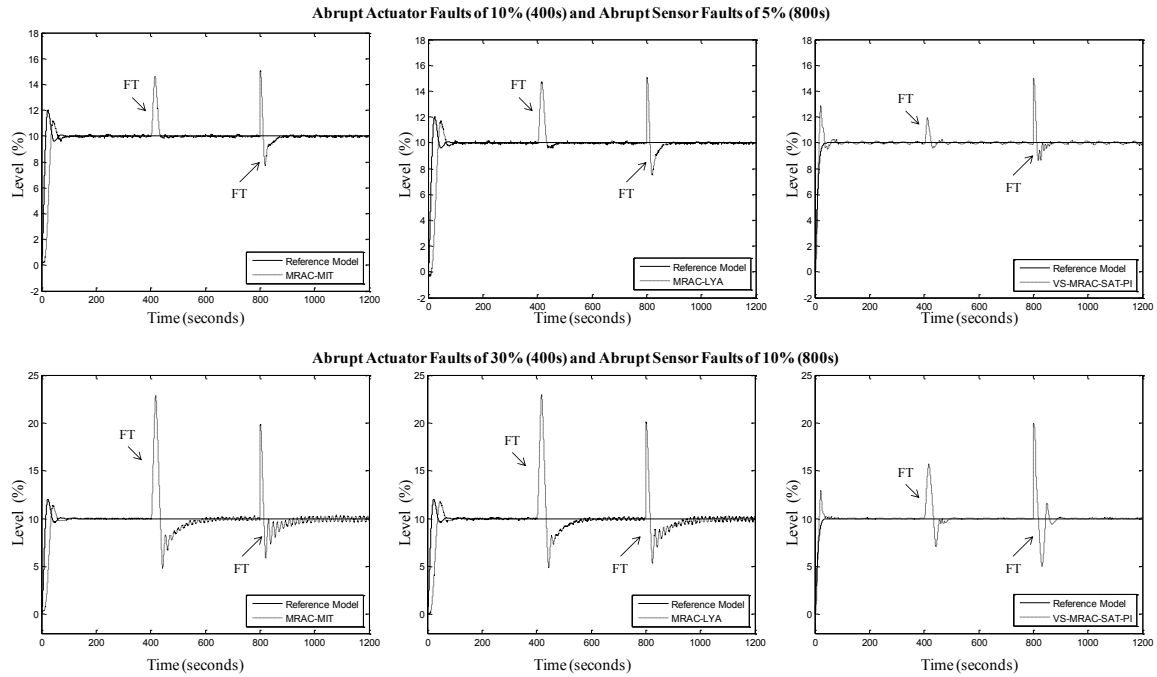
Fault Type		Fault Magnitude	MRAC-MIT	MRAC-LYA	VS-MRAC-SAT-PI
<b>Abrupt</b>	Sensor	5%	0	0	0.00068
		15%	0	0	1.33095
	Actuator	1%	0.00077	0.00068	0
		5%	0.00728	0.00951	0
<b>Gradual</b>	Sensor	5%	0	0	0.00068
		15%	0	0	1.03483
	Actuator	1%	0.00069	0.00068	0
		5%	0.00692	0.00951	0

#### *Results for the Physical Coupled-Tank System*

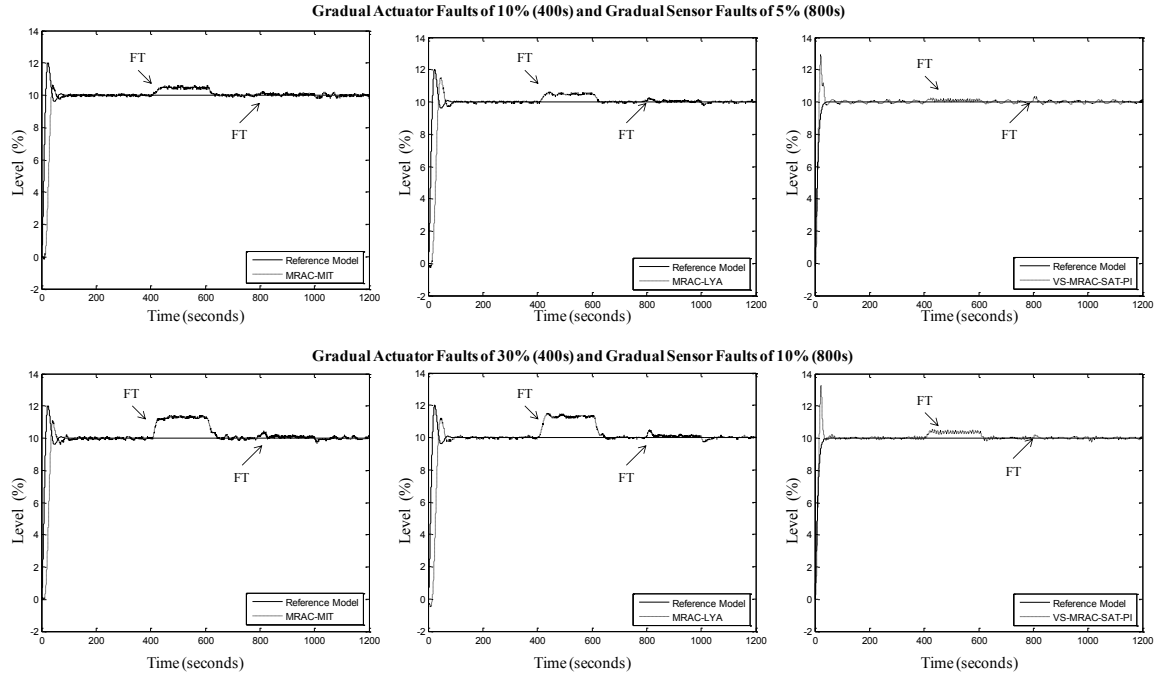
To test and compare the proposed schemes in the Coupled-Tank System, the same operating conditions were applied for the three schemes (MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI. To test the schemes, the system was already stabilized at the 50% of water level of the total capacity of both tanks. Then a 10% of change in reference is performed. The faults are introduced after the stabilization of the system. The total test time is of 1200 seconds, the actuator fault was introduced at 400 seconds and the sensor fault was introduced at 800 seconds. For actuator faults two different faults magnitudes were proved (10% and 30%). On the other hand, for sensor faults the magnitude of the faults were chosen as 5% and 10%. In addition the MSE error was calculated for each of the above experiments (see Table A-V).

First, the combination of abrupt actuator fault of magnitude 10% and abrupt sensor fault of magnitude 5% with the combination of abrupt actuator fault of magnitude 30% and abrupt sensor fault of magnitude 10% were tested (see Figure A. 12). The three FTC schemes were proved in the real Coupled-Tank system station. The value of  $\gamma$  for the MRAC-MIT and the MRAC-LYA was selected as 0.001.

Second, the combination of gradual actuator fault of magnitude 10% and gradual sensor fault of magnitude 5% with the combination of gradual actuator fault of magnitude 30% and gradual sensor fault of magnitude 10% were tested (see Figure A. 13). The three FTC schemes were proved in the real Coupled-Tank system station. The value of  $\gamma$  for the MRAC-MIT and the MRAC-LYA was selected as 0.001.



**Figure A. 12. Abrupt Sensor and Abrupt Actuator Faults in the MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI schemes.**



**Figure A. 13. Abrupt Sensor and Abrupt Actuator Faults in the MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI schemes.**

The summary of the results of the above figures can be observed in Table A-III and Table A-IV.

**Table A- III. Summary of results of Abrupt Faults**

Proposed Schemes	Abrupt Faults			
	Fault Combination 1		Fault combination 2	
	Actuator Fault 10%	Sensor Fault 5 %	Actuator Fault 30 %	Sensor Fault 10 %
<b>MRAC-MIT</b>	FT	FT	FT	FT
<b>MRAC-LYA</b>	FT	FT	FT	FT
<b>VS-MRAC-SAT-PI</b>	FT	FT	FT	FT

FT=Fault Tolerant

**Table A- IV. Summary of results of Gradual Faults**

Proposed Schemes	Gradual Faults			
	Fault Combination 1		Fault combination 2	
	Actuator Fault 10%	Sensor Fault 5 %	Actuator Fault 30 %	Sensor Fault 10 %
<b>MRAC-MIT</b>	FT	FT	FT	FT
<b>MRAC-LYA</b>	FT	FT	FT	FT
<b>VS-MRAC-SAT-PI</b>	FT	FT	FT	FT

FT=Fault Tolerant

It can be observed from Figures A. 12 and Figure A. 13 and from Table A-III and A-IV that for abrupt faults the three proposed schemes (MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI) resulted fault tolerant to the two different combinations of faults tested in this research. Also, for gradual faults the three schemes were fault tolerant to actuator faults and to sensor faults, no matter the fault combination magnitude.

Table A-V presents the result of the MSE from the above combination of actuator fault and sensor faults.

**Table A- V. MSE of the different designed schemes**

Fault Type	Actuator Fault Magnitude	Sensor Fault Magnitude	MRAC-MIT		MRAC-LYA		VS-MRAC-SAT-PI	
			MSE Actuator Fault	MSE Sensor Fault	MSE Actuator Fault	MSE Sensor Fault	MSE Actuator Fault	MSE Sensor Fault
Abrupt	10 %	5 %	0.5840	0.4690	0.6581	0.5354	0.0808	0.4054
	30 %	10 %	5.8530	2.0780	5.7538	2.4842	1.2220	2.5371
Gradual	10 %	5 %	0.1029	0.0059	0.1029	0.0047	0.0091	0.0072
	30 %	10 %	0.7477	0.0114	0.7673	0.0125	0.0434	0.0091

Table A-V corroborate the results showed from Figure A. 12 and A. 13 and the results of Table A-III and A-IV, in which in general the best designed scheme for abrupt and gradual actuator faults of 10% and 30% and for abrupt sensor fault of 5% and gradual sensor fault of 10% was the VS-MRAC-SAT-PI because has the lower MSE in the mentioned cases. On the other hand, for the abrupt sensor fault of 10% the best scheme is the MRAC-MIT and for gradual sensor fault of 5% the best scheme is the MRAC-LYA because have the lower MSE.

## Conclusions

For the experiments implemented in the Industrial Heat Exchanger, it can be observed that the MRAC-MIT and the MRAC-LYA schemes were the best schemes for abrupt sensor faults of 5% and 15% because both schemes were robust against these types of fault. On the other hand the VS-MRAC-SAT-PI scheme was fault tolerant for the abrupt sensor fault of 5% but was degraded for abrupt sensor fault of 15%. In the case of actuator faults, the MRAC-MIT and the MRAC-LYA schemes became degraded after the occurrence of the fault of 1% and 5%, while the VS-MRAC-SAT-PI scheme was robust for both fault magnitudes (abrupt actuator faults of 1% and 5%). In general, the same results were obtained for gradual faults. The MSE results (Table A-II) corroborate the above information, in which in general the best designed schemes for abrupt or gradual sensor faults were the MRAC-MIT and the MRAC-LYA schemes. And, the best designed scheme for abrupt or gradual actuator faults was the VS-MRAC-SAT-PI scheme

For the experiments implemented in the Coupled-Tank system, it can be observed that for abrupt faults the three proposed schemes (MRAC-MIT, MRAC-LYA and VS-MRAC-SAT-PI) were fault tolerant to the two different combinations of faults tested in this research (actuator faults of 10% and 30% and sensor faults of 5% and 10%). Also, for Gradual faults the three schemes were fault tolerant to actuator faults and to

sensor faults, no matter the fault combination magnitude. But, the MSE results (Table A-V) shows that in general the best designed scheme for abrupt and gradual actuator faults of 10% and 30% and for abrupt sensor fault of 5% and gradual sensor fault of 10% was the VS-MRAC-SAT-PI. And, for the abrupt sensor fault of 10% the best scheme is the MRAC-MIT and for gradual sensor fault of 5% the best scheme is the MRAC-LYA.

In summary, the controllers presented in this work allow the system availability in spite of the presence of a fault, because the implemented controllers were able to accommodate a fault between certain magnitude thresholds.



## Appendix B

First, this section explains the single multiplicative faults tested in the LPV system. The results of these experiments are shown from Table B-I and Table B-II. These results explain if the methodologies are robust, fault tolerant or degraded against the simulated fault and also demonstrate the Mean Square Error (MSE).

**Table B-I. Results of experiments using multiplicative sensor faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for LPV systems.**

Methodology	Sensor Faults Result and MSE			
	100%	90%	50%	5%
<b>MRAC-LPV based on MIT rule</b>	Fault Tolerant MSE=0.00133	Fault Tolerant MSE=0.00122	Fault Tolerant MSE=0.00082	Fault Tolerant MSE=0.00033
<b>MRAC-LPV based on Lyapunov</b>	Fault Tolerant MSE=0.00092	Fault Tolerant MSE=0.00083	Fault Tolerant MSE=0.00049	Fault Tolerant MSE=0.00022
<b>MRAC-<math>H_\infty</math>GS-LPV based on MIT rule</b>	Robust MSE=0.00122	Robust MSE=0.00122	Robust MSE=0.00122	Robust MSE=0.00122
<b>MRAC-<math>H_\infty</math>GS-LPV based on Lyapunov</b>	Robust MSE=0.00094	Robust MSE=0.00094	Robust MSE=0.00094	Robust MSE=0.00094

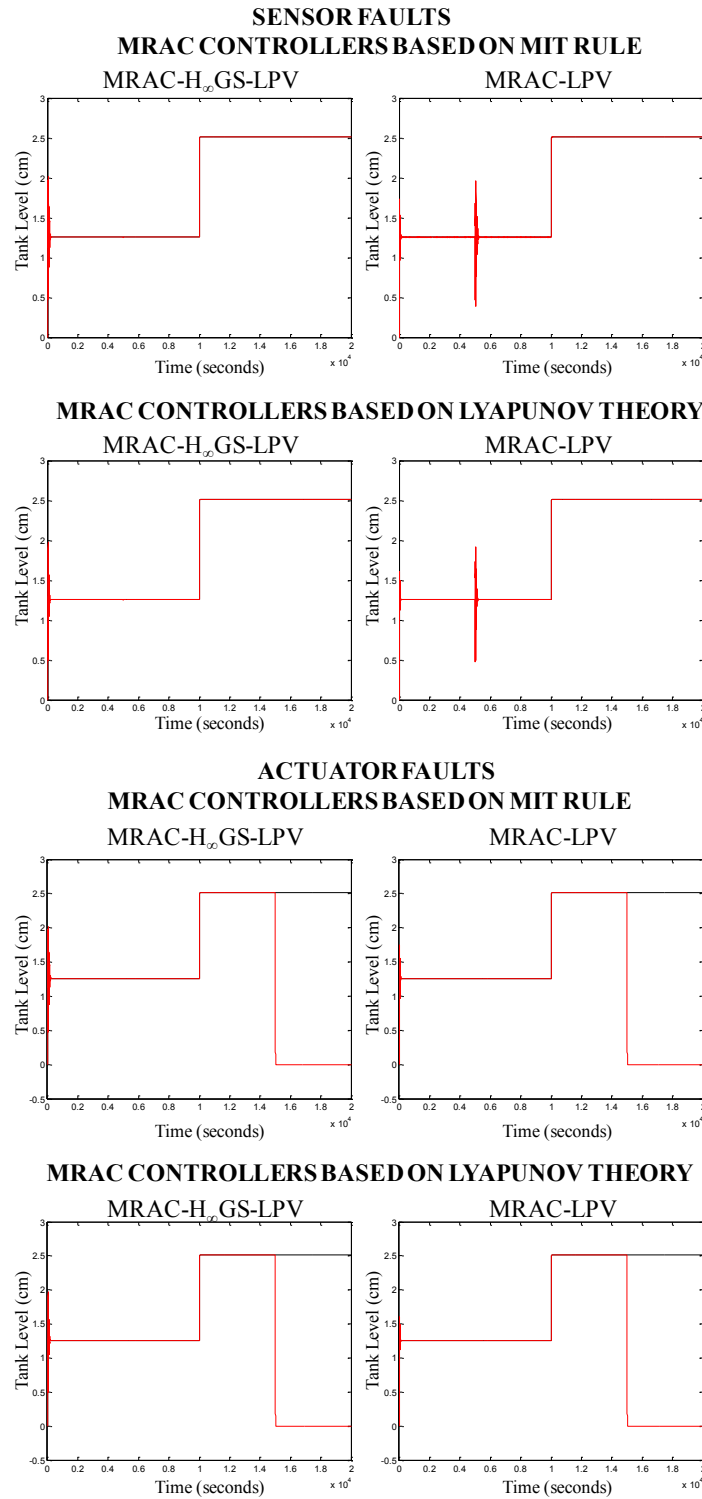
In Table B-I, it is observed that the best methodologies are the MRAC-LPV and the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory, because they are Fault Tolerant and Robust, respectively, against the sensor fault implemented at 5000 seconds and have the smaller Mean Square Error.

**Table B-II. Results of experiments using multiplicative actuator faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for LPV systems.**

Methodology	MSE Actuator Faults			
	100%	90%	50%	5%
<b>MRAC-LPV based on MIT rule</b>	Degraded MSE=1.57069	Degraded MSE=1.27232	Degraded MSE=0.39292	Degraded MSE=0.00425
<b>MRAC-LPV based on Lyapunov</b>	Degraded MSE=1.57058	Degraded MSE=1.27221	Degraded MSE=0.39281	Degraded MSE=0.00415
<b>MRAC-<math>H_\infty</math>GS-LPV based on MIT rule</b>	Degraded MSE=1.57158	Degraded MSE=1.27321	Degraded MSE=0.39381	Degraded MSE=0.00514
<b>MRAC-<math>H_\infty</math>GS-LPV based on Lyapunov</b>	Degraded MSE=1.57130	Degraded MSE=1.27293	Degraded MSE=0.39353	Degraded MSE=0.00486

In Table B-II, it is observed that for actuator faults the four methodologies could not accommodate the fault. The actuator fault was introduced at time 15000 seconds. In Figure B. 1, Figure B. 2, Figure B. 3 and Figure B. 4, it can be observe that for the multiplicative faults of magnitude 100%, 90%, 50% and 5% applied in the sensor, the MRAC- $H_\infty$ GS-LPV controller was robust against this type of fault. The above apply for the methodology based on the MIT rule and the methodology based on the Lyapunov theory. In addition, the MRAC-LPV controller was able to accommodate the multiplicative sensor faults of the different fault magnitudes. On the other hand, for the multiplicative actuator faults of magnitude 100%, 90%, 50% and 5% both controllers (MRAC- $H_\infty$ GS-LPV and MRAC-LPV) became degraded after the occurrence of the fault.

Also, it can be observe that in both cases (multiplicative sensor or multiplicative actuator faults) the controllers were able to have a change in the operating point at 10000 seconds.



**Figure B. 1. Comparison between the system output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 100% (multiplicative**

sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

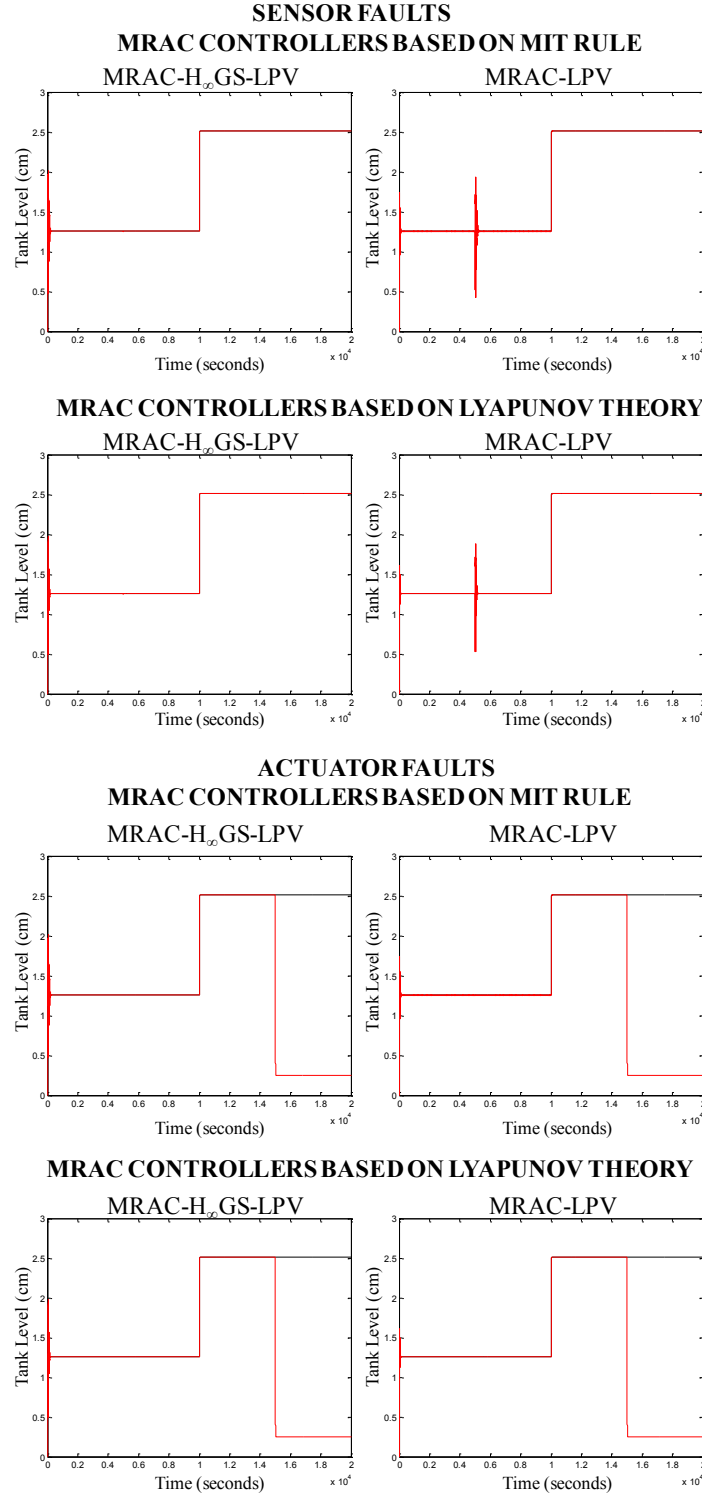


Figure B. 2. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 90% (multiplicative

sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

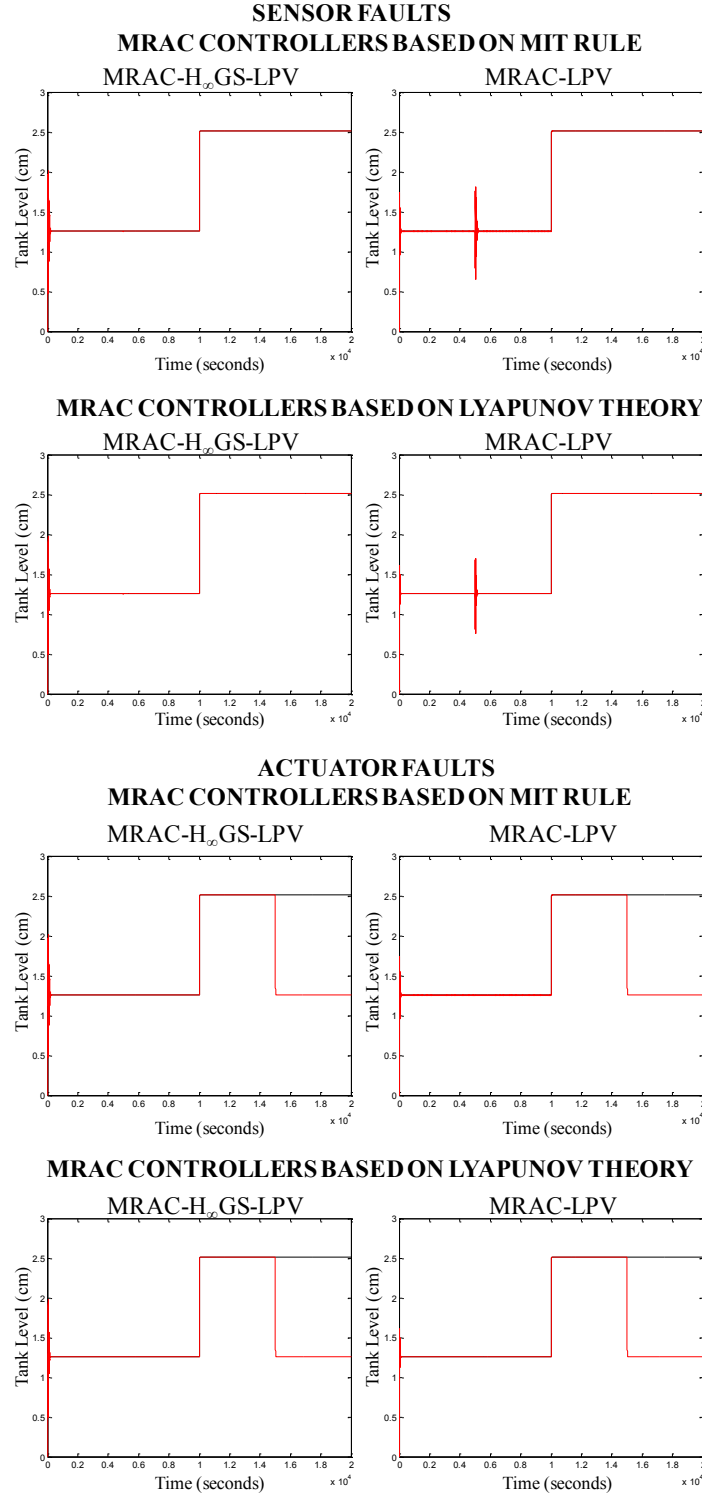


Figure B. 3. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 50% (multiplicative

sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

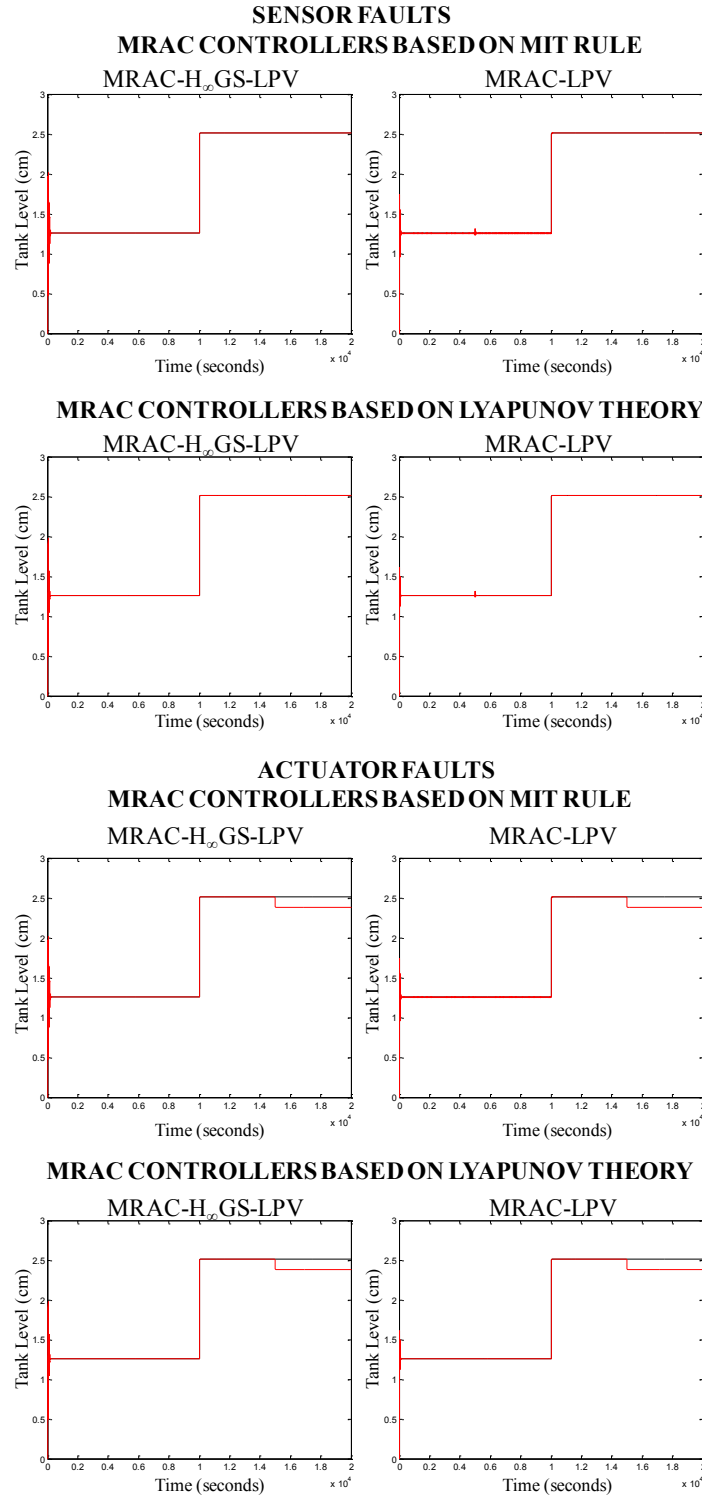
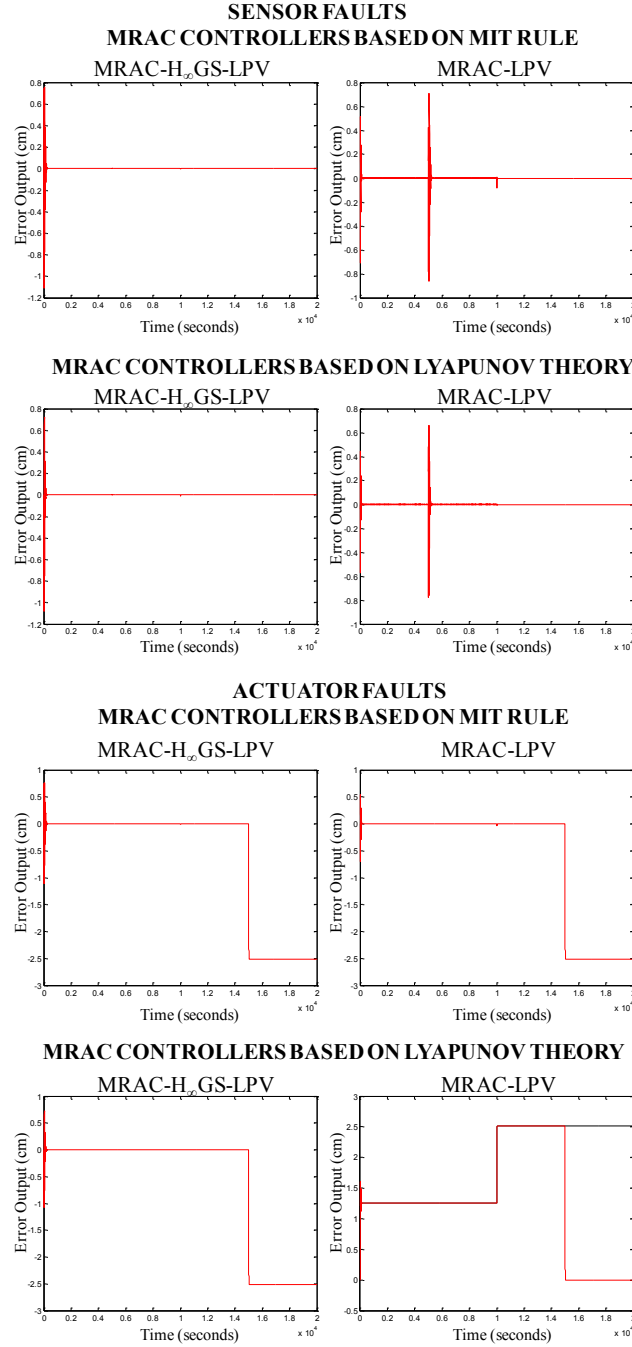


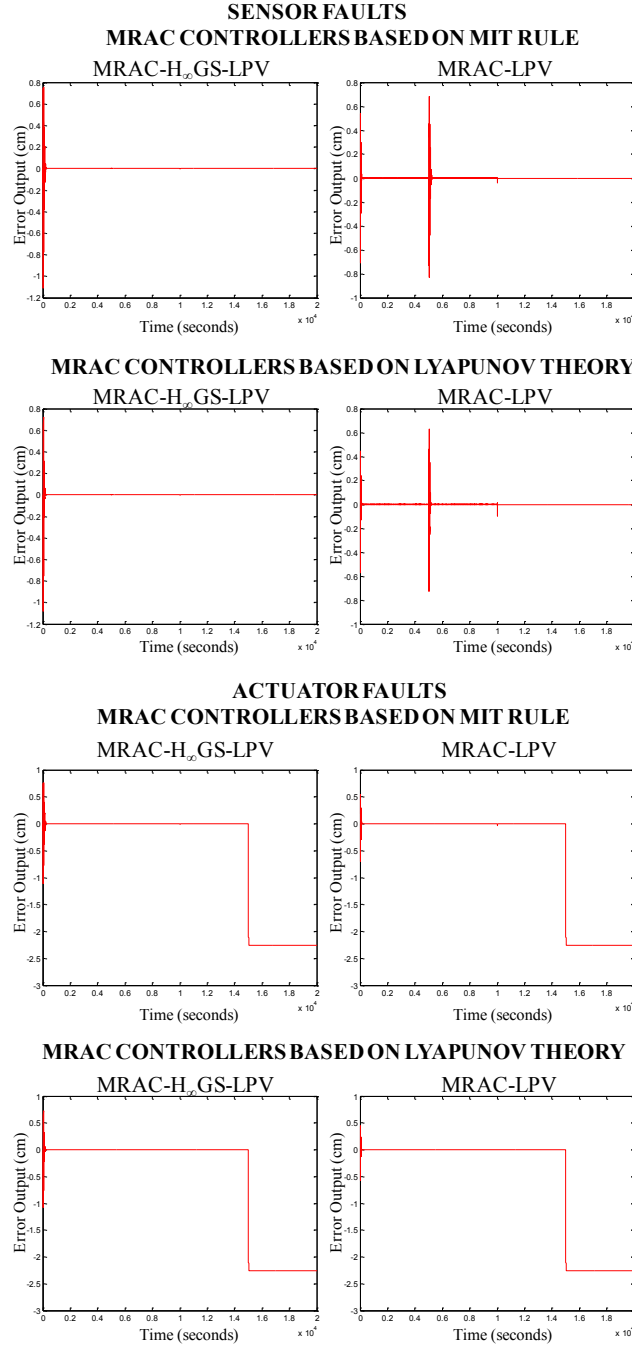
Figure B. 4. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of magnitude 5%

(multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

In addition to the experiments presented in this section, the error of these experiments was calculated and plotted, see the next figures.



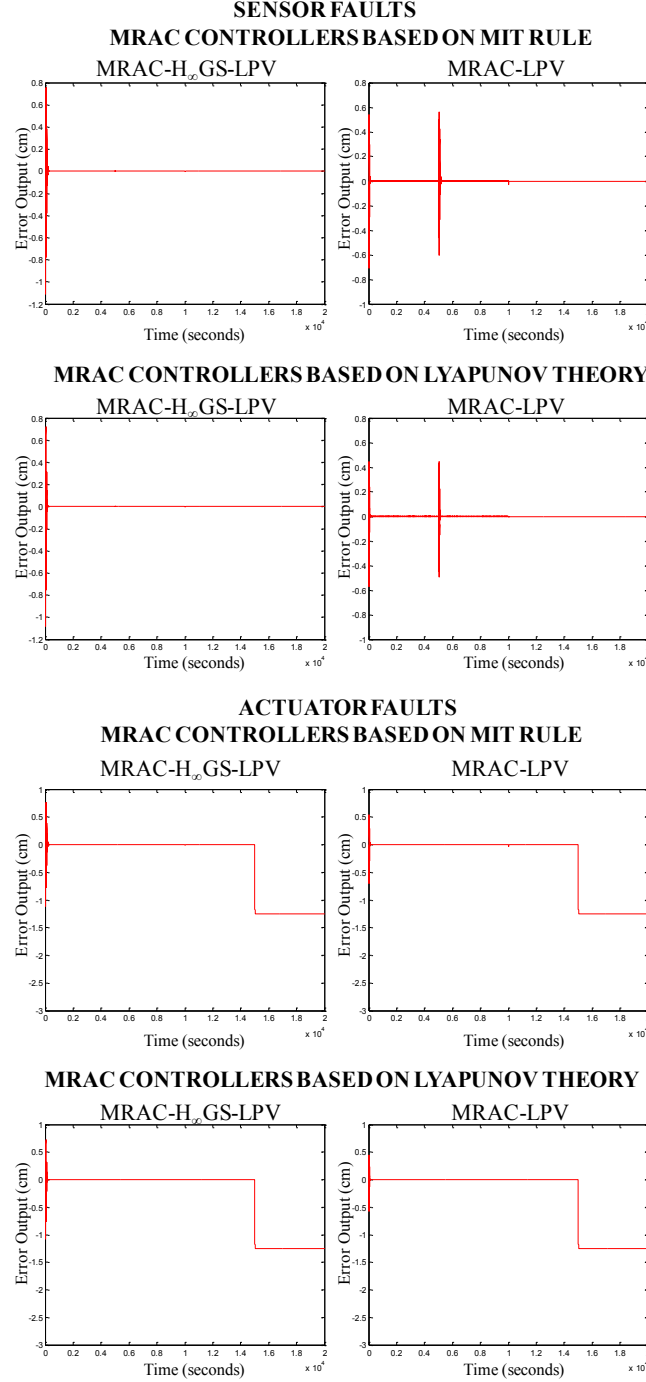
**Figure B. 5. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of magnitude 100% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.**



**Figure B. 6. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of magnitude 90% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.**

In Figure B. 5 and Figure B. 6, it can be observe that for the single multiplicative sensor fault of 100% and 90% the error range between +/- 1 cm from the set point. On the other hand for the single multiplicative actuator fault of magnitude 100% and 90% the error represents approximately 2.5 cm of the system output.

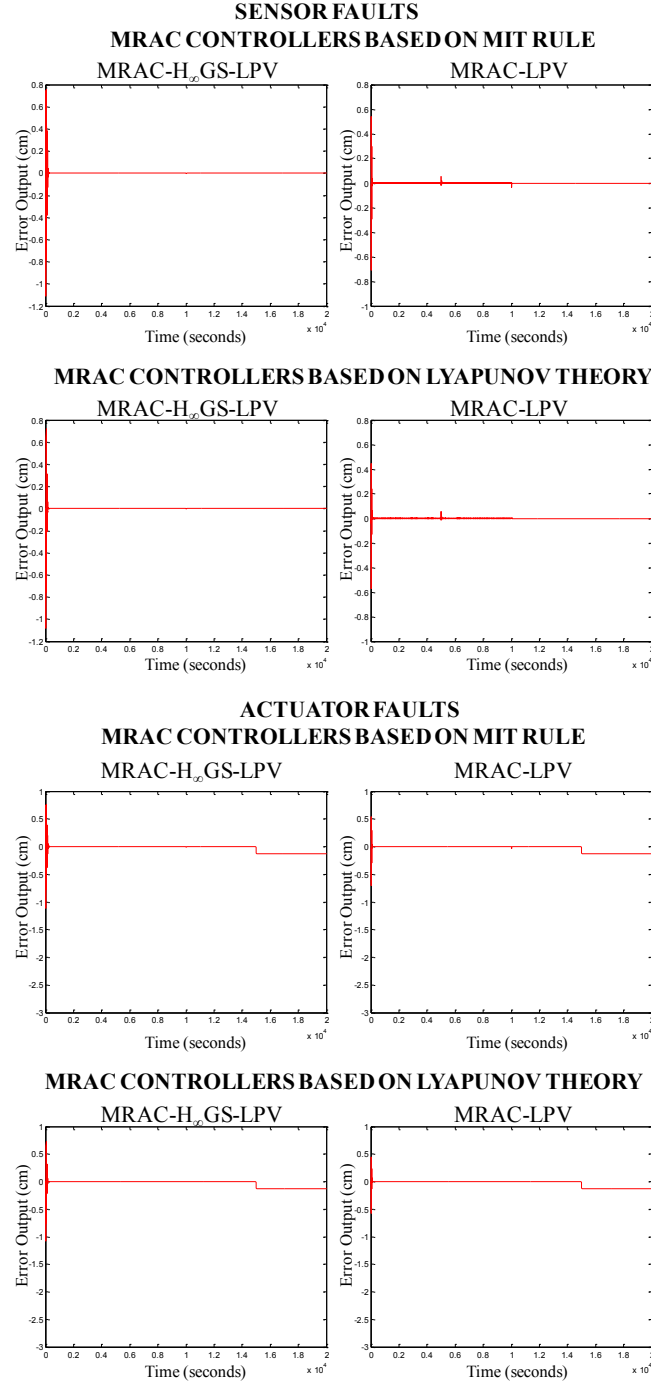
In figure B. 7, it can be observe that for the single multiplicative sensor fault of 50% the error range between  $\pm 1$  cm from the set point. On the other hand for the single multiplicative actuator fault of 50% the error represents approximately 1.5 cm of the system output.



**Figure B. 7. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of magnitude 50% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.**



In figure B. 8, it can be observe that for the single multiplicative sensor fault of magnitude 5% the error range between +/- 1 cm from the set point. On the other hand for the single multiplicative actuator fault of magnitude 5% the error represents approximately 1 cm of the system output.

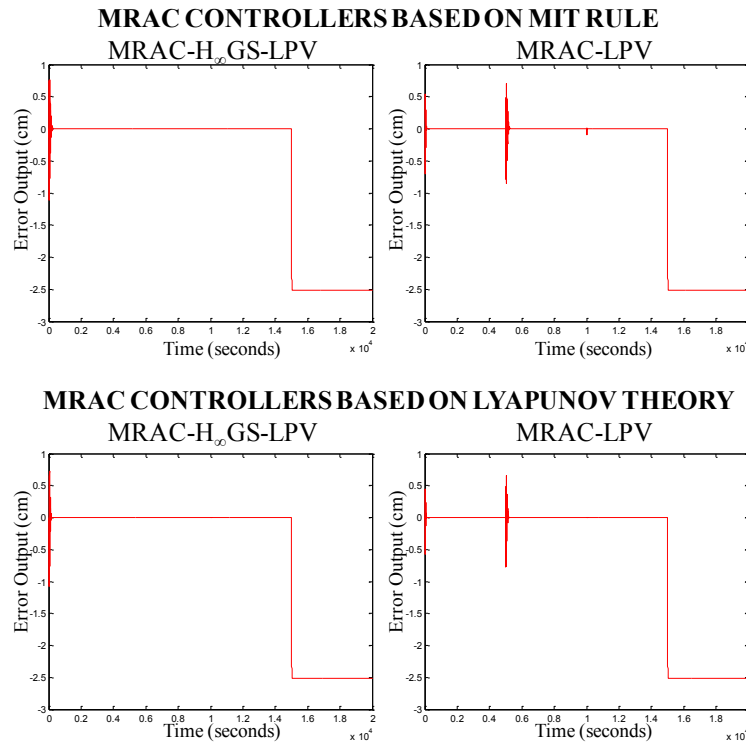


**Figure B. 8. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 5% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.**

In Figure B. 9 and Figure B. 10, for the combination of a multiplicative sensor fault and a multiplicative actuator fault both of 100% and 90%, the error represents approximately 2.5 cm of the system output.

In figure B. 11, for the combination of a multiplicative sensor fault and a multiplicative actuator fault both of 50%, the error represents approximately 1.5 cm of the system output.

In figure B. 12, for the combination of a multiplicative sensor fault and a multiplicative actuator fault both of 5%, the error represents approximately 1 cm of the system output.



**Figure B. 9. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 100%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.**

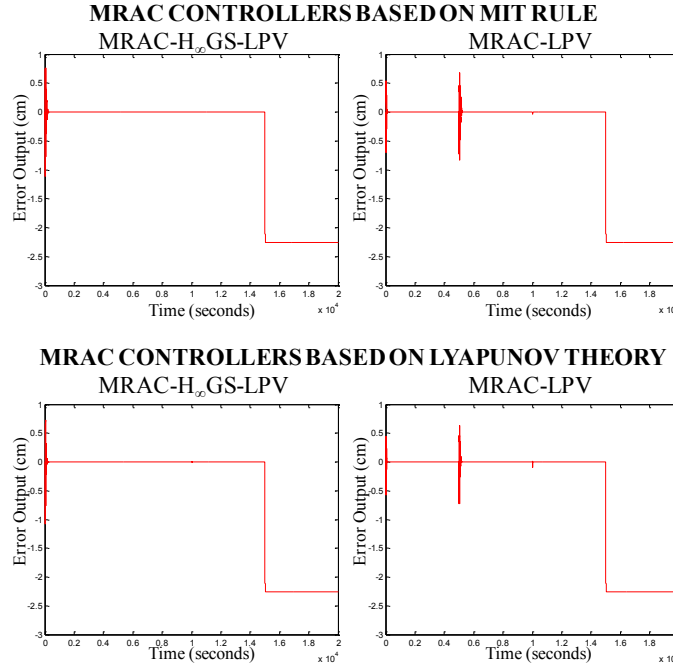


Figure B. 10. Comparison between the error output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

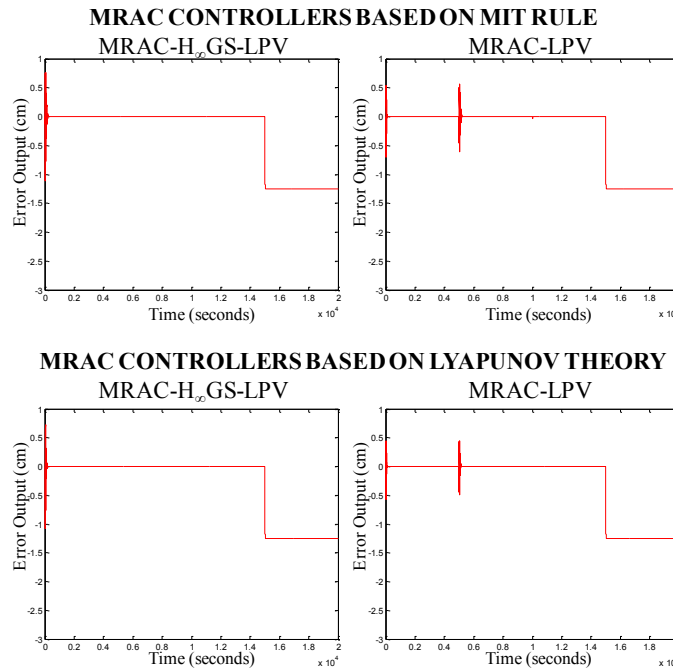


Figure B. 11. Comparison between the error output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%.

50%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

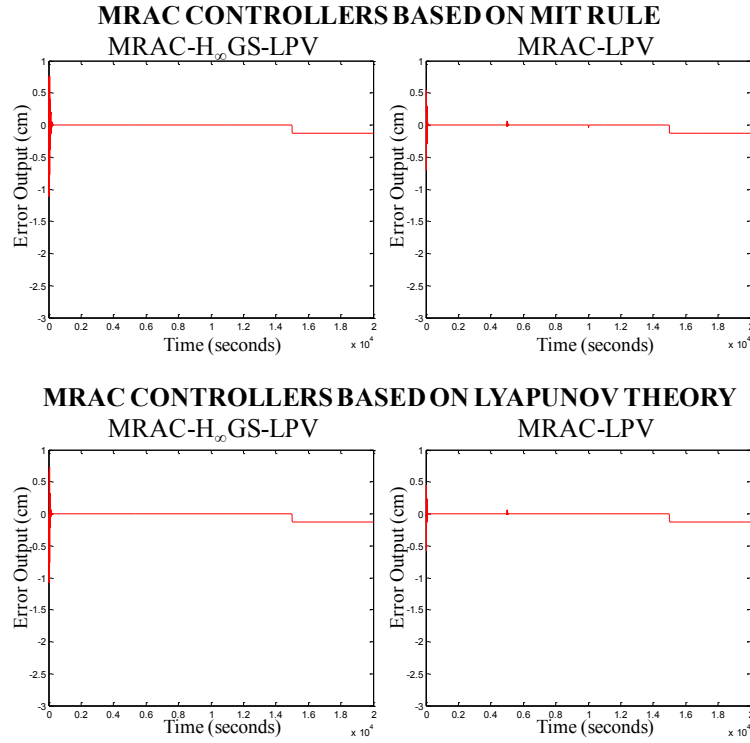
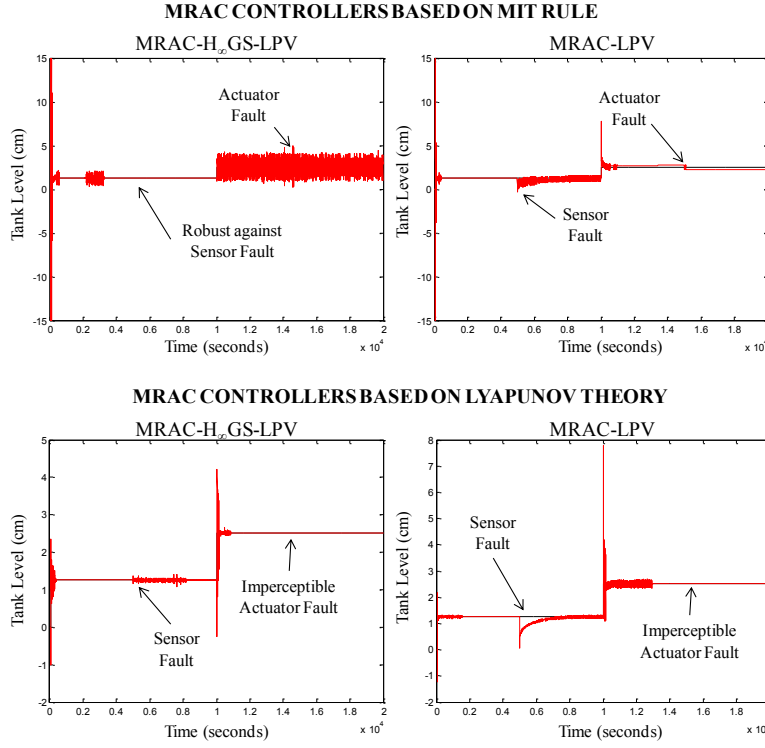


Figure B. 12. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the LPV system.

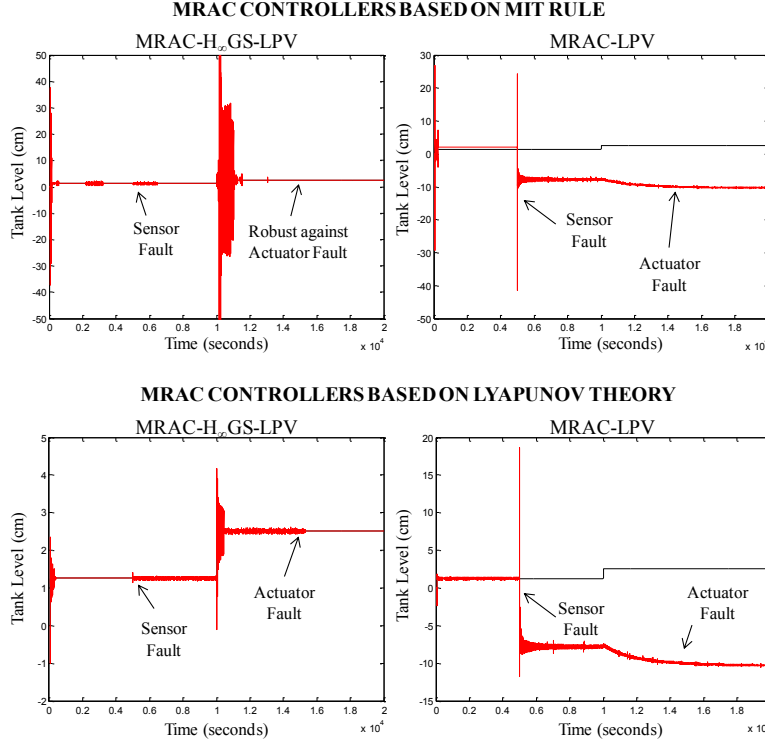
## Appendix C

The next figures show the results of the nonlinear model implementation using a value of  $\gamma = 10000$  with additive faults. These figure presented chattering and oscillation in comparison with the experiments when a value of  $\gamma = 0.003$  was used.



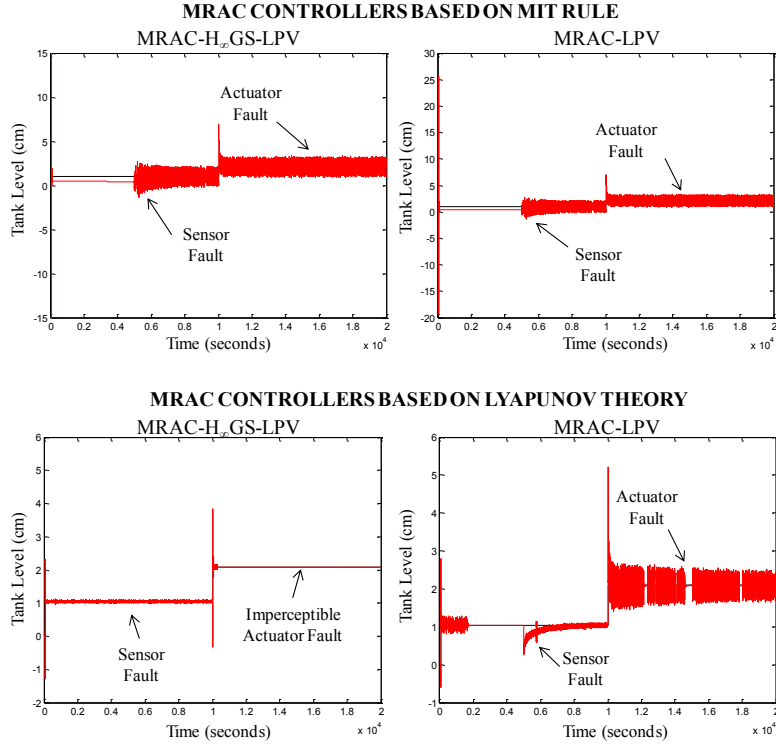
**Figure C. 1 . Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with an abrupt-sensor fault of 3% and an abrupt-actuator fault of 20% for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$ .**

In Figure C.1, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\phi_1=0.3$  and  $\phi_2=0.5$ , an abrupt-sensor fault of 3% was introduced at time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC- $H_\infty$ GS-LPV based on MIT rule was robust against the sensor fault but the systems remained with oscillations after the change in the operating point. The MRAC- $H_\infty$ GS-LPV based on Lyapunov theory was fault tolerant against the sensor fault, could tolerate the change in the operating point with certain oscillations and the actuator fault was imperceptible. On the other hand, the MRAC-LPV scheme based on MIT rule presented certain oscillations after the occurrence of the sensor fault, and was able to accommodate the fault after the occurrence of the actuator fault. Finally, the MRAC-LPV scheme based on Lyapunov theory presented certain oscillations after the occurrence of the sensor fault but was able to accommodate the fault after the change in the operating point and the actuator fault was imperceptible.



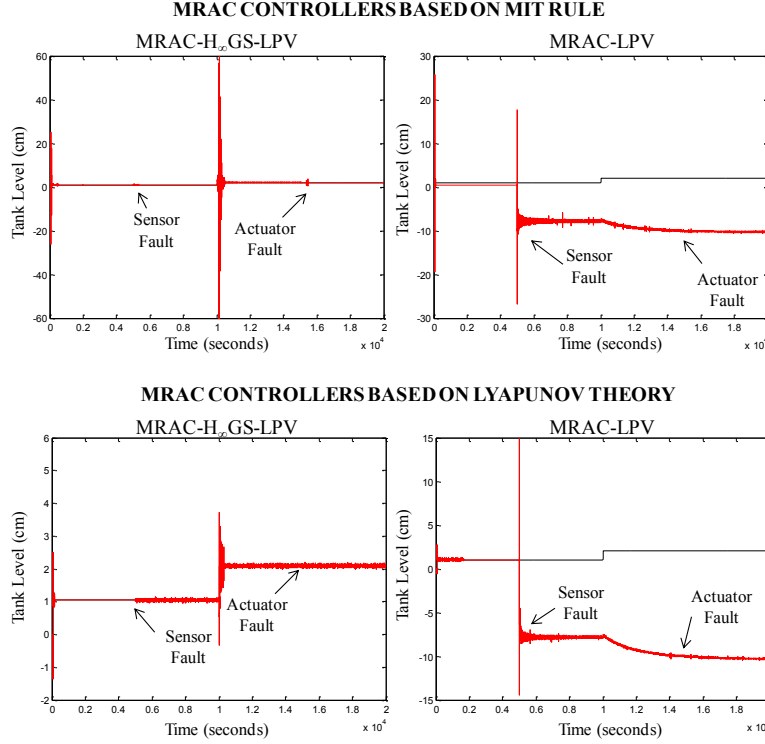
**Figure C. 2. Comparison between the Nonlinear Process MRAC-H<sub>∞</sub>GS-LPV and the Nonlinear Process MRAC-LPV Controllers with an abrupt-sensor fault of 160% and an abrupt-actuator fault of 20% for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  .**

In Figure C.2, the MRAC-H<sub>∞</sub>GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\phi_1=0.3$  and  $\phi_2=0.5$ , an abrupt-sensor fault of 160% was introduced in time 5000 seconds and an abrupt-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC-H<sub>∞</sub>GS-LPV based on MIT rule became unfeasible after the change in the operating point because it reached a value of +/- 60 and the maximum real physical value of the system output is 30. In the MRAC-H<sub>∞</sub>GS-LPV based on Lyapunov theory after the occurrence of the sensor fault the system presented a small oscillation that was corrected after the occurrence of the actuator fault. On the other hand, the MRAC-LPV scheme based on MIT rule and Lyapunov theory became degraded after the occurrence of the sensor fault.



**Figure C. 3. Comparison between the Nonlinear Process MRAC- $H_\infty$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with a gradual-sensor fault of 3% and a gradual-actuator fault of 20% for the operating points  $\phi_1=0.6$  and  $\phi_2=0.6$  .**

In Figure C.3, the MRAC- $H_\infty$ GS-LPV and the MRAC-LPV Controller based on MIT rule and on Lyapunov theory are compared. While both controllers are working in the operating point  $\phi_1=0.6$  and  $\phi_2=0.6$ , a gradual-sensor fault of 3% was introduced in time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC- $H_\infty$ GS-LPV based on MIT rule, the MRAC-LPV based on the MIT rule and based on Lyapunov theory schemas remained with oscillations problems after the occurrence of the sensor fault until the end of the simulation. On the other hand, the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory presented a small oscillation, since the beginning and during the sensor fault occurrence, almost unnoticed that is accommodated after the change in the operation point and the actuator fault was imperceptible.



**Figure C. 4. Comparison between the Nonlinear Process MRAC- $H_{\infty}$ GS-LPV and the Nonlinear Process MRAC-LPV Controllers with a gradual-sensor fault of 160% and a gradual-actuator fault of 20% for the operating points  $\phi_1=0.6$  and  $\phi_2=0.6$  .**

In Figure C.4, the MRAC- $H_{\infty}$ GS-LPV and the MRAC-LPV Controller based on the MIT rule and on the Lyapunov theory are compared. While both controllers are working in the operating point  $\phi_1=0.6$  and  $\phi_2=0.6$ , a gradual-sensor fault of 160% was introduced in time 5000 seconds and a gradual-actuator fault of 20% was introduced at time 15000 seconds. In addition, a change in the operating point was performed at time 10000 seconds. In this figure, it can be observed that the MRAC- $H_{\infty}$ GS-LPV based on MIT rule became unfeasible after the change in the operating point. The MRAC- $H_{\infty}$ GS-LPV based on Lyapunov theory after the occurrence of the sensor fault presented a small oscillation until the end of the simulation. On the other hand, the MRAC-LPV scheme based on MIT rule and Lyapunov theory became degraded after the occurrence of the sensor fault.

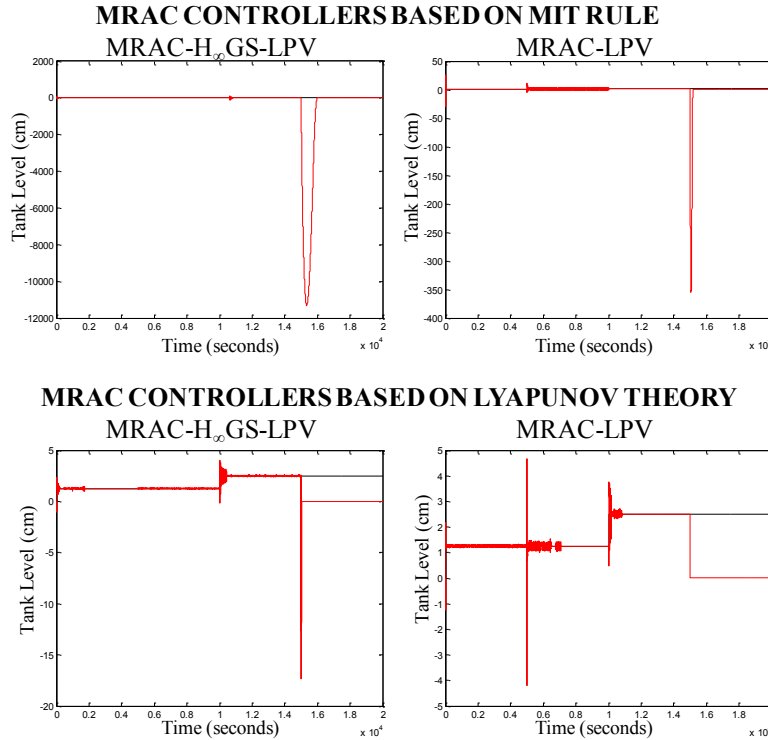
In addition, to compare the Nonlinear Process MRAC MIT and the Nonlinear Process MRAC Lyapunov based design the Mean Square Error (MSE) was calculated for all the experiments. The results are shown in Table C-I.



**Table C- I. MSE Results of the comparison between the Nonlinear MRAC-LPV and the Nonlinear MRAC- $H_\infty$ GS-LPV MIT and Lyapunov based design.**

Operating Point $\phi_1=0.3$ and $\phi_2=0.5$		
Methodology	MRAC MIT based design	MRAC Lyapunov based design
	Sensor Faults (3%) and Actuator Faults (20%)	Sensor Faults (3%) and Actuator Faults (20%)
MRAC-LPV	0.462414	0.022708
MRAC- $H_\infty$ GS-LPV	1.798773	0.01143
	Sensor Faults (160%) and Actuator Faults (20%)	Sensor Faults (160%) and Actuator Faults (20%)
MRAC-LPV	96.72275	95.6898
MRAC- $H_\infty$ GS-LPV	34.61832	0.013007
Operating Point $\phi_1=0.6$ and $\phi_2=0.6$		
Methodology	MRAC MIT based design	MRAC Lyapunov based design
	Sensor Faults (3%) and Actuator Faults (20%)	Sensor Faults (3%) and Actuator Faults (20%)
MRAC-LPV	0.659136	0.039946
MRAC- $H_\infty$ GS-LPV	1.238969	0.010449
	Sensor Faults (160%) and Actuator Faults (20%)	Sensor Faults (160%) and Actuator Faults (20%)
MRAC-LPV	90.21047	89.67406
MRAC- $H_\infty$ GS-LPV	8.678645	0.008226

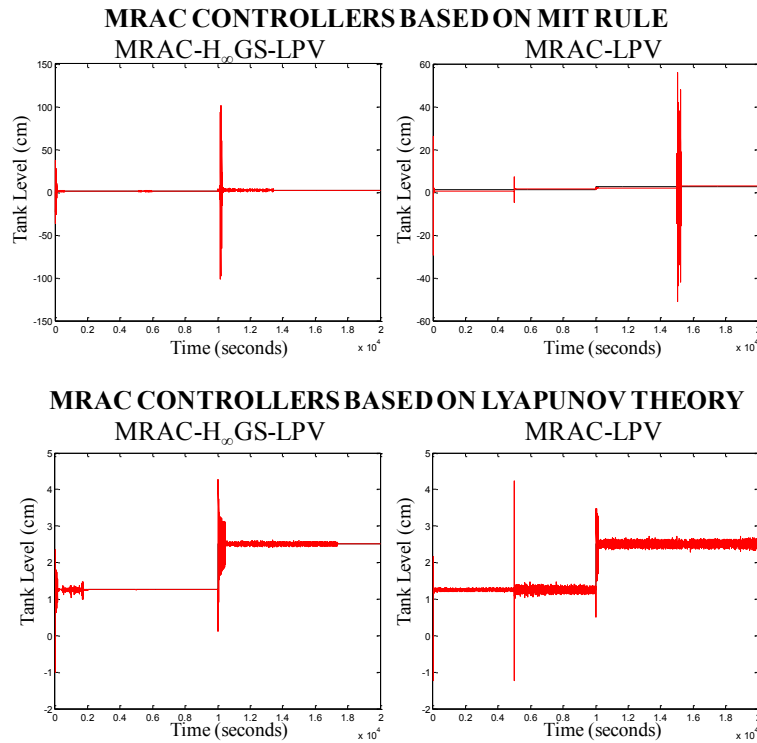
On the other hand, the following figures show the results of the nonlinear model implementation of multiplicative faults using a value of  $\gamma = 10000$ :



**Figure C. 5. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of**

100%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.

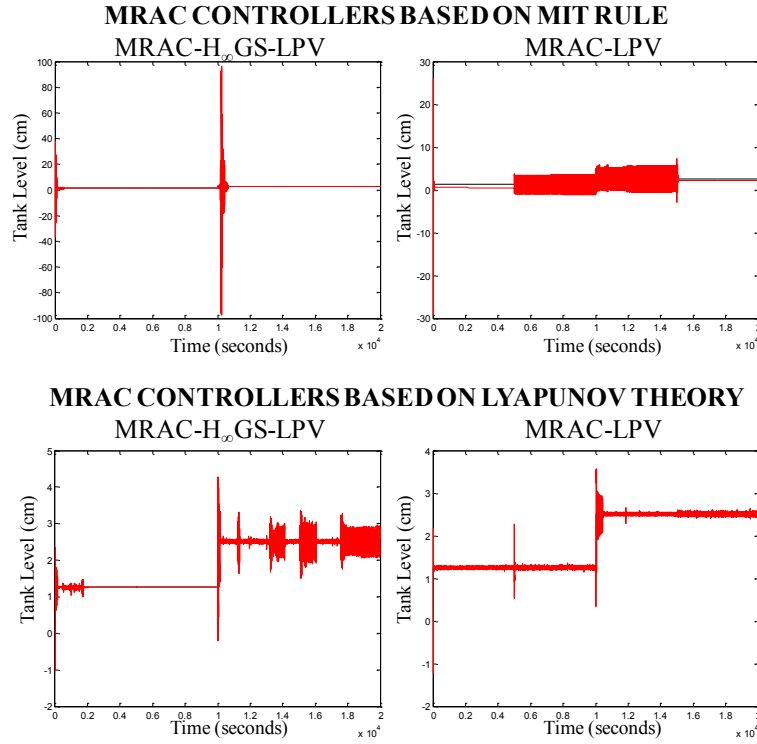
In Figure C.5, for the combination of multiplicative sensor fault of 100% at 5000 second and multiplicative actuator fault of 100% at 15000 seconds, it can be observe that MRAC- $H_\infty$ GS-LPV based on the MIT rule was robust to the sensor fault but became unfeasible after occurrence of the actuator fault. The same applies for the MRAC-LPV based on the MIT rule. On the other hand, the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, but became degraded after the appearance of the actuator fault. The MRAC-LPV based on Lyapunov theory was fault tolerant to the multiplicative sensor fault but also became degraded after the occurrence of the actuator fault.



**Figure C. 6. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

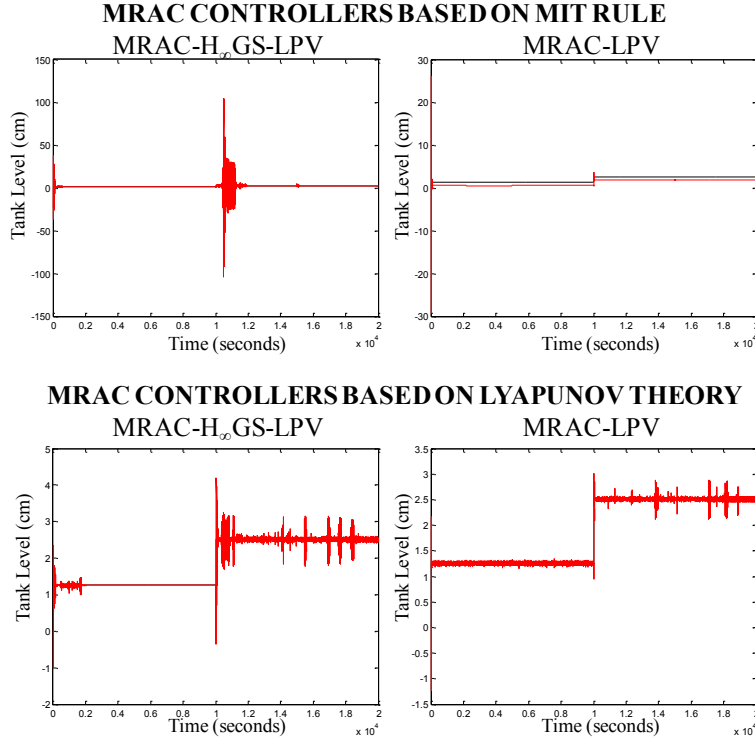
In Figure C.6, for the combination of multiplicative sensor fault of 90% at 5000 second and multiplicative actuator fault of 90% at 15000 seconds, it can be observe that MRAC- $H_\infty$ GS-LPV based on the MIT rule was fault tolerant to the sensor fault but became unfeasible after the change in the operating point. The MRAC-LPV based on the MIT rule presented and offset since the beginning of the simulation, was fault tolerant to the sensor fault but became unfeasible after the occurrence of the actuator fault. On the other hand, the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, presented some chattering after the change in the operating point and was fault tolerant to the actuator fault. The MRAC-LPV

based on Lyapunov theory presented some chattering since the beginning of the simulation and was fault tolerant to the multiplicative sensor and actuator fault.



**Figure C. 7. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 50%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure C.7, for the combination of multiplicative sensor fault of 50% at 5000 second and multiplicative actuator fault of 50% at 15000 seconds, it can be observe that MRAC- $H_\infty$ GS-LPV based on the MIT rule was fault tolerant to the sensor fault but became unfeasible after the change in the operating point. The MRAC-LPV based on the MIT rule presented and offset since the beginning of the simulation, and presented chattering problems after the sensor fault. On the other hand, the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, presented some chattering after the change in the operating point and was fault tolerant to the actuator fault. The MRAC-LPV based on Lyapunov theory presented some chattering since the beginning of the simulation and was fault tolerant to the multiplicative sensor and actuator fault.



**Figure C. 8. Comparison between the system output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure C.8, for the combination of multiplicative sensor fault of 5% at 5000 second and multiplicative actuator fault of 5% at 15000 seconds, it can be observe that MRAC- $H_\infty$ GS-LPV based on the MIT rule was fault tolerant to the sensor fault but became unfeasible after the change in the operating point. The MRAC-LPV based on the MIT rule presented and offset since the beginning. On the other hand, the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, presented some chattering after the change in the operating point and was fault tolerant to the actuator fault. The MRAC-LPV based on Lyapunov theory presented some chattering since the beginning of the simulation and was fault tolerant to the multiplicative sensor and actuator fault. In addition, the error from the multiplicative faults experiments was calculated and plotted.

Table C-II shows the results of the above multiplicative fault experiments. These results explain if the methodologies are robust, fault tolerant, degraded, and unfeasible or if they have a chattering against the simulated fault and also demonstrate the Mean Square Error (MSE). In Table 24, it is observed that just a few methodologies were able to tolerate the combination of the multiplicative sensor (5000 seconds) and actuator (15000 seconds) faults implemented in the nonlinear system, for example the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory with a sensor and an actuator fault of 90% was robust against the sensor fault and was fault tolerant against the actuator fault. In general, the MRAC-LPV and the MRAC- $H_\infty$ GS-LPV based on

Lyapunov theory presented the best behavior because the Mean Square Error was the lower in comparison with the methodologies using the MIT rule. In addition, most of the experiments presented some chattering effects due to the nonlinearities of the system.

**Table C- II. Results of experiments using multiplicative sensor (5000 seconds) and actuator (15000 seconds) faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for nonlinear systems.**

Methodology	MSE Sensor and Actuator Faults			
	100%	90%	50%	5%
<b>MRAC-LPV based on MIT rule</b>	FT, C→SF FT, UF→AF MSE=519.004	FT, O→SF UF, O→AF MSE=9.23630	O, C→SF C, O→AF MSE=2.71772	O, R→SF O, I→AF MSE=0.93032
<b>MRAC-LPV based on Lyapunov theory</b>	FT, C→SF D→AF MSE=1.58337	FT, C→SF C→AF MSE=0.00863	FT, C→SF C→AF MSE=0.00537	C→SF FT, C→AF MSE=0.00315
<b>MRAC-<math>H_\infty</math>GS-LPV based on MIT rule</b>	I→SF FT, UF→AF MSE=2912047.96	FT→SF I→AF Unfeasible MSE=18.53418	R→SF I→AF Unfeasible MSE=16.75147	R→SF FT→AF Unfeasible MSE=29.95529
<b>MRAC-<math>H_\infty</math>GS-LPV based on Lyapunov theory</b>	FT, C→SF D→AF MSE=1.94864	R→SF FT, C→AF MSE=0.01223	R→SF FT, C→AF MSE=0.03068	R→SF FT, C→AF MSE=0.01669

AF=Actuator Fault, C=Chattering, FT=Fault Tolerant, O=Offset,

R=Robust, SF=Sensor Fault, UF=Unfeasible, I=Imperceptible

In addition, single fault were tested. The results of these experiments are shown from Table C-III and Table C-IV. These results explain if the methodologies are robust, fault tolerant, degraded, and unfeasible or if they have a chattering against the simulated fault and also demonstrate the Mean Square Error (MSE).

**Table C- III. Results of experiments using multiplicative sensor faults in the MRAC-LPV and MRAC- $H_\infty$ GS-LPV methodologies based on MIT rule for nonlinear systems.**

Methodology	Sensor Faults Result and MSE			
	100%	90%	50%	5%
<b>MRAC-LPV based on MIT rule</b>	Chattering Offset MSE=1.35567	Offset MSE=0.72602	Offset Chattering MSE=4.36206	Offset MSE=0.94201
<b>MRAC-LPV based on Lyapunov theory</b>	Fault Tolerant Chattering MSE=0.01065	Fault Tolerant Chattering MSE=0.00870	Fault Tolerant Chattering MSE=0.00527	Chattering MSE=0.00238
<b>MRAC-<math>H_\infty</math>GS-LPV based on MIT rule</b>	Chattering Unfeasible MSE=16.2487	Fault Tolerant Chattering Unfeasible MSE=18.5341	Robust Chattering Unfeasible MSE=16.7514	Robust Chattering Unfeasible MSE=29.9521
<b>MRAC-<math>H_\infty</math>GS-LPV based on Lyapunov theory</b>	Chattering MSE=0.01253	Robust Chattering MSE=0.01230	Robust Chattering MSE=0.01544	Robust Chattering MSE=0.01257

In Table C-III, it is observed that the best methodologies are the MRAC-LPV and the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory, because they are Fault Tolerant and Robust, respectively, against the sensor fault implemented at 5000 seconds and have the smaller Mean Square Error. When the nonlinear system is

used as the plant it can be shown that the output signal presents some chattering effect due to the nonlinearities of the system.

**Table C- IV. Results of experiments using multiplicative actuator faults in the MRAC-LPV and MRAC-H<sub>∞</sub>GS-LPV methodologies based on MIT rule for nonlinear systems.**

Methodology	MSE Actuator Faults			
	100%	90%	50%	5%
<b>MRAC-LPV based on MIT rule</b>	Unfeasible MSE=3317.69	Offset Chattering MSE=1.20246	Offset MSE=0.93118	Offset MSE=1.04342
<b>MRAC-LPV based on Lyapunov theory</b>	Chattering Degraded MSE=1.62418	Chattering MSE=0.00327	Chattering MSE=0.00338	Chattering Fault Tolerant MSE=0.00318
<b>MRAC-H<sub>∞</sub>GS-LPV based on MIT rule</b>	Unfeasible MSE=2918.92	Chattering Unfeasible MSE=5.01029	Chattering Fault Tolerant Unfeasible MSE=13.38106	Unfeasible Chattering MSE=1.82513
<b>MRAC-H<sub>∞</sub>GS-LPV based on Lyapunov theory</b>	Chattering Degraded MSE=2.22839	Chattering Fault Tolerant MSE=0.02010	Chattering Fault Tolerant MSE=0.02013	Chattering MSE=0.02048

In Table C-IV, the actuator fault was introduced at 15000 seconds. It can be observed that some of the actuator faults implemented in the methodologies are unfeasible to the system, this mean that the results are above the physical capacities of the nonlinear system, for example, the MRAC-LPV based on MIT rule with an actuator fault of 100%. On the other hand, most of the methodologies presented some chattering effect. The best methodologies are the ones based on Lyapunov theory because they have the slower Mean Square Error.

In Figure C.9, it can be observe that for the multiplicative fault in sensors of 100% at time 5000 applied to the nonlinear system. The MRAC-H<sub>∞</sub>GS-LPV based on the MIT rule became unfeasible after the change in the operating point because it reached a system output value of +/- 100 cm and the high of the tanks that the system is representing is of 30 cm. The same applies for the MRAC-LPV based on the MIT rule. In addition both controllers presented chattering problems after the occurrence of the fault. On the other hand, in the MRAC-H<sub>∞</sub>GS-LPV based on Lyapunov after the occurrence of the multiplicative sensor fault, the system presented some reasonable chattering (+/- 0.2 cm) and could tolerate the change in the operating point. The chattering effects are due to the nonlinearities of the system. The same applies for the MRAC-LPV based on Lyapunov theory. For the multiplicative actuator fault of 100% applied at 15000, the MRAC-H<sub>∞</sub>GS-LPV and the MRAC-LPV based on the MIT rule became unfeasible after the occurrence of the fault. And on the other hand the MRAC-H<sub>∞</sub>GS-LPV and the MRAC-LPV based on Lyapunov theory became degraded after the presence of the fault. It is important to mention that just the schemes based on the Lyapunov theory were able to tolerate the change in the operating point having a controlled chattering of maximum +/- 1 cm.

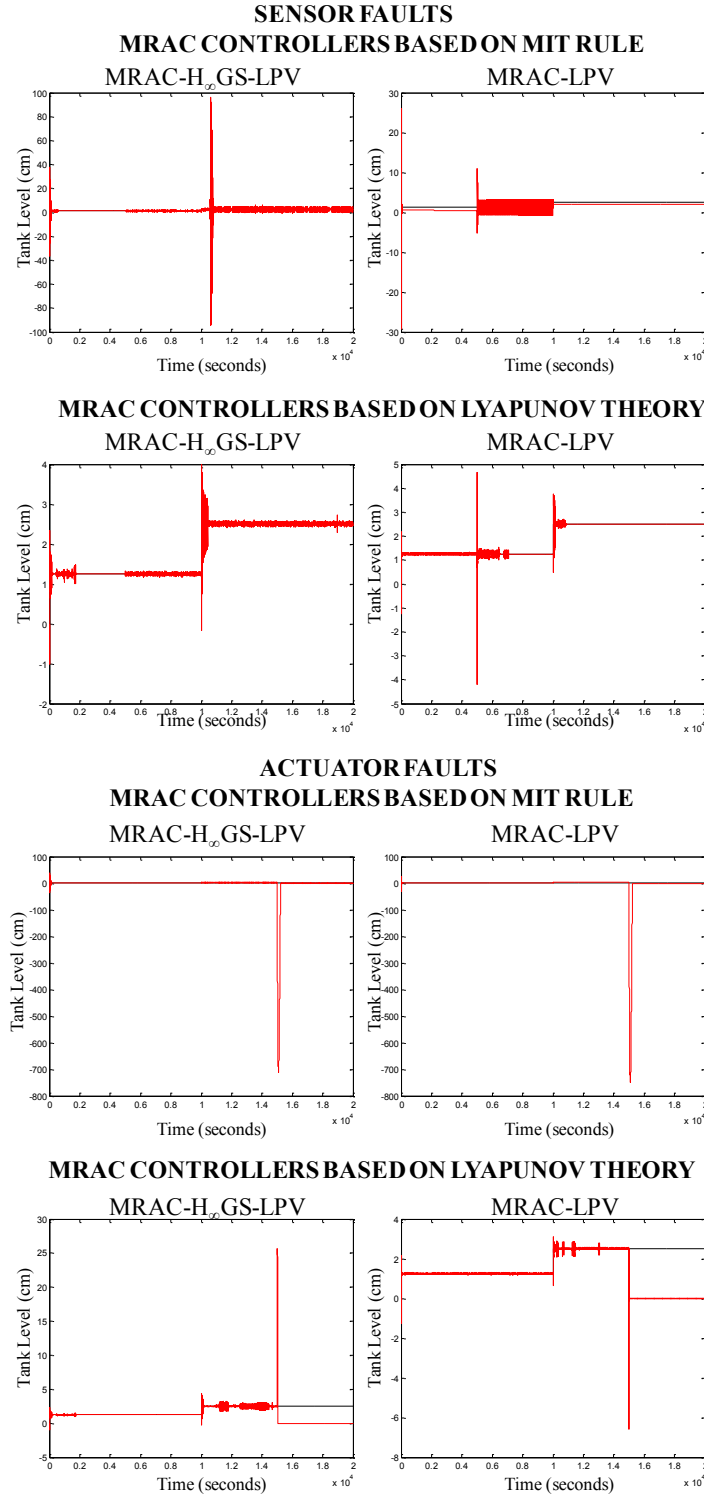
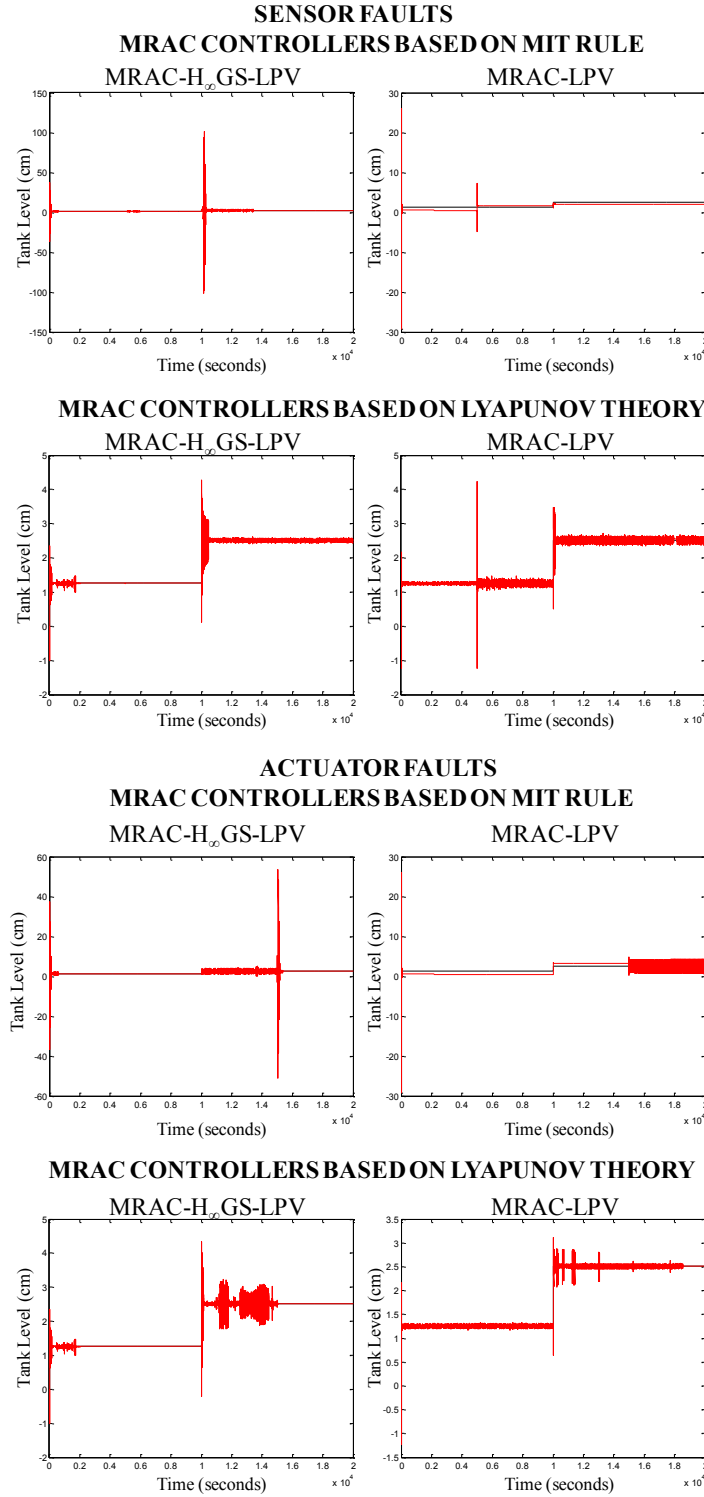


Figure C. 9. Comparison between the system output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 100% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.



**Figure C. 10. Comparison between the system output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 90% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**



In Figure C.10, it can be observe that for the multiplicative fault in sensors of 90% at time 5000 applied to the nonlinear system. The MRAC- $H_\infty$ GS-LPV based on the MIT rule became unfeasible after the change in the operating point because it reached a system output value of  $\pm 100$  cm and the high of the tanks that the system is representing is of 30 cm. The same applies for the MRAC-LPV based on the MIT rule in which the system output reached a value of more than  $\pm 30$  at the beginning of the simulation and presents an offset during the simulation. On the other hand, in the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, but presented some reasonable chattering after the change in the operating point. The chattering effects are due to the nonlinearities of the system. The MRAC-LPV was fault tolerant to the multiplicative sensor fault but stayed with some reasonable chattering. For the multiplicative actuator fault of 90% applied at 15000, the MRAC- $H_\infty$ GS-LPV based on the MIT rule became unfeasible after the occurrence of the fault. The MRAC-LPV based on MIT rule presented an offset since the beginning of the simulation and stayed with chattering problems ( $\pm 5$ cm) after the occurrence of the fault. On the other hand the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory presented some chattering (less than  $\pm 1$ cm) after the change in the operating point but was fault tolerant against the actuator multiplicative fault. In addition, the MRAC-LPV based on Lyapunov theory presented a chattering since de beginning of the simulation but also was fault tolerant to the multiplicative actuator fault. It is important to mention that just the schemes based on the Lyapunov theory were able to tolerate the change in the operating point having a controlled chattering of maximum  $\pm 1$  cm.

In Figure C.11, it can be observe that for the multiplicative fault in sensors of 50% at time 5000 applied to the nonlinear system, the MRAC- $H_\infty$ GS-LPV based on the MIT rule became unfeasible after the change in the operating point because it reached a system output value of  $\pm 100$  cm and the high of the tanks that the system is representing is of 30 cm. The same applies for the MRAC-LPV based on the MIT rule in which the system output reached a value of more than  $\pm 30$  at the beginning of the simulation and presents an offset during the simulation and chattering problems ( $\pm 5$  cm) after the occurrence of the fault. On the other hand, in the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, but presented some reasonable chattering after the change in the operating point. The chattering effects are due to the nonlinearities of the system. The MRAC-LPV was fault tolerant to the multiplicative sensor fault but stayed with some reasonable chattering. For the multiplicative actuator fault of 50% applied at 15000, the MRAC- $H_\infty$ GS-LPV based on the MIT rule became unfeasible after the occurrence of the fault. The MRAC-LPV based on MIT rule presented an offset since the beginning of the simulation. On the other hand the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory presented some chattering (less than  $\pm 1$ cm) after the change in the operating point but was fault tolerant against the actuator multiplicative fault. In addition, the MRAC-LPV based on Lyapunov theory presented a chattering since de beginning of the simulation but also was fault tolerant to the multiplicative actuator fault. It is important to mention that just the schemes based on the Lyapunov theory were able to tolerate the change in the operating point having a controlled chattering of maximum  $\pm 1$  cm.

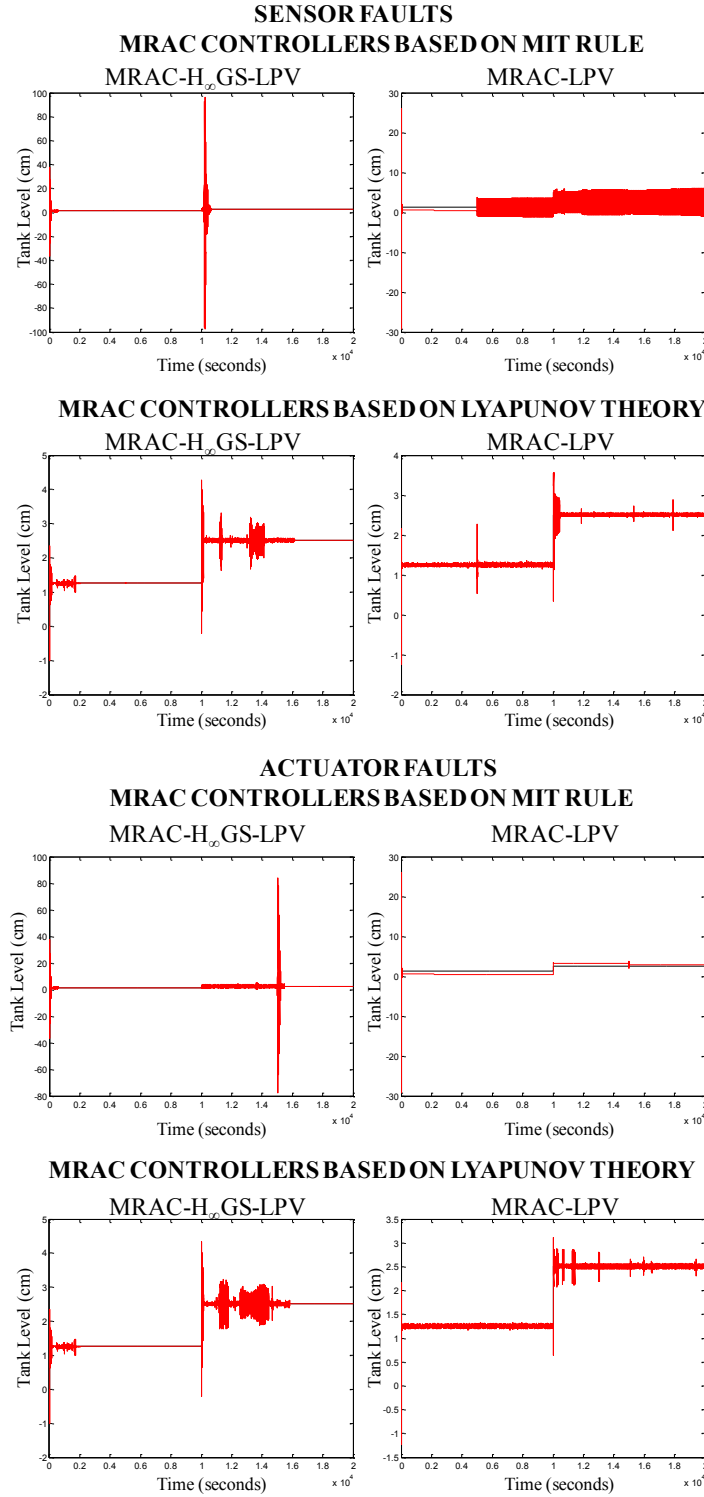
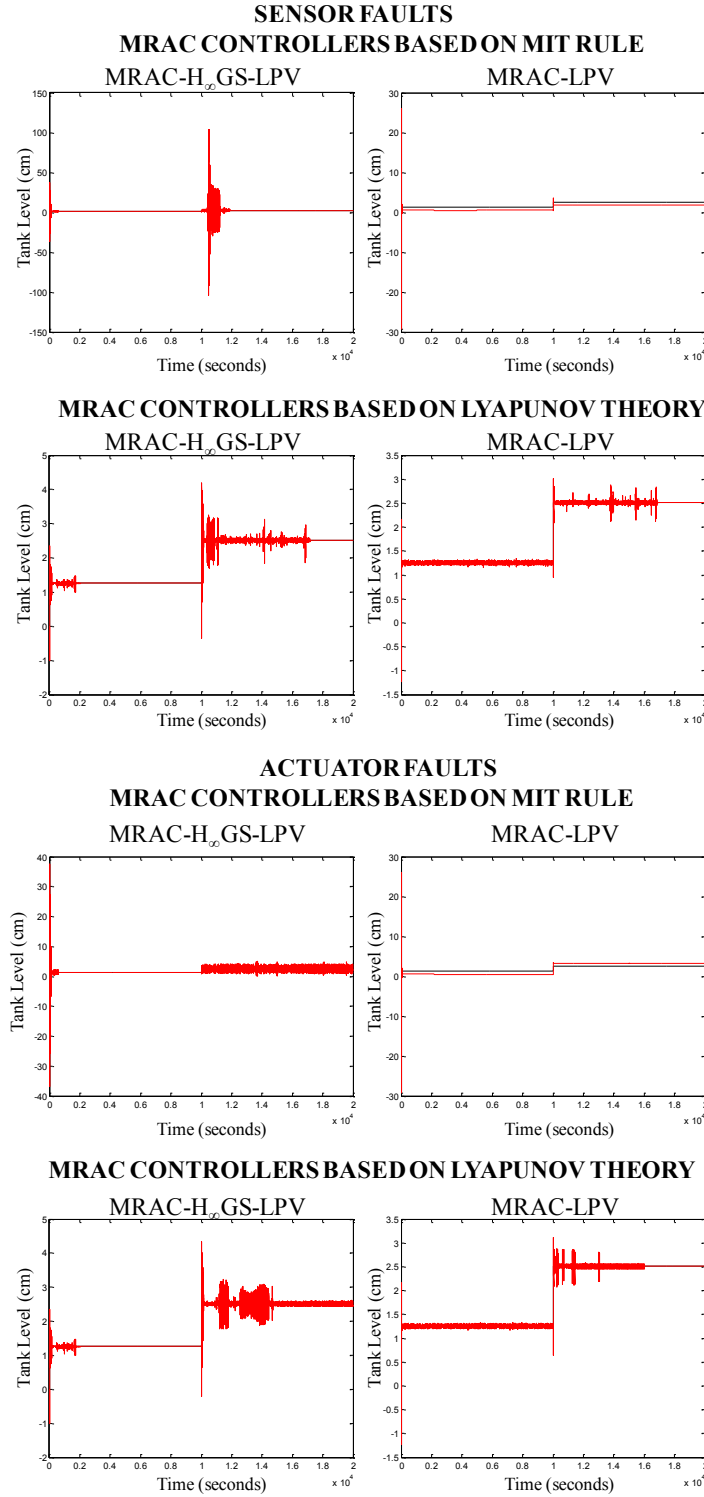


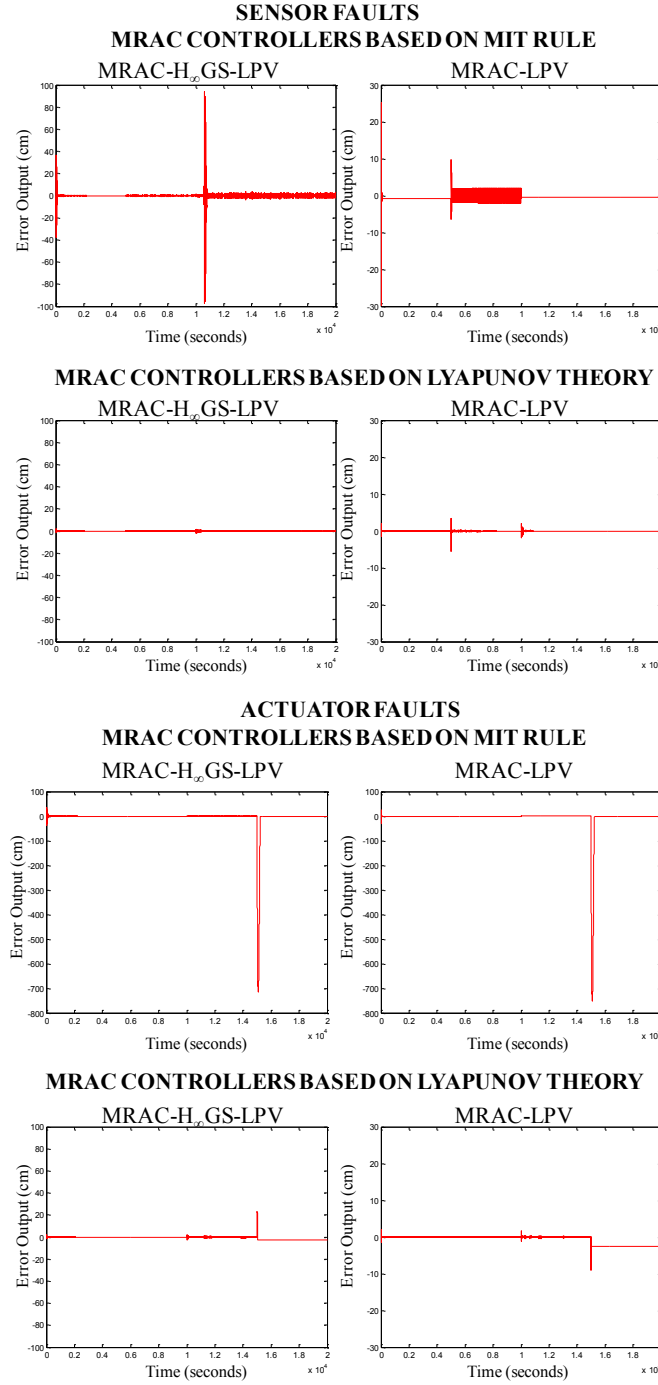
Figure C. 11. Comparison between the system output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 50% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.



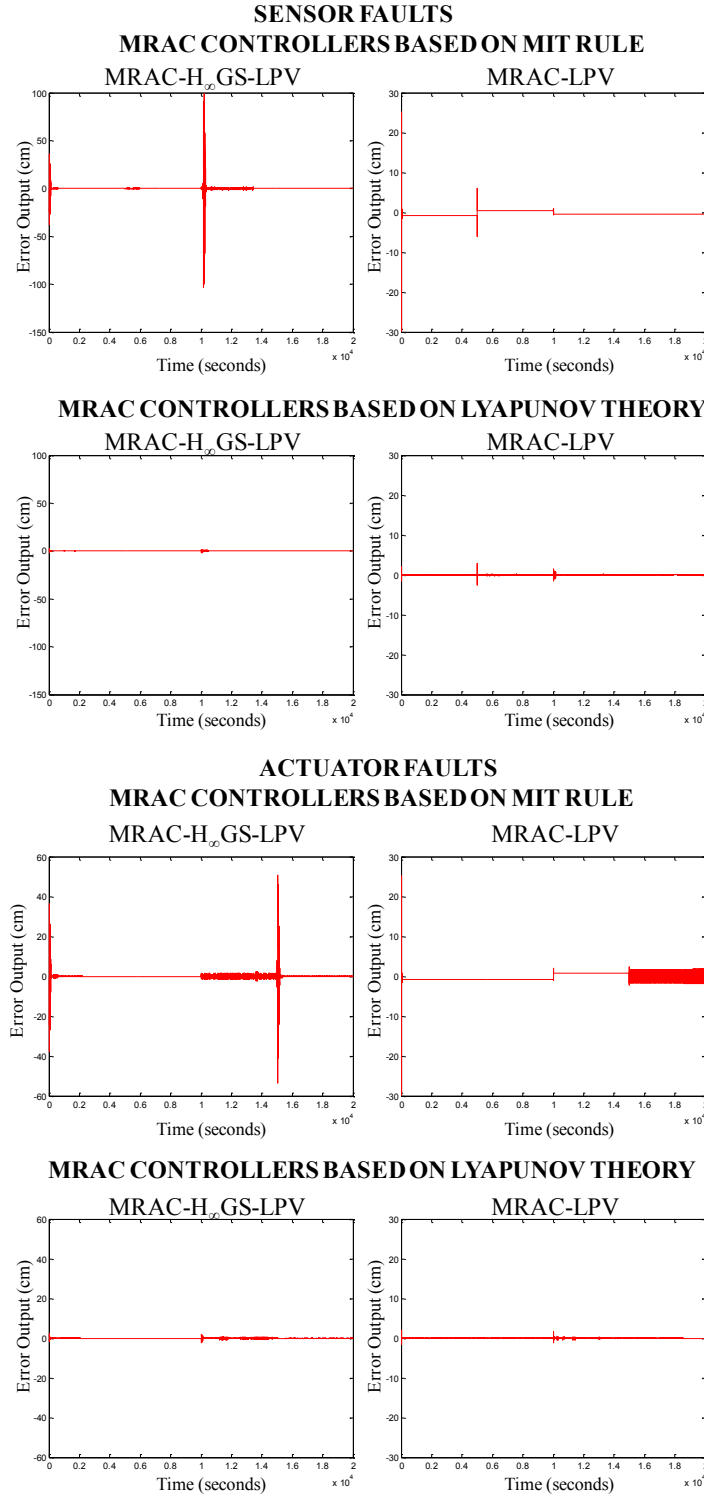
**Figure C. 12. Comparison between the system output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 5% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure C.12, it can be observe that for the multiplicative fault in sensors of 5% at time 5000 applied to the nonlinear system, the MRAC- $H_\infty$ GS-LPV based on the MIT rule became unfeasible after the change in the operating point because it reached a system output value of  $\pm 100$  cm and the high of the tanks that the system is representing is of 30 cm. The same applies for the MRAC-LPV based on the MIT rule in which the system output reached a value of more than  $\pm 30$  at the beginning of the simulation and presents an offset during the simulation. On the other hand, in the MRAC- $H_\infty$ GS-LPV based on Lyapunov was robust against the multiplicative sensor fault, but presented some reasonable chattering after the change in the operating point. The chattering effects are due to the nonlinearities of the system. The MRAC-LPV was fault tolerant to the multiplicative sensor fault but stayed with some reasonable chattering. For the multiplicative actuator fault of 5% applied at 15000, the MRAC- $H_\infty$ GS-LPV based on the MIT rule became unfeasible after the occurrence of the fault. The MRAC-LPV based on MIT rule presented an offset since the beginning of the simulation and also became unfeasible because it reached  $\pm 30$  cm of variation in the system output at the beginning of the simulation. On the other hand the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory presented some chattering (less than  $\pm 1$  cm) after the change in the operating point but the actuator multiplicative fault was imperceptible. In addition, the MRAC-LPV based on Lyapunov theory presented a chattering since de beginning of the simulation but also was fault tolerant to the multiplicative actuator fault. It is important to mention that just the schemes based on the Lyapunov theory were able to tolerate the change in the operating point having a controlled chattering of maximum  $\pm 1$  cm.

The next figures represent the error for the experiments tested in the nonlinear system. In Figure C.13, it can be observe that for the single multiplicative sensor fault of 100% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 30$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 5$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 2$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. On the other hand for the single multiplicative actuator fault of 100% the error represents approximately 800 cm of the system output for the schemes based on the MIT rule (MRAC- $H_\infty$ GS-LPV and MRAC-LPV), it varies 30 cm from the system output set point in the MRAC- $H_\infty$ GS-LPV scheme based on Lyapunov theory and it varies 10 cm in the MRAC-LPV scheme based on Lyapunov theory. In Figure C.14, it can be observe that for the single multiplicative sensor fault of 90% the error range between  $\pm 50$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 5$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 4$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 1$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. On the other hand for the single multiplicative actuator fault of 90% the error represents approximately  $\pm 60$  cm of the system output for the MRAC- $H_\infty$ GS-LPV scheme based on the MIT rule, it varies  $\pm 5$  cm of the system output for the MRAC-LPV scheme based on the MIT rule, it varies  $\pm 1$  cm from the system output set point in the MRAC- $H_\infty$ GS-LPV scheme based on Lyapunov theory and it varies  $\pm 3$  cm in the MRAC-LPV scheme based on Lyapunov theory.



**Figure C. 13. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 100% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

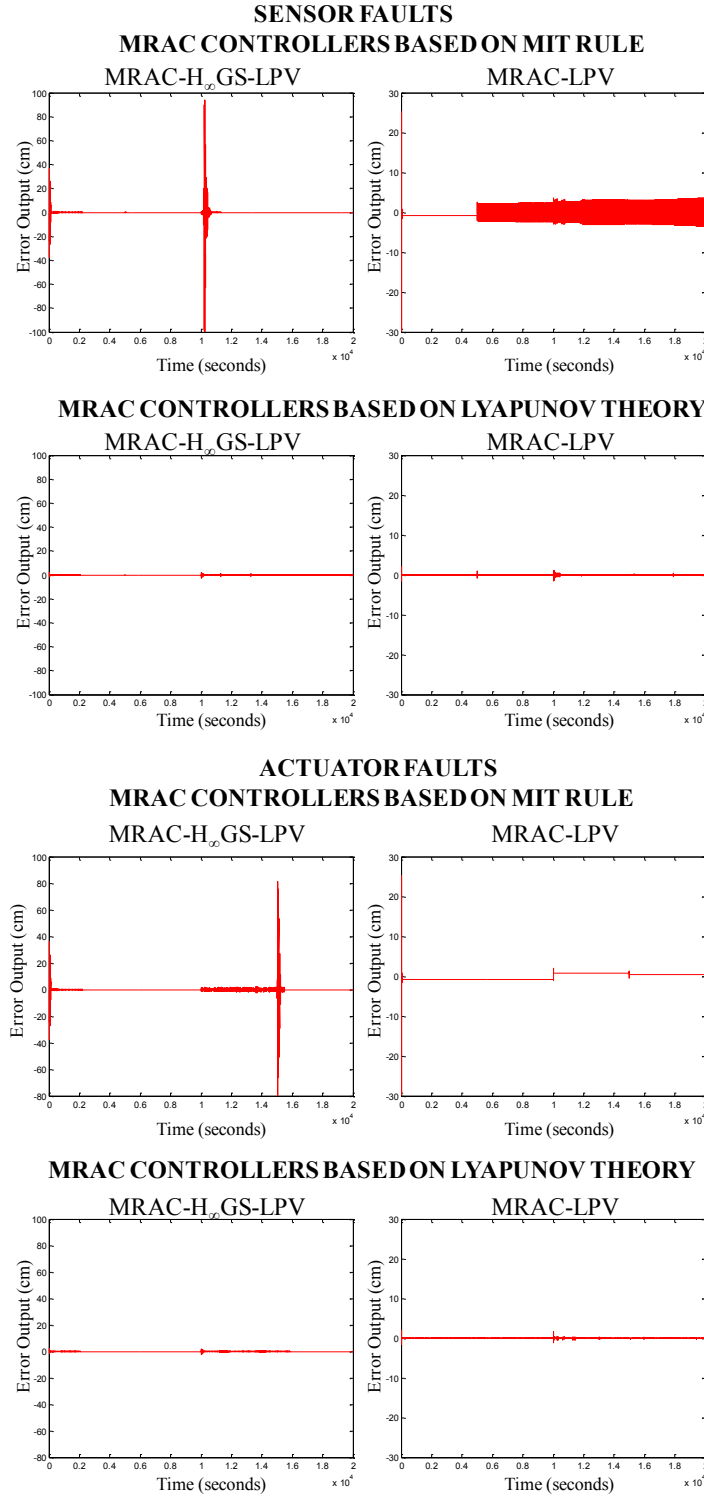


**Figure C. 14. Comparison between the error output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 90% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

In Figure C.14, it can be observe that for the single multiplicative sensor fault of 90% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 10$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 3$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 3$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. On the other hand for the single multiplicative actuator fault of 90% the error represents approximately  $\pm 80$  cm of the system output for the MRAC- $H_\infty$ GS-LPV scheme based on the MIT rule, it varies  $\pm 4$  cm of the system output for the MRAC-LPV scheme based on the MIT rule, it varies  $\pm 3$  cm from the system output set point in the MRAC- $H_\infty$ GS-LPV scheme based on Lyapunov theory and it varies  $\pm 3$  cm in the MRAC-LPV scheme based on Lyapunov theory.

In Figure C.15, it can be observe that for the single multiplicative sensor fault of 50% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 10$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 3$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 3$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. On the other hand for the single multiplicative actuator fault of 50% the error represents approximately  $\pm 80$  cm of the system output for the MRAC- $H_\infty$ GS-LPV scheme based on the MIT rule, it varies  $\pm 4$  cm of the system output for the MRAC-LPV scheme based on the MIT rule, it varies  $\pm 3$  cm from the system output set point in the MRAC- $H_\infty$ GS-LPV scheme based on Lyapunov theory and it varies  $\pm 3$  cm in the MRAC-LPV scheme based on Lyapunov theory.

In Figure C.16, it can be observe that for the single multiplicative sensor fault of 5% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 5$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 3$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 3$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. On the other hand for the single multiplicative actuator fault of 5% the error represents approximately  $\pm 30$  cm of the system output for the MRAC- $H_\infty$ GS-LPV scheme based on the MIT rule, it varies  $\pm 4$  cm of the system output for the MRAC-LPV scheme based on the MIT rule, it varies  $\pm 3$  cm from the system output set point in the MRAC- $H_\infty$ GS-LPV scheme based on Lyapunov theory and it varies  $\pm 3$  cm in the MRAC-LPV scheme based on Lyapunov theory.



**Figure C. 15. Comparison between the error output of the MRAC- $H_{\infty}$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 50% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**



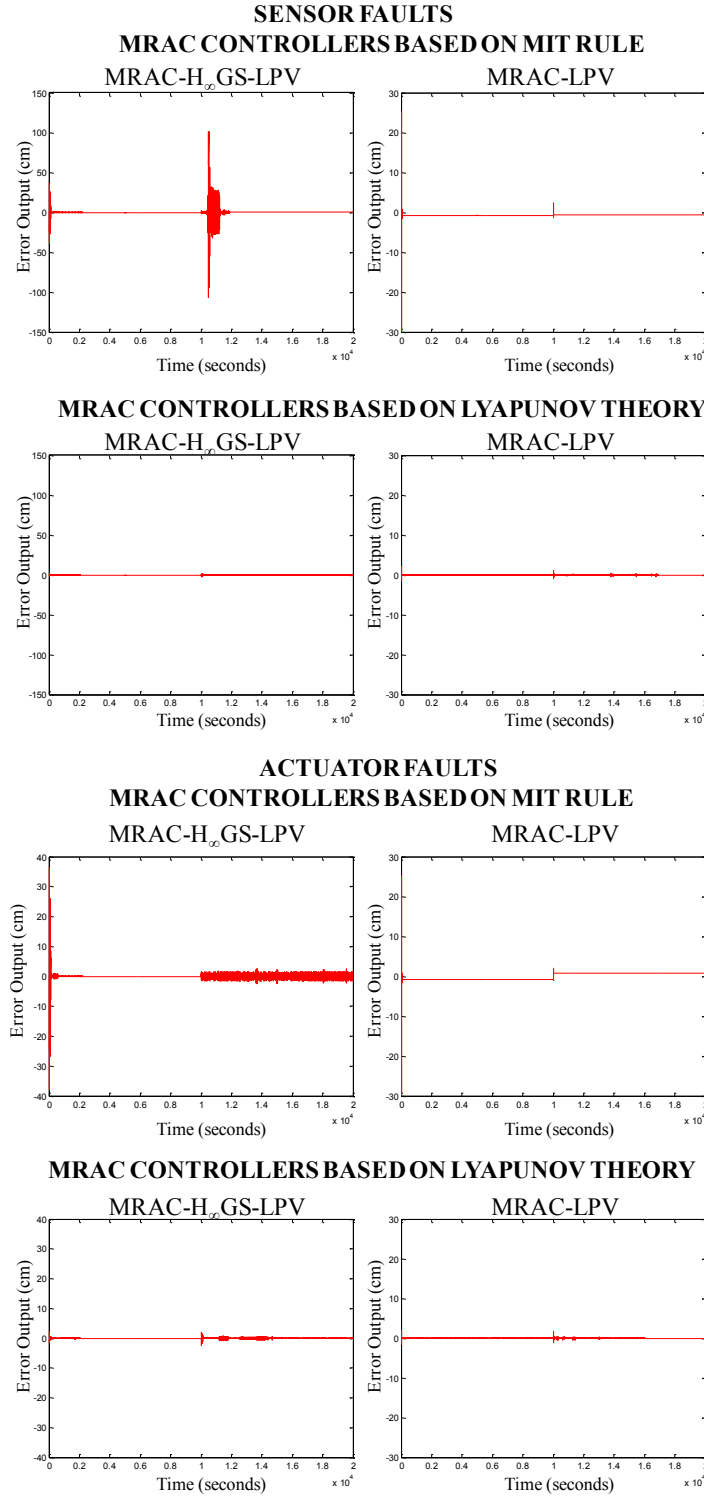
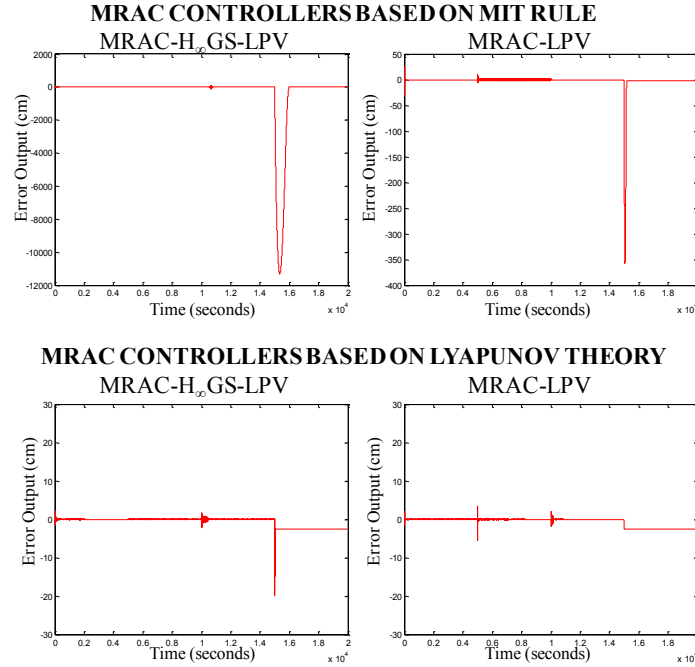


Figure C. 16. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a single fault of 5% (multiplicative sensor and multiplicative actuator fault) for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.

In Figure C.17, it can be observe that for the combination of multiplicative sensor fault and multiplicative actuator fault both of 100% the error range between - 12000 cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies - 400 cm for the MRAC-LPV based on MIT rule, varies  $\pm 5$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 20$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. In Figure C.18, it can be observe that for the combination of multiplicative sensor fault and multiplicative actuator fault both of 90% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 60$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 3$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 4$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. In Figure C.19, it can be observe that for the combination of multiplicative sensor fault and multiplicative actuator fault both of 50% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 30$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 4$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 4$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory. In Figure C.20, it can be observe that for the combination of multiplicative sensor fault and multiplicative actuator fault both of 5% the error range between  $\pm 100$  cm from the set point in the MRAC- $H_\infty$ GS-LPV based on the MIT rule, it varies  $\pm 30$  cm for the MRAC-LPV based on MIT rule, varies  $\pm 3$  cm in the MRAC-LPV based on Lyapunov theory and varies  $\pm 3$  cm in the MRAC- $H_\infty$ GS-LPV based on Lyapunov theory.



**Figure C. 17. Comparison between the error output of the MRAC- $H_\infty$ GS-LPV and MRAC-LPV Controllers based on the MIT rule and on the Lyapunov theory with a combination of a multiplicative sensor fault (5000 seconds) and a multiplicative actuator faults (15000 seconds), both of 100%, for the operating points  $\varphi_1=0.3$  and  $\varphi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.**

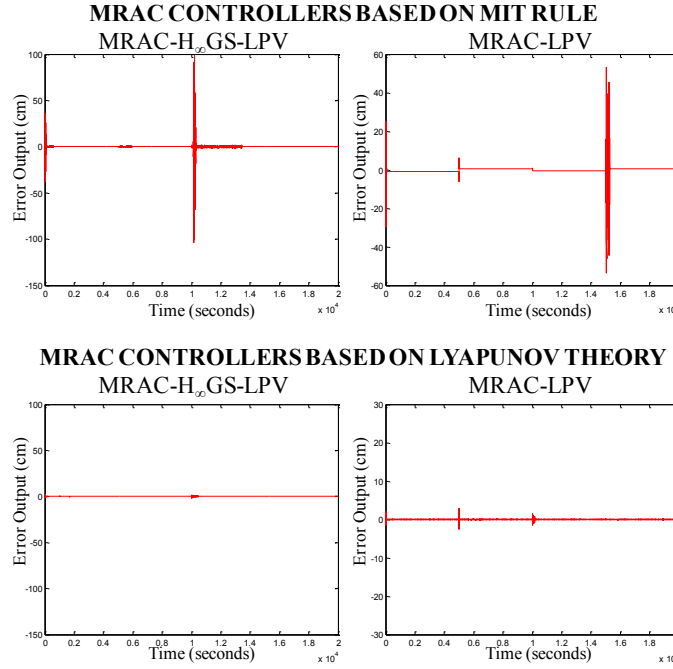


Figure C. 18. Comparison between the error output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.

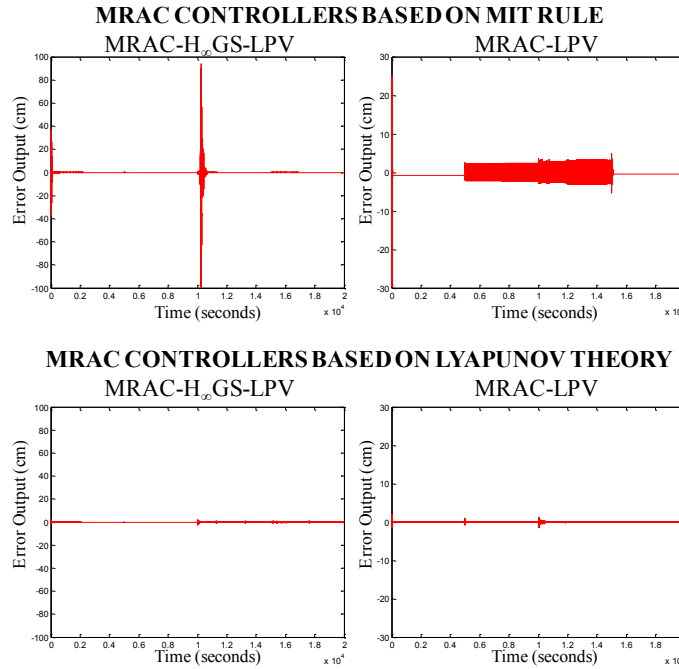


Figure C. 19. Comparison between the error output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 90%.

50%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.

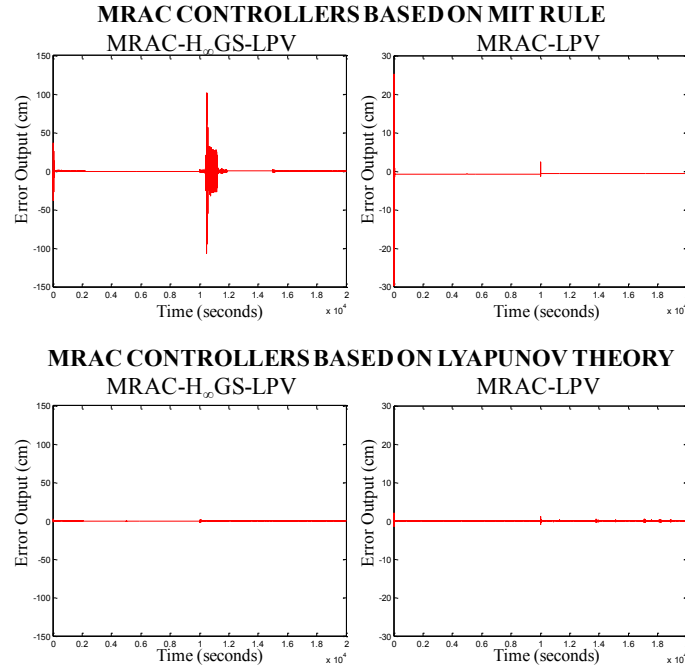


Figure C. 20. Comparison between the error output of the MRAC-H<sub>∞</sub>GS-LPV and MRAC-LPV Controllers based on the MIT rule and the Lyapunov theory with a combination of a multiplicative sensor fault at time 5000 seconds and a multiplicative actuator faults at time 15000 seconds both of 5%, for the operating points  $\phi_1=0.3$  and  $\phi_2=0.5$  and a change in the operating point at time 10000 seconds for the nonlinear system.