Control of Hybrid Systems and Discrete-Event Systems

Naly Rakoto-Ravalontsalama

Université de Nantes, France

HABILITATION A DIRIGER DES RECHERCHES

English Version

Naly Rakoto-Ravalontsalama

IMT Atlantique - LS2N

Control of Hybrid Systems and Discrete-Event Systems

Commande de Systèmes Hybrides et de Systèmes à Evènements Discrets

Ecole Doctorale : STIM

Specialization: Automatic Control

Date of defense: 6 September 2017

EXAMINING COMMITTEE

Reviewers:

Didier Henrion	Research Director, LAAS-CNRS Toulouse
Daniel Liberzon	Professor, University of Illinois at Urbana-Champaign, USA
Eric Niel	Professor, INSA Lyon - Ampère

Examiners:

Alexandre Dolgui	Professor, IMT Atlantique - LS2N
Claude Jard	Professor, University of Nantes - LS2N
Stephane Lafortune	Professor, University of Michigan, Ann Arbor, USA
Jean-Jacques Loiseau	Research Director, CNRS - LS2N

Contents

Remerciements

Je voudrais remercier :

- les rapporteurs : Didier Henrion, Daniel Liberzon et Eric Niel.
- les examinateurs : Alexandre Dolgui, Claude Jard, Stephane Lafortune et Jean-Jacques Loiseau.
- mes Doctorants: Santi, Jose-Luis, Eduardo et German.
- ma famille : Pascale, Alice et Camille, et ma mère, Emilienne

Enfin, cette Thèse d'HDR est dédiée à la mémoire de mon père, Prof. Dr. Georges Rakoto-Ravalontsalama (1937-2011).

Introduction

The HDR (Habilitation à Diriger des Recherches) is a French Degree that you get some years after the PhD. It allows the candidate to apply for some University Professor positions and/or to apply for a Research Director position at CNRS. Instead of explaining it in details, the selection phases process after an HDR is summarized with a Petri Net model in Figure 1.

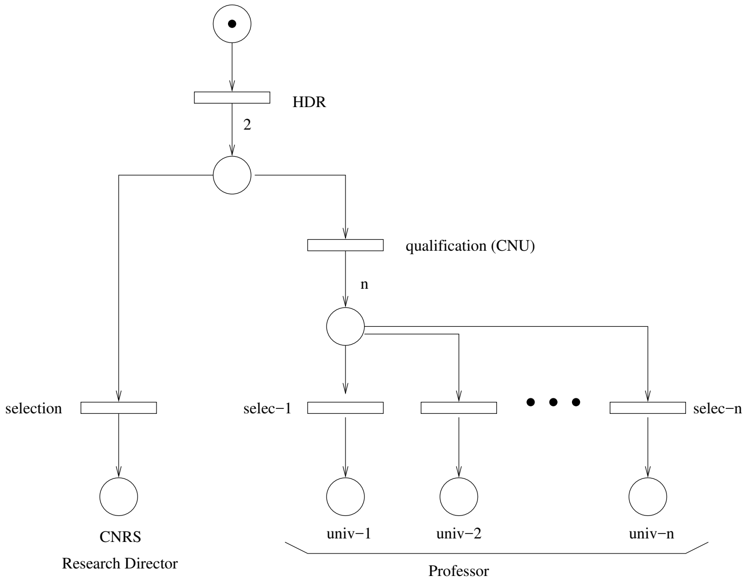


Figure 1: Selection Phases after the HDR

After obtaining the HDR Thesis Degree, the candidate is allowed to apply for a Research Director position at CNRS, after a national selection. On the other hand, in order to apply for some University Professor positions, the candidate should first apply for a National Qualifcation (CNU). Once this qualification obtained, the candidate can then apply to some University Professor positions, with a selection specific to each university.
This HDR Thesis is an extended abstract of my research work from my PhD Thesis defense in 1993 until now. This report is organized as follows.

- Chapter 1 is a Curriculum Vitae
- Chapter 2 presents the Analysis and Control of Hybrid and Switched System
- Chapter 3 is devoted to Supervisory Control of Discrete-Event Systems
- Chapter 4 gives the Conclusion and Future Work

Chapter 1

Curriculum Vitae

1.1 Personal Data

Born on February 19, 1965; Married and has 2 daughters (19 and 15 years old, resp.)
Citizenship: French.

A. Affiliation

- Dept. of Automation, Production and Computer Science (DAPI)

IMT Atlantique (ex-Mines Nantes)	Phone:	+33 (0)2 5185 8306
4 Rue Alfred Kastler	Fax:	+33 (0)2 5185 8349
44307 Nantes Cedex 03	e-mail:	`naly.rakoto@mines-nantes.fr`
France	web:	www.imt-atlantique.fr

- Also member of **LS2N** Laboratory, Nantes (UMR CNRS 6004) in the PSI Research Group
www.ls2n.fr

B. Education

1993	Ph.D. in Automatic Control *(Très Honorable)*	LAAS-CNRS and University of Toulouse, France
1989	M.Sc. [DEA] in Automatic Control *(A. Bien)*	LAAS-CNRS and University of Toulouse, France
1987-88	Licence + Maîtrise EEA *(Assez Bien)*	University of Toulouse, France
1985	DEUG A Physique-Chimie *(Bien)*	University of Toulouse, France
1983	National Service	Tananarive, Madagascar
1982	Baccalauréat série C *(Assez Bien)*	Tananarive, Madagascar

1.2 Teaching Activities

A. Teaching

I am teaching regularly a yearly *normal service* i.e. 192 h eq TD since I arrived at Ecole des Mines in September 1994. The subject and the students have a bit evolved since 1994. Here below is a summary of the courses that I gave during the Academic year 2015-2016. What is new compared to the beginning in 1994 are the courses given in the 2 Master of Science programs MOST (Management and Optimization in Supply Chain and Transport) and PM3E (Project Management for Environmental, and Energy Engineering). These 2 programs are offered entirely in English.

Promo	Course	CM	PC	TD	TP-MP	PFE	Resp	Lang.
A1	Automatique	10h		10h	10h			Fr.
A2	Optim.			10h	5h			Fr.
A2 AII	SED		10h	5h	5h			Fr.
A3 AII	SysHybrides		7.5h		7.5h			Fr.
MSc. PM3E	Control		7.5h		7.5h			Eng.
MSc. MOST	Simulation		5h	5h	5h			Eng.
MSc. MOST	Resp. MSc+UV						90h	Eng.
A3	PFE superv.					36h		Fr.
Masters	PFE superv.					36h		Eng.
Total 1	**272h**	10h	30h	30h	40h	72h	90h	
Total 2	**283up**	15up	36up	30up	40up	72up	90up	

Table 1.1: Teaching in 2015-2016

B. Responsabilities (Option AII, Auto-Prod, MSc MLPS, MSc MOST)

I had and am having the following administrative responsabilities at Mines Nantes:

- 1997-2000: Last Year's Option AII (Automatique et Informatique Industrielle)
- 2001-2004: First and Second Year: Control and Industrial Eng. courses at DAP
- 2006-2012: MSc. MLPS (Management of Logistic and Production Systems)
- 2012-present: MSc. MOST (Management and Optimization of Supply Chains and Transport)

C. Courses given abroad

- May 2008: Univ. of Cagliari (Italy): Control of Hybrid Systems (10h) *Erasmus*
- Apr. 2009: Univ. Tec. Bolivar UTB, Cartagena (Colombia): Tutorial on DES (15h)
- May 2014: Univ. Tec. Bolivar UTB, Cartagena (Colombia): Intro. to DES (15h)
- Dec 2015: ITB Bandung (Indonesia): Simulation with Petri Nets (10h) *Erasmus*
- Apr. 2017: Univ. of Liverpool (UK): Course 1 (10h) *Erasmus*
- May 2017: ITB Bandung (Indonesia): Simulation with Petri Nets (10h) *Erasmus*

1.3 Research Activities

My main topics of research are the following:

1. Analysis and control of hybrid and switched systems
2. Supervisory control of discrete-event systems

These will be detailed in Chapter 2 and Chapter 3, respectively. The following other topics of research will not be presented. However, the corresponding papers can be found in the Complete List of Publications.

- Resource Allocation
- Holonic Systems
- Inventory Control

1.4 Supervision of Students: PhD, MSc.

A. PhD Students:

- **Santi Esteva**, PhD defended in Girona in March 2003.
  -*Modelling, Control and Supervision for a Class of Hybrid Systems*
  - PhD Committee: J. Aguilar-Martin, J.L. de la Rosa, J. Colomer, J.C. Hennet, E. Garcia, J. Melendez, V. Puig, N. Rakoto, G. Roux
  - Supervision: J.L. de la Rosa (50 %), N. Rakoto (50 %)
  - Publications: 1 conference paper
  - Current activity: Associate Professor at University of Girona, Spain.

- **Jose-Luis Villa**, PhD defended in Nantes in February 2004.
  -*Modélisation et commande de systèmes hybrides : L'approche MLD*
  - PhD Committee: M. Morari, K.E. Arzen, M. Duque, A. Gauthier, J.J. Loiseau, N. Rakoto
  - Supervision: M. Duque (40 %), A. Gauthier (10 %), N. Rakoto (40 %), J.J. Loiseau (10 %)
  - Publications: 1 book chapter, 12 conference papers
  - Current activity: Profesor Titular at Universidad Tecnologica Bolivar, Cartagena, Colombia.

- **Eduardo Mojica**, PhD defended in Nantes in September 2009.
  -*A polynomial approach for analysis and optimal control of switched nonlinear systems*
  - PhD Committee: P. Caines, D. Henrion, A. Gauthier, J.J. Loiseau, N. Quijano, P. Riedinger, N. Rakoto
  - Supervision: M. Quijano (40 %), A. Gauthier (10 %), N. Rakoto (40 %), J.J. Loiseau (10 %)
  - Publications: 2 journal papers, 6 conference papers
  - Current activity: Associate Professor at Universidad Nacional, Bogota, Colombia.

- **German Obando**, PhD defended in Nantes in October 2015.
  -*Distributed methods for resource allocation: A passivity-based approach*
  - PhD Committee: C. Ocampo-Martinez, H. Gueguen, A. Dolgui, A. Gauthier, J.J. Loiseau, N. Quijano, N. Rakoto
  - Supervision: M. Quijano (40 %), A. Gauthier (10 %), N. Rakoto (40 %), J.J. Loiseau (10 %)
  - Publications: 1 journal paper, 2 conference papers
  - Current activity: PostDoc at Universidad de los Andes, Bogota, Colombia.

B. MSc Students:

- 2017: Nawapol Yamclee (IMTA / MSc. PM3E) – Control of Smart Grids
- 2017: Dina Lavender (IMTA / MSc. MOST) – Simulation with Stochastic Petri Nets
- 2010: Amadou Sagna (ECN / Master AIA) – Model Predictive Control
- 2003: Xiaoyu Chen (EMN Nantes / MSc. MOST) – Supervisory Control
- 1996: Nadia Pariset (EMN Nantes / Option AII) – Hybrid Petri Nets
- 1994: Jean-Sebastien Besse (INSA Toulouse) – G2 Expert System

C. Member of Ph.D. Thesis Committees (other than my PhD Students)

- Ph.D. examiner of D. Fragkoulis, LAAS-CNRS, Univ. of Toulouse, France (Nov. 2008)
Detection et localisation des défauts provenant des capteurs et des actionneurs : application sur un système non linéaire.

- Ph.D. examiner of Aimed Mokhtari, LAAS-CNRS, Univ. of Toulouse, France (Sep. 2007)
  Diagnostic de systèmes hybrides : développement d'une méthode associant la détection par classification et la simulation dynamique.
- Ph.D. examiner of Hector Hernandez de Leon, LAAS-CNRS, Univ. of Toulouse, France (Sep. 2006)
  Supervision et diagnostic des procédés de production d'eau potable.
- Ph.D. examiner of Flavio Neves-Junior, LAAS-CNRS, Univ. of Toulouse, France (Nov. 1998)
  Supervision et commande des phases transitoires des processus industriels : application a une colonne de distillation.

D. Short Research Visits

- May-June 2004: McGill University, Montreal, Canada. Host: Prof. Peter E. Caines (2 months)
- Sep. 2013: University of Michigan, Ann Arbor, MI, USA. Host: Prof. Stephane Lafortune (1 month)

E. Invited Plenary Talks

- Invited Plenary Talk, *Analysis and Control of Hybrid and Switched Systems*, Colombian Control Conf., Cartagena, Colombia, April 2009.

F. Member of Conference International Program Committees

- IFAC Conf. on Analysis and Design of Hybrid Systems (ADHS 2006), Alghero, Italy, June 2006.
- IEEE Conf. on Emerging Technologies and Factory Automation (ETFA 2001) Antibes, FR, 2001
- Int. Conf. on Automation of Mixed Processes: (ADPM 1998), Reims, France, March 1998.

G. Member of Conference Organizing Committees

- 7th Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing (SOHOMA 2017) Nantes, France, 2017.
- IFAC Conf. on Analysis and Design of Hybrid Systems (ADHS 2003), St. Malo, France, 2003.
- Conf. Int. Francophone en Automatique (CIFA 2002), Nantes, France, July 2002.
- Int. Conf. on Automation of Mixed Processes (ADPM 2000), Dortmund, Germany, Sep. 2000.

1.5 Funded and Submitted Projects

- Co-Principal Investigator (with Andi Cakravastia, ITB), *LOG-FLOW*, **PHC NUSANTARA France Indonesia**, Project N. 39069ZJ, 2017, *Accepted on 31 May 2017.*

- Participant, "Industrial Validation of Hybrid Systems", France and Colombia **ECOS Nord** Project N.C07M03, A. Gauthier and J.J. Loiseau PIs, Jan. 2007 to Dec. 2009 (3 years) Euro 12,000.

- Participant, French "Contrat Etat-Région" 2000-2006, **CER STIC 9** / N.18036, J.J. Loiseau PI, Euro 182,940 (US$ 182,940).

- Co-Principal Investigator (with Ph. Chevrel), *Modeling and Simulation of ESP Program*, **Peugeot-Citroen PSA France**, Sep. 2000 - Jan 2001, FF 20,000 (US$ 3,000).

- Co-Principal Investigator (with J. Aguilar-Martin), *Control and Supervision of a Distillation Process*, **Conseil Régional Midi-Pyrénées**, France, 1994-1995, FF 200,000 (US$ 30,000).

- Participant, **European Esprit Project IPCES** (Intelligent Process Control by means of Expert Systems), J. Aguilar-Martin PI, 1989-1992, Euro 500,000 (US$ 500,000).

1.6 Organization of Invited Sessions

- Invited Session, *Diagnosis and Prognosis of Discrete-Event Systems*, 48th IEEE CDC Shanghai, China, Dec 2009
  (jointly organized and chaired with Shigemasa Takai).
- Invited Session, *Diagnosis of DES Systems*, 1st IFAC DCDS 2007, Paris, France, June 2007
  (jointly organized and chaired with Shigemasa Takai).
- Invited Session, *DES and Hybrid Systems*, IEICE NOLTA 2006, Bologna, Italy, Sep. 2006
  (jointly organized and chaired with Shigemasa Takai).
- Invited Session, *Supervisory Control*, IFAC WODES, Reims, France, Sep. 2004
  (jointly organized and chaired with Toshimitsu Ushio).
- Invited Session, *Hybrid Systems*, IEEE ISIC 2001, Mexico City, Mexico, Sep. 2001
  (jointly organized and chaired with Michael Lemmon).
- Invited Session, *Knowledge Based Systems*, IEEE ISIC 1999, Cambridge, MA, USA, Sep. 1999
  (jointly organized and chaired with Karl-Erik Årzèn).
- Workshop on G2 Expert System, LAAS-CNRS, Toulouse, France, Oct. 1995
  (jointly organized and chaired with Joseph Aguilar-Martin).

1.7 Complete List of Publications

A summary of the papers, classified per year, from 1994 to 2017, is given in the following table.

	Conf.	Book Chap.	Book Ed.	Journal	Total
1994	2				2
1995	2		1	1	4
1996	1			1	2
1997		1			1
1998	2				2
1999	1				1
2000					
2001	4	1			5
2002	1				1
2003	5	1			6
2004	6				6
2005	1				1
2006	3				3
2007	4				4
2008	2				2
2009	1				1
2010				1	1
2011					
2012	1				1
2013	2				2
2014	4			1	5
2015	2	1			3
2016				4	4
2017	1+3*			1*	1+4*

Table 1.2: Number of published papers per year (as of 30 June 2017) – where (*) means *submitted*

Complete List of Publications

[] **Book Edition**

[B.1] *Supervision de processus à l'aide du système expert G2*, N. Rakoto-Ravalontsalama and J. Aguilar-Martin (Eds.), Hermes Ed. Paris, Oct. 1995, ISBN 2-86601-499-5.

[Proceedings of Workshop on G2 Expert System, LAAS-CNRS Toulouse, France, Oct. 1995] [Includes 4 papers in English and 6 papers in French]

International Refereed Journals

[J.sub1] F. Torres, C. Garcia-Diaz and N. Rakoto-Ravalontsalama. "Evolutionary Dynamics of Two-actor VMI-driven Supply Chains", *Submitted, Dec. 2016.*

[J.9] G. Obando, N. Quijano, and N. Rakoto-Ravalontsalama. "A Center-Free Approach for Resource Allocation with Lower Bounds", International Journal of Control, 2016. DOI: 10.1080/00207179.2016.1225167.

[J.8] C. Indriago, O. Cardin, N. Rakoto-Ravalontsalama, P. Castagna, E. Chacon. "H2CM: A holonic architecture for flexible hybrid control systems"', Computers in Industry, Elsevier, 77 (2016) pp. 15–28.

[J.7] C. Indriago, O. Cardin, O. Morineau, N. Rakoto-Ravalontsalama, P. Castagna, E. Chacon. "Performance evaluation of holonic control of a switch arrival system"', Concurrent Engineering: Research and Applications, SAGE, 2016, DOI: 10.1177/1063293X16643568.

[J.6] C. Indriago, O. Cardin, O. Bellenguez-Morineau, N. Rakoto, P. Castagna, E. Chacon. "Evaluation de l'application du paradigme holonique à un système de réservoirs"', Journal Européen des Systèmes Automatisés JESA, vol. 49 N.23, pp.325-347, 2016.

[J.5] E. Mojica, N. Quijano, and N. Rakoto-Ravalontsalama *A polynomial approach for optimal control of switched nonlinear systems*, Int. Journal of Robust and Nonlinear Control, Wiley, 2014, 24 (12), pp.1797-1808.

[J.4] E. Mojica, N. Quijano, N. Rakoto-Ravalontsalama, and A. Gauthier *A polynomial approach for stability analysis of switched systems*, Systems and Control Letters 59 (2010) 98–104.

[J.3] N. Rakoto-Ravalontsalama, J. Aguilar-Martin, *Knowledge-based modelling of a TV-tube manufacturing system*, IFAC Journal of Control Engineering Practice, Jan. 1996, 4(1), pp. 117–123.

[J.2] P. Bourseau, K. Bousson, P. Dague, J.L. Dormoy, J.M. Evrard, F. Guerrin, L. Leyval, O. Lhomme, B. Lucas, A. Missier, J. Montmain, N. Piera, N. Rakoto-Ravalontsalama, J.P. Steyer, M. Tomasena, L. Trave-Massuyes, M. Vescovi, S. Xanthakis and B. Yannou, *Qualitative reasoning: A survey of techniques and applications AICOM Journal*, Sept-Dec. 1995, vol. 8, N. 3-4, pp. 119–192.

[J.1] N. Rakoto-Ravalontsalama, A Missier, and J.S. Kikkert, *Qualitative operators and process engineer semantics of uncertainty.* In B. Bouchon-Meunier, L. Valverde, and R.R. Yager (Eds.)

Lecture Notes in Computer Science N. 682, IPMU'92 - Advanced Methods in Artificial Intelligence, Springer Verlag 1992, pp. 284–293.

Book Chapters

[B.Ch.4] C. Indriago, O. Cardin, N. Rakoto, E. Chacon, P. Castagna, "Application of holonic paradigm to hybrid processes: Case of a water treatment process"' Chapter of the book "Service Orientation in Holonic and Multi-agent Manufacturing", Springer, 2015 ISBN 978-3-319-15159-5.

[B.Ch.3] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Hybrid modeling of potable water treatment plant.* In. *Pumps, Electromechanical Devices and Systems Applied to Urban Water Management*, Cabrera and Cabrera Jr. Eds., 2003 Swets and Zeitlinger, Lisse, Switzerland, ISBN 90 5809 560 6, pp. 909–917.

[B.Ch.2] Y. Quenec'hdu, J. Buisson, N. Rakoto-Ravalontsalama, *Rappels sur les systèmes continus et échantillonnés*, Chapitre de l'ouvrage *Modélisation et commande de systèmes dynamiques hybrides* (J. Zaytoon coord.), Hermes Ed., Paris, 2001, pp. 29–59 (in French).

[B.Ch.1] N. Rakoto-Ravalontsalama, *Supervision et diagnostic de procédés industriels : IPCES*, Chapitre du livre *Le raisonnement qualitatif* (L. Trave-Massuyes, Ph. Dague, F. Guerin coord.), Hermes Ed., Paris 1997, pp. 279–322 (in French).

International Conferences with Proceedings

[C.sub3] D. Lavender, A. Cakravastia, Y Lafdail, and N. Rakoto-Ravalontsalama, *Modeling and Simulation of Baggage Handling System in a Large Airport , Submitted, June 2017.*

[C.sub2] N. Yamclee, C. Nicolas-Rodriguez, and N. Rakoto-Ravalontsalama, *Switched LQR Control of Interleaved Double Dual Boost Converters, Submitted, May 2017.*

[C.sub1] M. Canu and N. Rakoto-Ravalontsalama. On Switchable Languages of Discrete-Event Systems with Weighted Automata, *Submitted, March 2017.*

[C.50] Z. Michaelides, N. Rakoto, and R. Michaelides, Big Data Driven Demand Networks, Proc. of POMS 2017 Conf., Seattle, WA, USA, May 2017.

[C.49] G. Obando, N. Quijano, and N. Rakoto-Ravalontsalama. "Distributed resource allocation over stochastic networks: An application in smart grids ", Proc. of IEEE CCAC 2015, Manizales, Colombia, Oct 2015.

[C.48] C. Indriago, O. Cardin, O. Morineau, N. Rakoto, P. Castagna, "Performance evaluation of holonic-based online predictive-reactive scheduling for a switch arrival system"' Proc. of INCOM 2015, Ottawa, Canada, May 11-13, 2015. IFAC-PapersOnLine 48-3 (2015) pp. 1105–1110.

[C.47] C. Indriago, O. Cardin, N. Rakoto, E. Chacon, P. Castagna, "Application of holonic paradigm to a water treatment process"' Proc. of SOHOMA 2014, Nancy, France, Nov. 2014, pp. 32–39.

[C.46] C. Indriago, O. Cardin, N. Rakoto, P. Castagna, E. Chacon. "Application du paradigme holonique à un système de reservoirs"' Proc. of MOSIM 2014, Nancy, France, Nov. 2014.

[C.45] G. Obando, N. Quijano, and N. Rakoto-Ravalontsalama. "Distributed Building Temperature Control with Power Constraints ", Proc. of ECC 2014, pp. 2857–2862, Strasbourg, France, June 2014.

[C.44] F. Torres, C. Garcia-Diaz, and N. Rakoto-Ravalontsalama. "An Evolutionary Game Theory Approach to Modeling VMI Policies", Proc. of IFAC World Congress 2014, pp. 10737–10742, Capetown, South Africa, Aug. 2014.

[C.43] M. Canu and N. Rakoto-Ravalontsalama, *From mutually non-blocking to switched non-blocking DES*. Presented at MSR'13 Workshop (Poster Session), Rennes, France, Nov 13-15, 2013.

[C.42] F. Torres, C. Garcia-Diaz, and N. Rakoto-Ravalontsalama, *Evolutionary stability of a manufacturer-buyer VMI-conduced supply chain*, Proc. of POMS 2013 Conf., Denver, Colorado, USA, May 2013.

[C.41] N. Rakoto-Ravalontsalama, *On Stability Analysis of Switched Circulant Systems*, Proc. of MATHMOD 2012, Vienna, Austria, Feb 14-17, 2012.

[C.40] E. Mojica, N. Quijano, and N. Rakoto-Ravalontsalama, *A generalization of a polynomial control of switched systems*, Proc. of IFAC ADHS 2009, Zaragoza, Spain, Sep 2009, pp. 120–125.

[C.39] E. Mojica, N. Quijano, A. Gauthier, and N. Rakoto-Ravalontsalama, *Stability analysis of switched polynomial systems using dissipation inequalities*, Proc. of the 47th IEEE CDC 2008, Cancun, Mexico, Dec 2008, pp. 31–36.

[C.38] E. Mojica, R. Meziat, N. Quijano, A. Gauthier, and N. Rakoto-Ravalontsalama, *Optimal control of switched systems: A polynomial approach.* Proc. of 2008 IFAC World Congress, Seoul, Korea, July 2008, pp. 7808–7813.

[C.37] E. Mojica, A. Gauthier, and N. Rakoto-Ravalontsalama, *Canonical piecewise linear approximation of nonlinear cellular growth.* Proc. of the 46th IEEE CDC 2007 (Conference on Decision and Control), Dec 2007, New Orleans, LA, USA, pp. 1640-1645.

[C.36] E. Mojica, A. Gauthier, and N. Rakoto-Ravalontsalama, *Piecewise linear approximation of nonlinear cellular growth.* Proc. of 2007 IFAC SSSC (Symp. on Systems Structure and Control), Oct 17–19, 2007, Foz do Iguacu, Brazil.

[C.35] E. Mojica, A. Gauthier, and N. Rakoto-Ravalontsalama, *Probing control for PWL approximation of nonlinear cellular growth.* Proc. of 2007 IEEE MSC (Multi-conference on Systems and Control), Oct 1-3, 2007, Singapore.

[C.34] M. Canu, D. Morel, and N. Rakoto-Ravalontsalama. "Modeling and control of an experimental switched manufacturiung system," Proc. of ICINCO 2007, May 2007, Angers, France.

[C.33] M. Canu, J. Haurogne, D. Morel, and N. Rakoto-Ravalontsalama. "Supervisory control of an experimental switched DES," Proc. of IEICE NOLTA 2006, Sep 11-14, 2006, Bologna, Italy, pp. 275–278.

[C.32] N. Rakoto-Ravalontsalama. "Supervisory control of switched discrete-event systems," Proc. of Int. Symposium on MTNS 2006, July 24-28, 2006, Kyoto, Japan, pp. 2213–2217.

[C.31] M. Canu and N. Rakoto-Ravalontsalama, *Flatness Based Control of Switched Systems.* Presented at HSCC'06 Workshop (Poster Session), Santa Barbara, CA, USA, March 29–31, 2006.

[C.30] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Model predictive control of MLD models with integrators.* Proc. of 2005 IEEE CCA Conf. on Control Applications, Aug. 28-31, 2005, Toronto, Canada, pp 641-644.

[C.29] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Commande par MLD d'une usine de traitement d'eau.* Proc. of CIFA 2004, Douz, Tunisie, Nov 22-24, 2004 (in French).

[C.28] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Control of Hybrid Systems using the MLD Approach. Part I: Modelling.* In Proc. of VI Congreso Nacional de la Asociacion Colombiana de Automatica. Ibague, Colombia. 11-13 Nov. 2004 (in Spanish).

[C.27] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Control of Hybrid Systems using the MLD Approach. Part II: Synthesis of Control.* In Proc. of VI Congreso Nacional de la Asociacion Colombiana de Automatica. Ibague, Colombia. 11-13 Nov. 2004 (in Spanish).

[C.26] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, "Translating PWA systems into MLD systems". In Proc. of 2004 IEEE CCA/CASCD/ISIC Conf., Taipei, Taiwan, Sep 2-4, 2004, pp. 37-42.

[C.25] X. Chen, D. Morel, and N. Rakoto-Ravalontsalama *Multi-Agent Based Supervisory Control of an Experimental Manufacturing Cell.* Proc. of IFAC Symposium on Large Scale Systems (LSS 2004) Osaka, Japan, July 26-28, 2004, pp. 391–394.

[C.24] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *A new algorithm for translating MLD systems into PWA systems.* Proc. of IEEE American Control Conference (ACC 2004), June 30 - July 2, 2004, Boston, MA, USA, pp. 1208–1213.

[C.23] S. Esteva, J.L. de la Rosa, and N. Rakoto-Ravalontsalama, *Modeling Hybrid Systems by Piecewise Decomposition.* In Proc. of IEEE Conf. on Systems, Man and Cybernetics (SMC 2003), Washington DC, USA, Oct. 2003, pp. 195–198.

[C.22] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *MLD Control of Hybrid Systems: Application to the Three-Tank Benchmark Problem.* In Proc. of IEEE Conf. on Systems, Man and Cybernetics (SMC 2003), Washington DC, USA, Oct. 2003, pp. 666-671.

[C.21] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Modeling and Control of a Water Treatment Plant.* In Proc. of IEEE Conf. on Systems, Man and Cybernetics (SMC 2003), Washington DC, USA, Oct. 2003, pp. 171-176.

[C.20] N. Rakoto-Ravalontsalama, J.L. Villa, and D. Morel, *Supervisory Control Oriented Modeling of an Experimental Manufacturing Cell.* Proc. of IEEE Int. Symposium on Intelligent Control (ISIC 2003) Houston, TX, USA, Oct. 2003, pp. 395–398.

[C.19] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Supervision and Optimal Control of a Class of Industrial Processes.* In Proc. of IEEE Conf. on Emerging Technologies and Factory Automation (ETFA 2003), Lisbon, Portugal, Sep. 2003, vol.2, pp. 177–180.

[C.18] N. Rakoto-Ravalontsalama and J.L. Villa, *Modeling and simulation of an experimental cell.* Proc. of International Conf. on Computational Science and its Applications (ICCSA 2003), Montreal, Canada, May 2003, LNCS N. 2667, Springer Verlag, pp. 533-538.

[C.17] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama, *Hybrid modeling of potable water treatment plant.* In Proc. of Pumps, Electromechanical Devices and Systems (PEDS 2003), April 2003, Valencia, Spain, paper N.10089.

[C.16] N. Rakoto-Ravalontsalama. *Controllability Issues in Supervisory Control Systems with Order Relations.* Proc. of IEEE Conf. on Systems, Man and Cybernetics (SMC 2002), Hammamet, Tunisia, Oct. 2002, CD-ROM, paper MP1A1.

[C.15] N. Rakoto-Ravalontsalama, *Partial and total order relations for supervisory control of hybrid systems.* Proc. of IEEE Emerging Technologies and Factory Automation (ETFA 2001) Antibes, France, Oct 2001, vol.2, pp. 675–678.

[C.14] N. Rakoto-Ravalontsalama, *Modeling of continuous and hybrid systems using partial order relations.* Proc. of IEEE Int. Symposium on Intelligent Control (ISIC 2001) Mexico City, Mexico, Sep. 2001, pp. 156–160.

[C.13] N. Rakoto-Ravalontsalama, *Discrete approximation of continuous and hybrid systems: Some consequences to supervisory control.* Proc. of IFAC Symposium on System Structure and Control (SSSC 2001), Prague, Czech Republic, Aug. 2001.

[C.12] N. Rakoto-Ravalontsalama, *Diagnosis algorithms for a symbolically modeled manufacturing process.* Proc. of International Conf. on Computational Science (ICCS 2001), LNCS N. 2073, Springer Verlag, San Fransisco, CA, USA, May 2001, pp. 1228–1236.

[C.11] N. Rakoto-Ravalontsalama, *Knowledge-based process control.* Proc. of 1999 IEEE Int. Symposium on Intelligent Control (ISIC 1999), Cambridge, MA, USA, Sep. 1999, pp. 237–241.

[C.10] N. Rakoto-Ravalontsalama, *Mixed qualitative/quantitative modeling for process diagnosis.* Proc. of 13th European Simulation Multiconference (ESM 1999), Warsaw, Poland, June 1999, vol.2, pp. 414–418.

[C.9] L. Libeaut and N. Rakoto-Ravalontsalama, "Modeling and analysis of a manufacturing cell: Petri net vs. automata approach", Proc. of WESIC'98 Conf., Girona, Spain, June 1998, pp. 71–78.

[C.8] N. Rakoto-Ravalontsalama and J. Aguilar-Martin, *Diagnosing uncertain parameters to improve hybrid process model.* Proc. of 3rd, International Conf. on Automation of Mixed Processes: Dynamic Hybrid Systems (ADPM'98), Reims, France, March 1998, pp. 49–53.

[C.7] N. Rakoto-Ravalontsalama and J. Aguilar-Martin, *Control of hybrid systems: The expert system approach.* In Proc. of IEEE/IMACS Computational Engineering in Systems Applications (CESA 1996) Symp. on Discrete Event and Manufacturing Systems, Lille, France, July 1996, pp. 402–406.

[C.6] N. Rakoto-Ravalontsalama, *Process supervision with expert systems: The Esprit IPCES experience*, in "Supervision de Processus à l'Aide du Système Expert G2", Rakoto-Ravalontsalama and Aguilar-Martin (Eds.), Hermes Ed. Paris 1995, pp. 11–20.

[C.5] N. Rakoto-Ravalontsalama and J. Aguilar-Martin, *Modelling and simulation of hybrid systems: The expert system approach*, Analysis and Design of Event-Driven Operations in Process Systems (ADEDOPS'95) Workshop, London, UK, April 10-11, 1995.

[C.4] N. Rakoto-Ravalontsalama and J. Aguilar-Martin, *Knowledge-based modelling of a TV-tube manufacturing process.* In Proc. of IFAC Workshop on Computer Software Structures Integrating AI/KBS in Process Control, Lund, Sweden, Aug. 1994, pp. 111–117.

[C.3] J. Aguilar-Martin and N. Rakoto-Ravalontsalama, *Uncertainty propagation in fuzzy simulation of dynamic system – Application to a simplified turbine.* Proc. of SCS IFQN'94 Conf., Barcelona, Spain, June 1994.

[C.2] N. Rakoto-Ravalontsalama and J. Aguilar-Martin (1992), *Automatic clustering for symbolic evaluation for dynamical system supervision.* In Proc. of Conf. Canadienne sur l'Automatisation (CCA 1992), Montreal, Canada, June 1992, vol. 1 pp. 2.9–2.12.

[C.1] N. Rakoto-Ravalontsalama and J. Aguilar-Martin, *Automatic clustering for symbolic evaluation for dynamical system supervision.* In Proc. of IEEE American Control Conference (ACC 1992), Chicago, USA, June 1992, vol. 3, pp. 1895–1897.

Chapter 2

Analysis and Control of Hybrid and Switched Systems

2.1 Modeling and Control of MLD systems

Piecewise affine (PWA) systems have been receiving increasing interest, as a particular class of hybrid system, see e.g. [2], [13], [11], [16], [14], [12] and references therein. PWA systems arise as an approximation of smooth nonlinear systems [15] and they are also equivalent to some classes of hybrid systems e.g. Linear complementarity systems [9]. On the other hand Mixed Logical and Dynamical (MLD) systems have been introduced by Bemporad and Morari as a suitable representation for hybrid dynamical systems [3]. MLD models are obtained originally from PWA system, where propositional logic relations are transformed into mixed-integer inequalities involving integer and continuous variables. Then mixed-integer optimization techniques are applied to the MLD system in order to stabilize MLD system on desired reference trajectories under some constraints. Equivalences between PWA systems and MLD models have been established in [9]. More precisely, every well-posed PWA system can be rewritten as an MLD system assumung that the set of feasible states and inputs is bounded and a completely well-posed MLD system can be rewritten as a PWA system [9]. Conversion methods from MLD systems to equivalent PWA models have been proposed in [4], [5], [6] and [?]. Vice versa, translation methods from PWA to MLD systems have been studied in [3] (the original one), and then in [8], [?]. A tool that deals with both MLD and PWA systems is HYSDEL [17].
The motivations for studying new methods of conversion from PWA systems into their equivalent MLD models are the following. Firstly the original motivation of obtaining MLD models is to rewrite a PWA system into a model that allows the designer to use existing optimization algorithms such as mixed integer quadratic programming (MIQP) or mixed integer linear programmimg (MILP). Secondly there is no unique formulation of PWA systems. We can always address some particular cases that introduce some differences in the conversions. Finally, it has been shown that the stability analysis of PWA systems with two polyhedral regions is in general NP-complete or undecidable [7]. The conversion to MLD systems may be another way to tackle this problem.

2.1.1 Piecewise Affine (PWA) Systems

A particular class of hybrid dynamical systems is the system described as follows.

$$\begin{cases} \dot{x}(t) &= A_i x(t) + a_i + B_i u(t) \\ y(t) &= C_i x(t) + c_i + D_i u(t) \end{cases} \tag{2.1}$$

where $i \in \mathcal{I}$, the set of indexes, $x(t) \in X_i$ which is a sub-space of the real space $\mathbb{R}^n$, and $\mathbb{R}^+$ is the set of positive real numbers including the zero element. In addition to this equation it is necessary to define the form as the system switches among its several modes. This equation is affine in the state space x and the systems described in this form are called piecewise affine (PWA) systems

[15], [9]. The discrete-time version of this equation will be used in this work and can be described as follows.

$$\begin{cases} x(k+1) & = & A_i x(k) + b_i + B_i u(k) \\ y(k) & = & C_i x(k) + d_i + D_i u(k) \end{cases} \tag{2.2}$$

where $i \in \mathcal{I}$ is a set of indexes, X_i is a sub-space of the real space $\mathbb{R}^n$, and $\mathbb{R}^+$ is the set of positive integer numbers including the zero element, or an homeomorphic set to $\mathbb{Z}^+$.

2.1.2 Mixed Logical Dynamical (MLD) Systems

The idea in the MLD framework is to represent logical propositions with the equivalent mixed integer expressions. MLD form is obtained in three steps [3], [4]. The first step is to associate a binary variable $\delta \in \{0, 1\}$ with a proposition S, that may be true or false. δ is equal to 1 if and only if proposition S is true. A composed proposition of elementary propositions $S_1, \ldots, S_q$ combined using the boolean operators like AND, OR, NOT may be expressed with integer inequalities over corresponding binary variables $\delta_i, i = 1, \ \ldots q$. The second step is to replace the products of linear functions and logic variables by a new auxiliary variable $z = \delta a^T x$ where a^T is a constant vector. The variable z is obtained by mixed linear inequalities evaluation. The third step is to describe the dynamical system, binary variables and auxiliary variables in a linear time invariant system. An hybrid system described in MLD form is represented by Equations (2.3-2.5).

$$x(k+1) = Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k) \tag{2.3}$$

$$y(k) = Cx(k) + D_1 u(k) + D_2 \delta(k) + D_3 z(k) \tag{2.4}$$

$$E_2 \delta(k) + E_3 z(k) \leq E_1 u(k) + E_4 x(k) + E_5 \tag{2.5}$$

where $x = [x_c^T x_l^T] \in \mathbf{R}^{n_c} \times \{0, 1\}^{n_l}$ are the continuous and binary states, respectively, $u = [u_C^T u_l^T] \in \mathbf{R}^{m_c} \times \{0, 1\}^{m_l}$ are the inputs, $y = [y_c^T y_l^T] \in \mathbf{R}^{p_c} \times \{0, 1\}^{p_l}$ the outputs, and $\delta \in \{0, 1\}^{r_l}$, $z \in \mathbf{R}^{r_c}$, represent the binary and continuous auxiliary variables, respectively. The constraints over state, input, output, z and δ variables are included in (2.5).

2.1.3 Converting PWA into MLD Systems

In this subsection two algorithms for converting PWA systems into MLD systems are given. The first case consists of several sub-affine systems with switching regions are explained in detail. The second case deals with several sub-affine systems, each of them belongs to a region which is described by linear inequalities is a variation of the first case. Each case is applied to an example in order to show the validity of the algorithm.

A. Case I

The PWA system is represented by the following equations:

$$\begin{cases} x(k+1) & = & A_i x(k) + B_i u(k) + f_i \\ y(k) & = & C_i x(k) + D_i u(k) + g_i \\ S_{ij} & = & \{x, u | k_{1ij}^T x + k_{2ij}^T u + k_{3ij} \leq 0\} \end{cases} \tag{2.6}$$

where $i \in I = \{1, \ldots, n\}$. The case with jumps can be included in this representation considering each jump as a discrete affine behavior valid during only one sample time. The switching region S_{ij} is a convex polytope which volume, or hypervolume, can be infinite, and the sub-scripts denotes the switching from mode i to mode j. For this purpose we introduce a binary variable δ_i for each index of the set $\mathcal{I}$ and a binary variable $\delta_{i,j}$ for each switching region S_{ij}. In order to gain insight in the following equations, we consider the hybrid the partition and the corresponding automaton is depicted in Figure 2.1. Introductory material on hybrid automata can be found in [1] and [10].

Figure 2.1: Partition and Automaton

The δ_{ij} variables are not dynamical and, when the elements k in S_{ij} are vectors, the binary variable can be evaluated by the next mixed integer inequality

$$(\delta_{ij} = 1) \Leftrightarrow (k_{1ij}^T x + k_{2ij}^T u + k_{3ij} \leq 0) \tag{2.7}$$

which is equivalent to:

$$\left\{ \begin{array}{lcl} k_{1ij}x + k_{2ij}u + k_{3ij} - M(1-\delta_{ij}) & \leq & 0 \\ -k_{1ij}x - k_{2ij}u - k_{3ij} + \epsilon + (m-\epsilon)\delta_{ij} & \leq & 0 \end{array} \right. \tag{2.8}$$

When the elements k in S_{ij} are matrices, it is necessary to introduce some auxiliary binary variables for each row describing a sub-constraint in S_{ij} in the next form:

$$\begin{array}{lcl} \delta_k & = & 1(\Leftrightarrow k_{1,k}x + k_{2,k}u + k_{3,k} \leq 0) \\ \delta_{ij} & = & \bigwedge_k \delta_k \end{array} \tag{2.9}$$

which is equivalent to:

$$\left\{ \begin{array}{rcl} k_{1ij,k}x + k_{2ij,k}u + k_{3ij,k} - M(1-\delta_{ij,k}) & \leq & 0 \\ -k_{1ij,k}x - k_{2ij,k}u - k_{3ij,k} + \epsilon + (m-\epsilon)\delta_{ij,k} & \leq & 0 \\ \delta_{ij} - \delta_{ij,k} & \leq & 0 \\ \sum_k(\delta_{ij,k} - 1) - \delta_{ij} & \leq & -1 \end{array} \right. \tag{2.10}$$

The binary vector $x_\delta = [\delta_1 \delta_2 \ldots \delta_n]^T$ is such that its dynamics is given by:

$$x_{\delta i}(k+1) = (x_{\delta i}(k) \wedge \bigwedge_{j \neq i} \neg \delta ij) \vee \bigvee_{j \neq i} (x_{\delta j}(k) \wedge \delta ji) \tag{2.11}$$

where k is an index of time, and $\wedge$, $\vee$, and $\neg$, are standard for the logical operations AND, OR, NOT, respectively. This equation can be explained as follows: The mode of the system in the next time is i if the current mode is mode i and any switching region is enabled in this time, or, the current mode of the system is j different to i and a switching region that enables the system to go into mode i is enabled. Considering that the PWA system is well posed, i.e. for a given initial state $[x^T i^T]^{T_0}$ and a given input $u_{0,\tau}$ there exists only one possible trajectory $[x^T i^T]^{T_{0,x}}$. That is equivalent to the following conditions:

$$\sum_{i \in I} x_{\delta i} = 1, \quad \prod_{i \in I} x_{\delta i} = 0 \tag{2.12}$$

The dynamical equations for x_δ vector are equivalent to the next integer inequalities:

$$\left\{ \begin{array}{rcl} x_{\delta j}(k) + \delta_{ji} - x_{\delta i}(k+1) & \leq & 1, \quad \forall i,j \in I, i \neq j \\ x_{\delta i}(k) - \sum_{j \neq i} \delta_{ij} - x_{\delta i}(k+1) & \leq & 0, \quad \forall i,j \in I, i \neq j \\ -x_{\delta i}(k) - \sum_{j \neq i} \delta_{ji} - x_{\delta i}(k+1) & \leq & 0, \quad \forall i,j \in I, i \neq j \end{array} \right. \tag{2.13}$$

The first inequality states that the next mode of the system should be mode i if the current mode is j different to i and a switching region for going from mode j to mode i is enabled. The second inequality means that the next mode of the system should be mode i if the current mode is i and any switching region for going from mode i into mode j different to i is enabled. And the third equation states that the system cannot be in mode i in the next time if the current mode of the system is not mode i and any switching region for going from mode i, (j different to i), into mode i is enabled.

This form for finding $x_\delta(k+1)$ causes a problem in the final model because it cannot be represented by a linear equation in function of x, u, δ and Z. For this reason, $x_\delta(k+1)$ is aggregated to the δ general vector of binary variables, and finally assigned directly to $x_\delta(k+1)$. The dynamics and outputs of the system can be represented by the next equations:

$$\left\{\begin{array}{lcl} x(k+1) & = & Ax(k)+Bu(k)+\sum_{i\in I}(A_ix(k)+B_iu(k)+f_i)\times x_{\delta i}(k) \\ y(k) & = & Cx(k)+Du(k)+\sum_{i\in I}(C_ix(k)+D_iu(k)+g_i)\times x_{\delta i}(k) \end{array}\right. \tag{2.14}$$

If we introduce some auxiliary variables:

$$\left\{\begin{array}{lcl} Z_{1i}(k) & = & (A_ix(k)+B_iu(k)+f_i)\times x_{\delta i}(k) \\ Z_{2i}(k) & = & (C_ix(k)+D_iu(k)+g_i)\times x_{\delta i}(k) \end{array}\right. \tag{2.15}$$

which are equivalent to:

$$\left\{\begin{array}{lcl} Z_{1i} & \leq & Mx_{\delta i}(k) \\ -Z_{1i} & \leq & -mx_{\delta i}(k) \\ Z_{1i} & \leq & A_ix(k)+B_iu(k)+f_i-m(1-x_{\delta i}(k)) \\ -Z_{1i} & \leq & -A_ix(k)-B_iu(k)-f_i+M(1-x_{\delta i}(k)) \end{array}\right. \tag{2.16}$$

$$\left\{\begin{array}{lcl} Z_{2i} & \leq & Mx_{\delta i}(k) \\ -Z_{2i} & \leq & -mx_{\delta i}(k) \\ Z_{2i} & \leq & C_ix(k)+D_iu(k)+g_i-m(1-x_{\delta i}(k)) \\ -Z_{2i} & \leq & -C_ix(k)-D_iu(k)-g_i+M(1-x_{\delta i}(k)) \end{array}\right. \tag{2.17}$$

where M and m are vectors representing the maximum and minimum values, respectively, of the variables Z, these values can be arbitrary large. Using the previous equivalences, the PWA system (2.2) can be rewritten in an equivalent MLD model as follows:

$$\left\{\begin{array}{rcl} x(k+1) & = & A_{rr}x(k)+A_{br}x_\delta(k)+B_{1r}u(k)+B_{2r}\delta+B_{3r}\sum_{i\in I}Z_{1i}(k) \\ x_\delta(k+1) & = & A_{rb}x(k)+A_{bb}x_\delta(k)+B_{1b}u(k)+B_{2b}\delta+B_{3b}\sum_{i\in I}Z_{1i}(k) \\ y_r(k) & = & C_{rr}x(k)+C_{br}x_\delta(k)+D_{1r}u(k)+D_{2r}\delta+D_{3r}\sum_{i\in I}Z_{2i}(k) \\ y_\delta(k) & = & C_{rb}x(k)+C_{bb}x_\delta(k)+D_{1b}u(k)+D_{2b}\delta+D_{3b}\sum_{i\in I}Z_{2i}(k) \end{array}\right. \tag{2.18}$$

s.t.

$$E_2\begin{bmatrix} x_\delta(k+1) \\ \delta_{ij} \\ \delta_k \end{bmatrix}+E_3Z(k)\leq E_4\begin{bmatrix} x(k) \\ \delta_{ij} \\ \delta_k \end{bmatrix}+E_1u(k)+E_5 \tag{2.19}$$

Using this algorithm, most part of the matrices are zero, because x and y are defined by Z, and x_δ is defined by δ. This situation can be avoided by defining the next matrices at the beginning of the procedure:

$$\left\{\begin{array}{ll} A=\frac{1}{n}(A_1+\ldots+A_n), & \overline{A_i}=A_i-A,\ \ \forall i\in I \\ B=\frac{1}{n}(B_1+\ldots+B_n), & \overline{B_i}=B_i-B,\ \ \forall i\in I \\ C=\frac{1}{n}(C_1+\ldots+C_n), & \overline{C_i}=C_i-C,\ \ \forall i\in I \\ D=\frac{1}{n}(D_1+\ldots+D_n), & \overline{D_i}=D_i-D,\ \ \forall i\in I \end{array}\right. \tag{2.20}$$

Finally, the equality matrices in (2.18) and (2.19) can be chosen as follows:

$$\left\{\begin{array}{l} A_{rr}=A,\ A_{br}=0_{n_c\times n},\ B_{1r}=B,\ B_{2r}=0_{n_c\times(n+m+tk)}, \\ B_{3r}=[I_{n_c\times n_c}0_{n_c\times p_c}I_{n_c\times n_c}0_{n_c\times p_c}\ldots I_{n_c\times n_c}0_{n_c\times p_c}]_{n_c\times n\times(n_c+p_c)} \\ A_{rb}=0_{n\times n_c},\ A_{bb}=0_{x\times n},\ B_{1b}=0_{n\times m_c}, \\ B_{2b}=[I_{n\times n}0_{n\times m}0_{n\times tk}]_{n\times n\times(n+m+tk)},\ B_{3b}=0_{n_c\times n\times(n_c+p_c)} \end{array}\right. \tag{2.21}$$

$$\left\{\begin{array}{l} C_{rr}=C,\ C_{br}=0_{p_c\times n},\ D_{1r}=D,\ D_{2r}=0_{p_c\times(n+m+tk)},\\ D_{3r}=[0_{p_c\times n_c}I_{p_c\times p_c}0_{p_c\times n_c}I_{p_c\times p_c}\dots 0_{p_c\times n_c}I_{p_c\times p_c}]_{p_c\times n\times(n_c+p_c)}\\ C_{rb}=0_{n\times n_c},\ C_{bb}=0_{x\times n},\ D_{1b}=0_{n\times m_c},\\ D_{2b}=[I_{n\times n}0_{n\times m}0_{n\times tk}]_{n\times n\times(n+m)},\ D_{3b}=0_{n\times n\times(n_c+p_c)} \end{array}\right. \tag{2.22}$$

where n_C is the number of continuous state variables, m_C the number of continuous input variables, p_C the number of continuous output variables, n the number of affine sub-systems, m the number of switching regions and tk the number of auxiliary binary variables. The algorithm for converting a PWA system in the form of (2.1) into its equivalent MLD system can be summarized as follows:

B. Algorithm 1

1. Compute matrices A, B, C, D and $\overline{A_i}$, $\overline{B_i}$, $\overline{C_i}$ and $\overline{D_i}$ using (2.20).
2. Initialize E_1, E_2, E_3, E_4, E_5 matrices.
3. For the m switching regions $S_{j,i}$, include the inequalities defined in (2.8) or (2.10) which define the values of the m auxiliary binary variables $\delta_{j,i}$.
4. Generate $2*nx_{\delta i}$ auxiliary binary dynamical variables associated with the n affine models and m auxiliary binary variables $\delta_{j,i}$ associated with the m S_{ij} switching regions.
5. For $i = 1$ *to* n include the inequalities using (2.13) representing the behavior on the x_δ vector.
6. For $i = 1$ *to* n generate the n_c-dimensional Z_{1i} vector and p_c-dimensional Z_{2i} vector of auxiliary variables Z.
7. For each Z_{1i} vector introduce the inequalities defined in (2.16), by replacing A_i, and B_i by $\overline{A_i}$, and $\overline{B_i}$, computed in Step 1. M and m are n_c-dimensional vectors of maximum and minimum values of x, respectively.
8. For each Z_{2i} vector introduce the inequalities defined in (2.17), by replacing C_i, and D_i by $\overline{C_i}$, and $\overline{D_i}$, computed in Step 1. M and m are p_c-dimensional vectors of maximum and minimum values of x, respectively *(This completes the inequality matrices)*.
9. Compute the matrices defined in (2.21) and (2.22)
10. End.

C. Example 1

Consider the system whose behavior is defined by the following PWA model:

$$\left\{\begin{array}{lcl} x(k+1) & = & A_i x(k),\quad i\in\{1,2\}\\ S_{1,2} & = & \{(x_1,x_2)|(x_1\le 1.3x_2)\wedge(0.7x_2\le x_1)\wedge(x_2>0)\}\\ S_{2,1} & = & \{(x_1,x_2)|(x_1\le 0.7x_2)\wedge(1.3x_2\le x_1)\wedge(x_2<0)\} \end{array}\right.$$

where $A_1 = \begin{bmatrix} 0.9802 & 0.0987 \\ -0.1974 & 0.9802 \end{bmatrix}$, $A_2 = \begin{bmatrix} 0.9876 & -0.0989 \\ 0.0495 & 0.9876 \end{bmatrix}$ The behavior of the system is presented in Figure 2.2. The initial points are $(x_{10}, x_{20}) = (1, 0.8)$. We can see that the system switches between the two behaviors, from A_1 to A_2 in the switching region $S_{1,2}$, and from A_2 to A_1 in the switching region $S_{2,1}$, alternatively. The switched system is stable.

D. Case 2

Consider now the system whose behavior is defined by the following PWA model:

$$\left\{\begin{array}{lcl} x(k+1) & = & A_i x(k)+b_i+B_i u(k),\quad i\in I, x(k)\in X_i\\ y(k) & = & C_i x(k)+d_i+D_i u(k),\quad i\in I, x(k)\in X_i \end{array}\right. \tag{2.23}$$

with conditions $X_i \cap X_{j\neq i} = \emptyset$, $\forall i,j \in \mathcal{I}, \bigcup_{i\in\mathcal{I}} X_i = X$, where X is the admissible space for the PWA system, and $X_i = \{x, u|k_{1i}x + k_{2i}u + k_{3i} \le 0\}$ does not need the dynamical binary

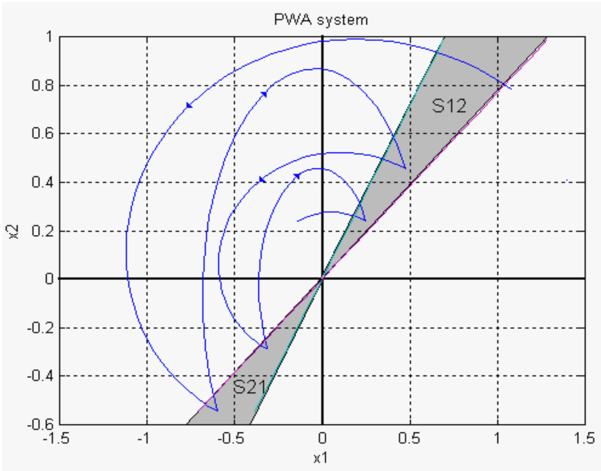


Figure 2.2: Phase portrait of Example 1 in PWA

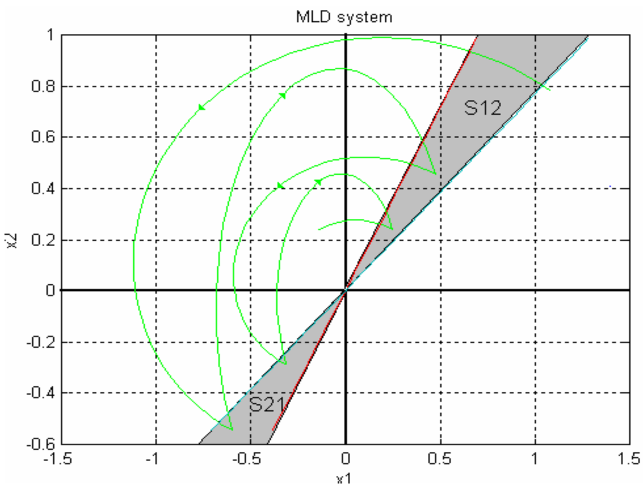


Figure 2.3: Phase portrait of Example 1 MLD

variables and can be represented using the appropriate δ variables instead of $x_\delta(k)$ variables in the definition of Z in (2.16) and (2.17). However, note that the conditions $X_i \cap X_{j\neq i} = \emptyset \ \forall i, j \in \mathcal{I}$ and $\bigcup_{i\in\mathcal{I}} X_i = X$ require a careful definition in the sub-spaces X_i in order to avoid a violation to these conditions in its bounds. On the other hand, the MLD representation uses non-strict inequalities in its representation and the ε value in (2.8) and (2.9) should be chosen appropriately. Another way to overcome this situation and to insure an appropriated representation is the use of the following conditions in the bounds of the sub-spaces X_i:

$$\delta_{ij} = \delta_i \otimes \delta_j$$

which is equivalent to:

$$\begin{cases} \delta_i + \delta_j - 1 & \leq & 0 \\ 1 - \delta_i - \delta_j & \leq & 0 \end{cases} \qquad \text{or more generally} \qquad \begin{cases} \displaystyle\sum_{i\in\mathcal{I}} \delta_i - 1 & \leq & 0 \\ 1 - \displaystyle\sum_{i\in\mathcal{I}} \delta_i & \leq & 0 \end{cases} \tag{2.24}$$

We now modify Equations (2.8), (2.10), (2.16), (2.17), (2.21), and (2.22) as follows:

$$\begin{cases} k_{1i}x + k_{2i}u + k_{3i} - M(1-\delta_i) & \leq & 0 \\ -k_{1i}x - k_{2i}u - k_{3i} + \epsilon + (m-\epsilon)\delta_i) & \leq & 0 \end{cases} \tag{2.25}$$

$$\begin{cases} k_{1i,k}x + k_{2i,k}u + k_{3i,k} - M(1-\delta_{i,k}) & \leq & 0 \\ -k_{1i,k}x - k_{2i,k}u - k_{3i,k} + \epsilon + (m-\epsilon)\delta_{i,k} & \leq & 0 \\ \delta_i - \delta_{i,k} & \leq & 0 \\ \sum_k(\delta_{i,k}-1) - \delta_i & \leq & -1 \end{cases} \tag{2.26}$$

The auxiliary variables Z_{1i} become:

$$\begin{cases} Z_{1i} & \leq & M\delta_i(k) \\ -Z_{1i} & \leq & -m\delta_i(k) \\ Z_{1i} & \leq & \overline{A_i}x(k) + \overline{B_i}u(k) + f_i - m(1-\delta_i(k)) \\ -Z_{1i} & \leq & -\overline{A_i}x(k) - \overline{B_i}u(k) - f_i + M(1-\delta_i(k)) \end{cases} \tag{2.27}$$

where the matrices $\overline{A_i}$ and $\overline{B_i}$ are those previously defined in Equation (2.20).The auxiliary variable Z_{2i} is now modified according to the following equations:

$$\begin{cases} Z_{2i} & \leq & M\delta_i(k) \\ -Z_{2i} & \leq & -m\delta_i(k) \\ Z_{2i} & \leq & \overline{C_i}x(k) + \overline{D_i}u(k) + g_i - m(1-\delta_i(k)) \\ -Z_{2i} & \leq & -\overline{C_i}x(k) - \overline{D_i}u(k) - g_i + M(1-\delta_i(k)) \end{cases} \tag{2.28}$$

where the matrices $\overline{C_i}$ and $\overline{D_i}$ are those that have been defined in Equation (2.20). Finally the matrices from Equation (2.18) can be chosen as follows:

$$\begin{cases} A_{rr} = A,\ A_{br} = 0_{n_c\times n},\ B_{1rr} = B,\ B_{2rb} = 0_{n_c\times(n+tk)}, \\ B_{3rr} = [I_{n_c\times n_c}0_{n_c\times p_c}I_{n_c\times n_c}0_{n_c\times p_c}\dots I_{n_c\times n_c}0_{n_c\times p_c}]_{n_c\times n\times(n_c+p_c)} \\ C_{rr} = C,\ C_{br} = 0_{p_c\times n},\ D_{1rr} = D,\ D_{1rb} = [\ ],\ D_{2rb} = 0_{p_c\times(n+tk)}, \\ D_{3rr} = [0_{p_c\times n_c}I_{p_c\times p_c}0_{p_c\times n_c}I_{p_c\times p_c}\dots 0_{p_c\times n_c}I_{p_c\times p_c}]_{p_c\times n\times(n_c+p_c)} \end{cases} \tag{2.29}$$

We give now an algorithm that converts a PWA system in the form of (2.23) into its equivalent MLD system.

E. Algorithm 2

1. Compute matrices A, B, C, D and $\overline{A_i}$, $\overline{B_i}$, $\overline{C_i}$ and $\overline{D_i}$ using (2.20).
2. Initialize E_1, E_2, E_3, E_4, E_5 matrices.
3. For $i = 1$ *to* n include the inequalities using (2.25) or (2.26) that represent the behavior on the n affine regions of the PWA system.
4. For all affine regions include the inequalities in (2.24).
5. For $i = 1$ *to* n generate the nc-dimensional Z_{1i} vector and p_c -dimensional Z_{2i} vector of auxiliary variables Z.
6. For each Z_{1i} vector introduce the inequalities defined in (2.27). M and m are n_c-dimensional vectors of maximum and minimum values of x, respectively.
7. For each Z_{2i} vector introduce the inequalities defined in (2.28). M and m are p_c-dimensional vectors of maximum and minimum values of x, respectively *(This completes the inequality matrices)*.
8. Compute the matrices defined in (2.29) where the binary state variables are removed.
9. End.

F. Example 2

Consider the system whose behavior is defined by the following PWA model:

$$\begin{cases} x(k+1) & = & A_i x(k),\quad i\in\{1,2\} \\ i & = & 1\ \ if\ \ x_1x_2 \geq 0 \\ i & = & 2\ \ if\ \ x_1x_2 < 0 \end{cases}$$

where $A_1 = \begin{bmatrix} 0.9960 & 0.0199 \\ -0.1995 & 0.9960 \end{bmatrix}$, $A_2 = \begin{bmatrix} 0.9960 & 0.1995 \\ -0.0199 & 0.9960 \end{bmatrix}$ The behavior of the system is presented in Figure 2.4. The PWA system with linear constraints has 4 sub-affine systems. Algorithm 2 produces an MLD system with 12 binary variables (4 variables for the affine sub-system, and 8 auxiliary variables), 16 auxiliary variables Z and 94 constraints.

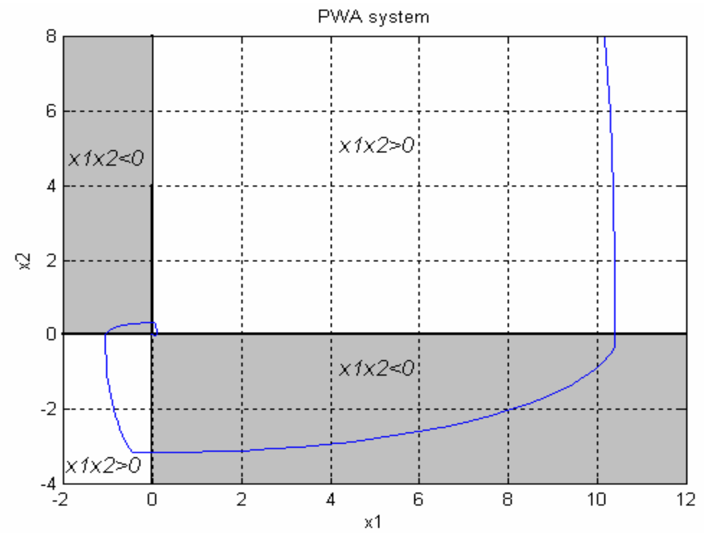


Figure 2.4: Phase portrait of Example 2 in PWA

The behavior of the equivalent MLD system is shown in Figure 2.5. We can notice that the behavior of the MLD system is exactly the same as the original PWA model.

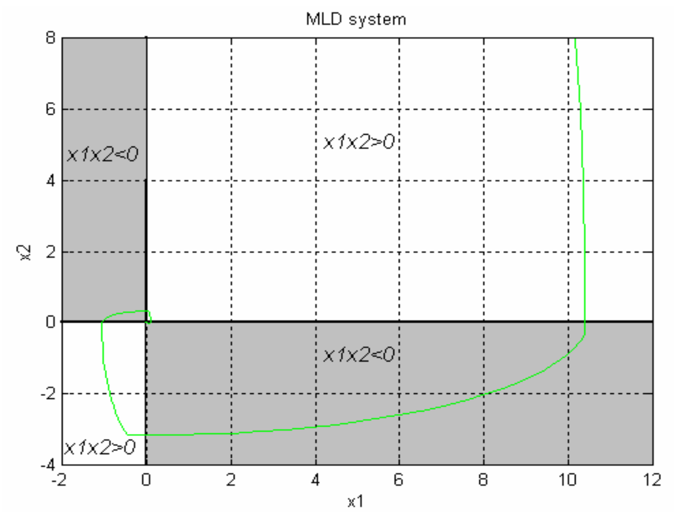


Figure 2.5: Phase portrait of Example 2 in MLD

2.2 Stability of Switched Systems

A polynomial approach to deal with the stability analysis of switched non-linear systems under arbitrary switching using dissipation inequalities is presented. It is shown that a representation of the original switched problem into a continuous polynomial system allows us to use dissipation inequalities for the stability analysis of polynomial systems. With this method and from a theoretical point of view, we provide an alternative way to search for a common Lyapunov function for switched non-linear systems. We deal with the stability analysis of switched non-linear systems, i.e., continuous systems with switching signals under arbitrary switching. Most of the efforts in switched systems research have been typically focused on the analysis of dynamical behavior with respect to switching signals. Several methods have been proposed for stability analysis (see [53], [19], and references therein), but most of them have been focused on switched linear systems. Stability analysis under arbitrary switching is a fundamental problem in the analysis and design of switched systems. For this problem, it is necessary that all the subsystems be asymptotically stable. However, in general, the above stability condition is not sufficient to guarantee stability for the switched system under arbitrary switching. It is well known that if there exists a common Lyapunov function for all the subsystems, then the stability of the switched system is guaranteed under arbitrary switching. Previous attempts for general constructions of a common Lyapunov function for switched non-linear systems have been presented in [20], [21] using converse Lyapunov theorems. Also in [22], a construction of a common Lyapunov function is presented for a particular case when the individual systems are handled sequentially rather than simultaneously for a family of pairwise commuting systems. These methodologies are presented in a very general framework, and even though they are mathematically sound, they are too restrictive from a computational point of view, mainly because it is usually hard to check for the set of necessary conditions for a common function over all the subsystems (it could not exist). Also, these constructions are usually iterative, which involves running backwards in time for all possible switching signals, being prohibitive when the number of modes increases.
The main contribution of this topic of stability of switched systems is twofold. First, we present a reformulation of the switched system as an ordinary differential equation on a constraint manifold. This representation opens several possibilities of analysis and design of switched systems in a consistent way, and also with numerical efficiency [C.39], [C.38], which is possible thanks to some tools developed in the last decade for polynomial differential-algebraic equations analysis [8,10]. The second contribution is an efficient numerical method to search for a common Lyapunov function for switched systems using results of stability analysis of polynomial systems based on dissipativity theory [23], [C.39]. We propose a methodology to construct common Lyapunov functions that provides a less conservative test for proving stability under arbitrary switching. It has been mentioned in [26] that the sum of squares decomposition, presented only for switched polynomial systems, can sometimes be made for a system with a non-polynomial vector fields. However, those cases are restricted to subsystems that preserve the same dimension after a recasting process.

2.3 Optimal Control of Switched Systems

2.3.1 Switched Linear Systems

A polynomial approach to solve the optimal control problem of switched systems is presented. It is shown that the representation of the original switched problem into a continuous polynomial systems allow us to use the method of moments. With this method and from a theoretical point of view, we provide necessary and sufficient conditions for the existence of minimizer by using particular features of the minimizer of its relaxed, convex formulation. Even in the absence of classical minimizers of the switched system, the solution of its relaxed formulation provide minimizers.

We consider the optimal control problem of switched systems, i.e., continuous systems with switching signals. Recent efforts in switched systems research have been typically focused on the analysis of dynamic behaviors, such as stability, controllability and observability, etc. (e.g., [19], [53]). Although there are several studies facing the problem of optimal control of switched systems (both from theoretical and from computational point of view [37], [36], [27], [39], there are still some problems not tackled, especially in issues where the switching mechanism is a design variable. There, we see how these difficulties arise, and how tools from non-smooth calculus and optimal control can be combined to solve optimal control problems. Previously, the approach based on convex analysis have been treated in [36], and further developed in [27], considering an optimal control problem for a switched system, these approaches do not take into account assumptions about the number of switches nor about the mode sequence, because they are given by the solution of the problem. The authors use a switched system that is embedded into a larger family of systems and the optimal control problem is formulated for this family. When the necessary conditions indicate a bang-bang-type of solution, they obtain a solution to the original problem. However, in the cases when a bang-bang type solution does not exist, the solution to the embedded optimal control problem can be approximated by the trajectory of the switched system generated by an appropriate switching control. On the other hand, in [36] and [34] the authors determine the appropriated control law by finding the singular trajectory along some time with non null measure.

2.3.2 Switched Nonlinear Systems

The nonlinear, non-convex form of the control variable, prevents us from using the Hamilton equations of the maximum principle and nonlinear mathematical programming techniques on them. Both approaches would entail severe difficulties, either in the integration of the Hamilton equations or in the search method of any numerical optimization algorithm. Consequently, we propose to convexify the control variable by using the method of moments in the polynomial expression in order to deal with this kind of problems. In this paper we present a method for solving optimal control for an autonomous switched systems problem based on the method of moments developed in for optimal control, and in [28], [29], [30] and [32] for global optimization. We propose an alternative approach for computing effectively the solution of nonlinear, optimal control problems. This method works properly when the control variable (i.e., the switching signal) can be expressed as polynomials. The essential of this paper is the transformation of a nonlinear, non-convex optimal control problem (i.e., the switched system) into an equivalent optimal control problem with linear and convex structure, which allows us to obtain an equivalent convex formulation more appropriate to be solved by high performance numerical computing. To this end, first of all, it is necessary to transform the original switched system into a continuous non-switched system for which the theory of moments is able to work. Namely, we relate with a given controllable switched system, a controllable continuous non-switched polynomial system.

Optimal control problems for switched nonlinear systems are investigated. We propose an alternative approach for solving the optimal control problem for a nonlinear switched system based on the theory of moments. The essence of this method is the transformation of a nonlinear, non-convex optimal control problem, that is, the switched system, into an equivalent optimal control problem with linear and convex structure, which allows us to obtain an equivalent convex formulation more appropriate to be solved by high-performance numerical computing. Consequently, we propose to convexify the control variables by means of the method of moments obtaining semidefinite programs. The paper dealing with this approach is given in the Appendix 2, paper [J.5].

Chapter 3

Supervisory Control of Discrete-Event Systems

3.1 Multi-Agent Based Supervisory Control

Supervisory control initiated by Ramadge and Wonham [56] provides a systematic approach for the control of discrete event system (DES) plant. The discrete event system plant be is modeled by a finite state automaton [50],[43]:

Definition 1 *(Finite-state automaton). A finite-state automaton is defined as a 5-tuple*

$$G = (Q, \Sigma, \delta, q_0, Q_m, \mathbb{C})$$

where

- Q *is the finite set of states,*
- Σ *is the finite set of events,*
- $\delta : Q \times \Sigma \to Q$ *is the partial transition function,*
- $q_0 \subseteq Q$ *is the initial state,*
- $Q_m \subseteq Q$ *is the set of* marked *states (final states),*

Let Σ^* be the set of all finite strings of elements in Σ including the empty string ε. The transition function δ can be generalized to $\delta : \Sigma^* \times Q \to Q$ in the following recursive manner:

$$\delta(\varepsilon, q) = q$$

$$\delta(\omega\sigma, q) = \delta(\sigma, \delta(\omega, q)) \text{ for } \omega \in \Sigma^*$$

The notation $\delta(\sigma, q)!$ for any $\sigma \in \Sigma^*$ and $q \in Q$ denotes that $\delta(\sigma, q)$ is defined. Let $L(G) \subseteq \Sigma^*$ be the language generated by G, that is,

$$L(G) = \{\sigma \in \Sigma^* | \delta(\sigma, q_0)!\}$$

Let $K \subseteq \Sigma^*$ be a language. The set of all prefixes of strings in K is denoted by $pr(K)$ with $pr(K) = \{\sigma \in \Sigma^* | \exists\ t \in \Sigma^*; \sigma t \in K\}$. A language K is said to be *prefix closed* if $K = pr(K)$. The event set Σ is decomposed into two subsets Σ_c and Σ_{uc} of controllable and uncontrollable events, respectively, where $\Sigma_c \cap \Sigma_{uc} = \emptyset$. A controller, called a supervisor, controls the plant by dynamically disabling some of the controllable events.

A sequence $\sigma_1\sigma_2 \dots \sigma_n \in \Sigma^*$ is called a *trace* or a *word* in term of language. We call a *valid trace* a path from the initial state to a marked state ($\delta(\omega, q_0) = q_m$ where $\omega \in \Sigma^*$ and $q_m \in Q_m$).

In this section we will focus on the Multi-Agent Based Supervisory Control, introduced by Hubbard and Caines [64]; and the modified approach proposed by Takai and Ushio [65]. The two approaches have been applied to the supervisory control of the EMN Experimental Manufacturing Cell. This cell is composed of two robotized workstations connected to a central conveyor belt. Then, three new semi- automated workstations have been added in order to increase the flexibility aspects of the cell. Indeed, each semi-automated workstation can perform either manual of robotized tasks. These two aspects correspond to the two different approaches of multi-agent product of subsystems, for supervisory control purpose. The results can be found in [C.25].

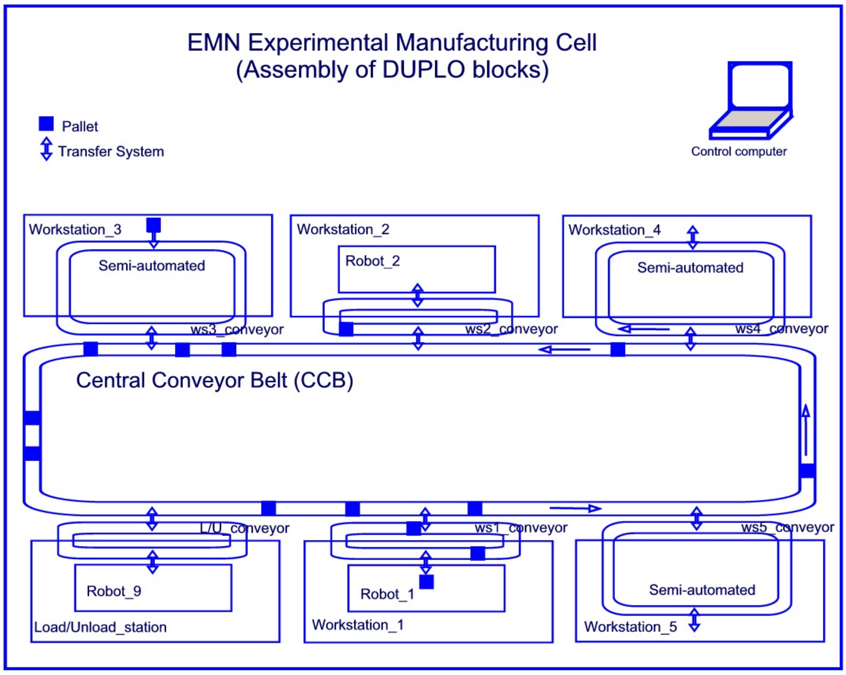


Figure 3.1: EMN Cell

3.2 Switched Discrete-Event Systems

The notion of **switched** discrete-event systems corresponds to a class of DES where each automaton is the composition of two basic automata, but with different composition operators. A switching occurs when there is a change of the composition operator, but keeping the same two basic automata. A mode behavior, or mode for short, is defined to be by the DES behavior for a given composition operator. Composition operators are supposed to change more than once so that each mode is visited more than once. This new class of DES includes the DES in the context of fault diagnosis where different modes such as e.g., normal, degenerated, emergency modes can be found. The studied situations are the ones where the DES switch between different normal modes, and not necessary the degenerated and the emergency ones.

The most common composition operators used in supervisory control theory are the product and the parallel composition [43], [63] However many different types of composition operators have been defined, e.g., the prioritized synchronous composition [49], the biased synchronous composition [52], see [61] for a review of most of the composition operators. Multi-Agent composition operator [57], [58] is another kind of operator, which differs from the synchronous product in the aspects of simultaneity and synchronization.

The new class of DES that we define in this paper includes the class of DES in the context of fault diagnosis, with different operating modes. Furthermore this new class addresses especially the DES for which the system can switch from a given normal mode, to another normal mode. More

precisely this new class of DES is an automaton which is the composition of two basic automata, but with different composition operators. A switching corresponds to the change of composition operator, but the two basic automata remains the same. A mode behavior (or mode for short) is defined to be the DES situation for a given composition operator. Composition operators are supposed to change more than once so that each mode is visited more than once.
We give here below some examples of switched DES:

- Manufacturing systems where the operating modes are changing (e.g. from normal mode to degenerated mode)
- Discrete event systems after an emergency signal (from normal to safety mode)
- Complex systems changing from normal mode to recovery mode (or from safety mode to normal mode).

We can distinguish, like for the switched continuous-time systems, the notion of *autonomous* switching where no external action is performed and the notion of *controlled* switching, where the switching is *forced*. The results for this section can be found in [55].

3.3 Switchable Languages of DES

The notion of switchable languages has been defined by Kumar, Takai, Fabian and Ushio in [Kumar-et-al. 2005]. It deals with switching supervisory control, where switching means switching between two specifi- cations. In this paper, we first extend the notion of switchable languages to n languages, $(n \geq 3)$. Then we consider a discrete-event system modeled with weighted automata. The switching supervisory control strategy is based on the cost associated to each event, and it allows us to synthesize an optimal supervisory controller. Finally the proposed methodology is applied to a simple example.
We now give the main results of this paper. First, we define a triplet of switchable languages. Second we derive a necessary and sufficient condition for the transitivity of switchable languages $(n = 3)$. Third we generalize this definition to a n-uplet of switchable languages, with $n > 3$. And fourth we derive a necessary and sufficient condition for the transitivity of switchable languages for $n > 3$.

3.3.1 Triplet of Switchable Languages

We extend the notion of pair of switchable languages, defined in [51], to a triplet of switchable languages.

Definition 2 *(Triplet of switchable languages). A triplet of languages* (K_1, K_2, K_3)*,* $K_i \subseteq L_m(G)$ *with* $H_i \subseteq K_i,\ i = \{1, 2, 3\}$ *are said to be a* triplet of switchable languages *if they are pairwise switchable languages, that is,*

$SW(K_1, K_2, K_3) := SW(K_i, K_j),\ i \neq j,\ i, j = \{1, 2, 3\}.$

Another expression of the triplet of switchable languages is given by the following lemma.

Lemma 1 *(Triplet of switchable languages). A triplet of languages* (K_1, K_2, K_3)*,* $K_i \subseteq L_m(G)$ *with* $H_i \subseteq K_i,\ i = \{1, 2, 3\}$ *are said to be a* triplet of switchable languages *if the following holds:*

$SW(K_1, K_2, K_3) = \{(H_1, H_2, H_3) \mid H_i \subseteq K_i \cap pr(H_j), i \neq j, \text{ and } H_i \text{ controllable}\}.$

3.3.2 Transitivity of Switchable Languages ($n = 3$)

The following theorem gives a necessary and sufficient condition for the transitivity of switchable languages.

Theorem 1 *(Transitivity of switchable languages, n = 3) . Given 3 specifications* (K_1, K_2, K_3)*,* $K_i \subseteq L_m(G)$ *with* $H_i \subseteq K_i,\ i = \{1, 2, 3\}$ *such that* $SW(K_1, K_2)$ *and* $SW(K_2, K_3)$*.*
(K_1, K_3) *is a pair of switchable languages, i.e.* $SW(K_1, K_3)$*, if and only if*

1. $H_1 \cap pr(H_3) = H_1$*, and*
2. $H_3 \cap pr(H_1) = H_3$*.*

The proof can be found in [42].

3.3.3 N-uplet of Switchable Languages

We now extend the notion of switchable languages, to a n-uplet of switchable languages, with ($n > 3$).

Definition 3 *(N-uplet of switchable languages, n > 3). A n-uplet of languages* $(K_1, ..., K_n)$*,* $K_i \subseteq L_m(G)$ *with* $H_i \subseteq K_i,\ i = \{1, ..., n\}, n > 2$*, is said to be a* n-uplet of switchable languages *if the languages are pairwise switchable that is,*

$SW(K_1, ..., K_n) := SW(K_i, K_j),\ i \neq j,\ i, j = \{1, ..., n\}, n > 2.$

As for the triplet of switchable languages, an alternative expression of the n-uplet of switchable languages is given by the following lemma.

Lemma 2 *(N-uplet of switchable languages, n > 3). A n-uplet of languages* $(K_1, \ldots, K_n)$*,* $K_i \subseteq L_m(G)$ *with* $H_i \subseteq K_i,\ i = \{1, ..., n\},\ n > 3$ *are said to be a* n-uplet of switchable languages *if the following holds:*

$SW(K_1, ..., K_n) = \{(H_1, ..., H_n) \mid H_i \subseteq K_i \cap pr(H_j), i \neq j, \text{ and } H_i \text{ controllable}\}.$

3.3.4 Transitivity of Switchable Languages ($n > 3$)

We are now able to derive the following theorem that gives a necessary and sufficient condition for the transitivity of n switchable languages.

Theorem 2 *(Transitivity of n switchable languages, n > 3) . Given n specifications* $(K_1, ..., K_n)$*,* $K_i \subseteq L_m(G)$ *with* $H_i \subseteq K_i,\ i = \{1, ..., n\}$*. Moreover, assume that each language* K_i *is at least switchable with another language* $K_j,\ i \neq j$*.*
A pair of languages (K_k, K_l) *is switchable i.e.* $SW(K_k, K_l)$*, if and only if*

1. $H_k \cap pr(H_l) = H_k$*, and*
2. $H_l \cap pr(H_k) = H_l$*.*

The proof is similar to the proof of Theorem 6 and can be found in [42]. It is to be noted that the assumption that each of the n languages be at least switchable with another language is important, in order to derive the above result. The results can be found in [C.sub1].

Chapter 4

Conclusion and Future Work

4.1 Summary of Contributions

In this HDR Thesis, I have presented a summary of contribution, in Analysis and Control of Hybrid Systems, as well as in Supervisory Control of Discrete-event Systems.

- Analysis and Control of Hybrid and Switched Systems
  - Modeling and Control of MLD Systems
  - Stability of Switched Systems
  - Optimal Control of Switched Systems
- Supervisory Control of Discrete-Event Systems
  - Multi-Agent Based Supervisory Control
  - Switched Discrete-Event Systems
  - Switchable Languages of DES

I have chosen to not present some work like the **Distributed Resource Allocation Problem**, the **Holonic Systems**, and the **VMI-Inventory Control** work. However the references of the corresponding papers are given in the complete list of publications. My perspectives of research in the coming years are threefold: 1) **Control of Smart Grids**, 2) **Simulation with Stochastic Petri Nets** and 3) **Planning and Inventory Control**.

4.2 Perspective 1: Control of Smart Grids

According to the US Department of Energy's Electricity Advisory Committee, "A **Smart Grid** brings the power of networked, interactive technologies into an electricity system, giving utilities and consumers unprecedented control over energy use, improving power grid operations and ultimately reducing costs to consumers."

The transformation from traditional electric network, with centralized energy production to complex and interconnected network will lead to a smart grid. The five main triggers of Smart grid, according to a major industrial point of view, are 1) Smart energy generation, 2) Flexible distribution, 3) Active energy efficiency, 4) Electric vehicles, and 5) Demand response.
From a control point of view, a smart grid is a system of interconnected micro-grids. A **micro-grid** is a power distribution network where generators and users interact. Generators technologies include renewable energy such as wind turbines or photovoltaic cells.
The objective of this project is to **simulate and control a simplified model of a micro-grid** that is a part of a Smart Grid. After a literature review, a simplified model for control will be chosen. Different realistic scenarios will be tested in simulation with MATLAB. Finally different

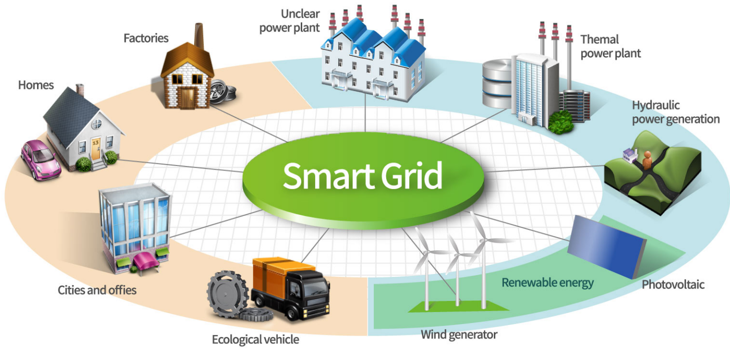


Figure 4.1: Smart Grid

control strategies e.g. LQ/LQR Control, MPC and Hybrid Control will be tested in simulation with MATLAB.

4.3 Perspective 2: Simulation with Stochastic Petri Nets

The Air France CDG Airport Hub in Paris-Roissy is dealing daily with 40,000 transfer luggages and 30,000 local luggages (leaving from or arriving at CDG Airport). For this purpose Air France is exploiting the Sorting Infrastructure of Paris Aeroport, and has to propose a Logistical Scheme Allocation for each luggage in order to optimize the sorting and to minimize the number of *failed* luggages. By failed luggages, we mean a luggage that does not arrive in time for the assigned flight. The KPI Objective for 2017 is to have less than 20 failed luggages out of 1000 passengers.

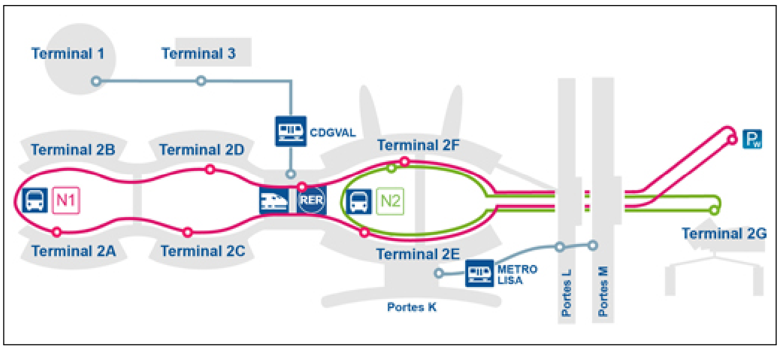


Figure 4.2: CDG Airport Paris-Roissy

4.4 Perspective 3: Planning/Inventory Control

The strategy of integration known as VMI (Vendor-Managed Inventory) allows the coordination of inventory policies between producers and buyers in supply chains. Based on a new proposed model for the implementation of VMI in a chain of two links composed of a producer and a buyer, this paper studies the evolution of individual strategies of the producer and the buyer by a formalism derived from the theory of evolutionary games. The conditions that determine the stability of evolutionarily stable strategies are derived and analyzed. Work results specify analytical conditions that favor the implementation of VMI on traditional chains without VMI.

References

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Sci* vol. 138, pp. 3-34, 1995.

[2] P.J. Antsaklis. "A brief introduction to the theory and applications of hybrid systems," *Proc. of the IEEE*, vol.88, no. 7, pp. 879-887, July 2000.

[3] A. Bemporad and M. Morari. "Control of systems integrating logic, dynamics, and constraints,". *Automatica*, vol. 35 no. 3, pp. 407-427, 1999.

[4] A. Bemporad, G. Ferrari-Trecate, and M. Morari. "Observability and controllability of PWA and hybrid systems," *IEEE Trans. on Automatic Control*, vol. 45, no. 10, pp. 1864–1876, Oct. 2000.

[5] A. Bemporad. "An efficient technique for translating mixed logical dynamical systems into piecewise affine systems," in *Proc. of 42nd IEEE Conf. Decision Control*, 2002, pp. 1970-1975.

[6] A. Bemporad. "Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form," *IEEE Trans. on Automatic Control*, vol. 49, no. 5, pp. 832–838, May 2004.

[7] V.D. Blondel and J.N. Tsitsiklis. "Complexity of stability and controllability of elementary hybrid systems,". *Automatica*, vol. 35 no. 3, pp. 479-490, 1999.

[8] T. Geyer, F.D. Torrisi, and M. Morari. "Efficient mode enumeration of compositional hybrid systems," in *Proc. of HSCC 2003*, O. Maler and A. Pnueli (Eds.), LNCS 2623, pp. 216-232, 2003.

[9] W.P.M.H. Heemels, B. de Schutter, and A. Bemporad. "Equivalence of hybrid dynamical models," In Proc. of *IEEE Symp. on Logic in Computer Science*, New Brunswick, NJ, USA, pp. 278–292, 1996.

[10] T.A. Henzinger. "The theory of hybrid automata," Proc. of *the 11th Annual Symp. on LICS*, pp. 278-292, IEEE Computer Society Press, 1996.

[11] H. Lin and P.J. Antsaklis. "Stability and stabilizability of switched linear systems: A short survey of recent results," in *Proc. Int. Symp. on Intelligent Control*, Limassol, Cyprus, June 2005, pp. 24-29.

[12] M. Margaliot. " Stability analysis of switched systems using variational principles,". *Automatica*, vol. 42, pp. 2059-2077, 2006.

[13] A. Rantzer and M. Johansson. "Piecewise linear quadratic optimal control of hybrid systems," *IEEE Trans. on Automatic Control*, vol. 45, no. 4, pp. 629–637, April 2000.

[14] C. Seatzu, D. Corona, A. Giua, and A. Bemporad. "Optimal control of continuous-time switched affine systems," *IEEE Trans. on Automatic Control*, vol. 49, no. 5, pp. 726–741, May 2006.

[15] E.D. Sontag. "Nonlinear regulation: The piecewise affine approach," *IEEE Trans. on Automatic Control*, vol. 51, no. 5, pp. 346-358, April 1981.

[16] Z. Sun and S.S. Ge. "Analysis and synthesis of switched linear control systems,". *Automatica*, vol. 41, pp. 181-195, 2005.

[17] F.D. Torrisi and A. Bemporad. "HYSDEL– A tool for generating computational hybrid models,". *IEEE Trans. Control System Technology*, Vol. 12, N.2, pp. 235-249, Mar 2004.

[18] D. Liberzon, "Switching in Systems and Control," Birkhauser, Boston, 2003.

[19] H. Lin, P.J. Antsaklis. "Stability and stabilizability of switched linear systems: A survey of recent results" *IEEE Transactions on Automatic Control* , vol.54, no. 2, pp. 308–322, 2009.

[20] W. Dayawansa, C. Martin, " A converse Lyapunov theorem for a class of dynamical systems which undergo switching" *IEEE Transactions on Automatic Control* , vol. 44 N. pp.751–760, 1999.

[21] J. Mancilla-Aguilar, R. Garcia, "A converse Lyapunov theorem for nonlinear switched systems," *Systems and Control Letters*, vol. 41 N. 1 pp.67–71, 2000.

[22] L. Vu, D. Liberzon, "Common Lyapunov functions for families of commuting nonlinear systems" *Systems and Control Letters*, vol. 54 N. 1 pp. 405–416, 2005.

[23] C. Ebenbauer, F. Allgwer, "Stability analysis of constrained control systems: An alternative approach," *Systems and Control Letters*, vol. 56 pp. 93–98, 2007.

[24] A. Papachristodoulou, S. Prajna, "Analysis of non-polynomial systems using the sum of squares decomposition," *Positive Polynomials in Control* Springer, pp. 23–43, 2005.

[25] P. Kunkel, V. Mehrmann, "Stability properties of differential-algebraic equations and spin-stabilized discretizations," *Electronic Transactions on Numerical Analysis* vol. 26, 385–420, 2007.

[26] S. Prajna, A. Papachristodoulou, "Analysis of switched and hybrid systems?Beyond piecewise quadratic methods *Proc. of the American Control Conference*, Boston, MA, USA, pp. 1208-1213, July 2004.

[27] S. C. Bengea and R. A. DeCarlo "Optimal control of switching systems". *Automatica*, vol. 41, pp. 11–27, 2005.

[28] J.B. Lasserre, "Global optimization with polynomials and the problem of moments.". *SIAM J. Optimization*, vol. 11, pp. 796-817, 2001.

[29] J.B. Lasserre, C. Prieur, D. Henrion, "Nonlinear optimal control: numerical approximation via moments and LMI relaxations.". *Proc. of the Joint IEEE CDC and ECC 2005, Sevilla, Spain*, Dec. 2005.

[30] J.B. Lasserre, D. Henrion, C. Prieur, E. Trelat, "Nonlinear optimal control via occupation measures and LMI relaxations.". *SIAM J. Control and Optimization*, vol. 47 N. 4, pp. 1643-1666, 2008.

[31] D. Liberzon, "Switching in Systems and Control," Birkhauser, Boston, 2003.

[32] R. Meziat, D. Patino, and P. Pedregal, "An alternative approach for non-linear optimal control problems based on the method of moments, ". *Computational Optimization and Applications*, vol. 38, pp. 147–171, 2007

[33] P. Parrilo, "Structured semidefinite programs and semi-algebraic geometry methods in robustness and optimization. ". Ph.D. Thesis, California Institute of Technology, 2000.

[34] D. Patino, P. Riedinger, and C. Iung, " Optimal state feedback control law for continuous-time affine switched systems.," 2007.

[35] P. C. Perera and W. P. Dayawansa, "Asymptotic feedback controllability of switched control systems to the origin," *Proc. of the American Control Conference*, pp. 5806–5811, 2004.

[36] P. Riedinger, J. Daafouz, and C. Iung, "Suboptimal switched controls in context of singular arcs," *Proc. of 42nd IEEE Conference on Decision and Control*, 6254–6259, 2003.

[37] W. Spinelli, P. Bolzern, and P. Colaneri, "A note on optimal control of autonomous switched systems on a finite time interval," *Proc. of the American Control Conference*, 2006.

[38] K. C. Toh, M. J. Todd, and R. Tutuncu, "SDPT3 ? A Matlab software package for semidefinite programming, " *Optimization Methods and Software*, vol.11, pp. 545–581, 1999.

[39] X. Xu, P.J. Antsaklis. "Optimal control of switched systems based on parameterization of the switching instants, " *IEEE Transactions on Automatic Control* , vol.49, pp. 2–16, 2004.

[40] S.E. Bourdon, M. Lawford, and W.M. Wonham "Robust nonblocking supervisory control of discrete event systems," In *IEEE Trans. on Automatic Control*, vol. 50, N.12 pp. 2015–2021, 2005.

[41] M. Canu and N. Rakoto-Ravalontsalama, *From mutually non-blocking to switched non-blocking DES*. Presented at MSR'13 Workshop (Poster Session), Rennes, France, Nov 13-15, 2013.

[42] M. Canu and N. Rakoto-Ravalontsalama. *On Switchable Languages of Discrete-Event Systems with Weighted Automata*, Technical Report Mines Nantes, March 2017.

[43] C.G. Cassandras and S. Lafortune, "Introduction to Discrete Event Systems," 2nd Edition, Springer Verlag, 2008.

[44] M. Fabian and R. Kumar. "Mutually non-blocking supervisory control of discrete-event systems," In *Automatica*, 36(12) pp. 1863–1869, 2000.

[45] G. Faraut, L. Pietrac, and E. Niel, "Formal Approach to Multimodal Control Design: Application to Mode Switching", In *IEEE Trans. on Industrial Informatics*, vol.5, N.4 pp. 443–453, Nov 2009.

[46] G. Faraut, L. Pietrac, and E. Niel, "Process Tracking by Equivalent States in Modal Supervisory Control", Proc. of *IEEE ETFA*, Sep. 2011, Toulouse, France.

[47] J. Girault, J.J. Loiseau, O.H. Roux "Synthese en ligne de superviseur compositionnel pour une flotte de robots mobiles." European Journal of Automation, MSR'13, vol 47/1-3, pp. 195–210, 2013.

[48] J. Girault, J.J. Loiseau, O.H. Roux "On-line optimal compositional controller synthesis for AGV by unfolding," Proc. of DCDS 2015, IFAC-PapersOnLine 48-7 (2015) pp. 167–173.

[49] M. Heymann, "Concurrency and discrete event control," In *IEEE Control Systems Magazine*, vol 10, N.4 pp. 103–112, 1990.

[50] J.E. Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley, Reading, MA, USA, 1979.

[51] R. Kumar, S. Takai, M. Fabian, and T. Ushio, "Maximally Permissive Mutually and Globally Nonblocking Supervision with Application to Switching Control," In *Automatica*, 41(8) pp. 1299–1312, 2005.

[52] S. Lafortune, E. Chen, "The infimal closed controllable superlanguage and its application to supervisory control, " *IEEE Transactions on Automatic Control* , vol.35, N.4, pp. 398–405, 1990.

[53] D. Liberzon, "Switching in Systems and Control", ser. Systems and Control: Foundations and Applications. Boston: Birkhauser, 2003.

[54] R. Malik and R. Leduc, "Generalised nonblocking"', in Proc. 9th Int. Workshop on Discrete Event Systems, WODES 2008, Goteborg, Sweden, May 2008, pp. 340–345.

[55] N. Rakoto-Ravalontsalama. "Supervisory control of switched discrete-event systems," in Proc. of 17th Symp. on MTNS 2006, Kyoto, Japan 2006, pp. 2213–2217.

[56] P.J. Ramadge and W.M. Wonham. "Supervisory control of a class of discrete-event processes," In *SIAM J. Control and Optimization*, vol.25 pp. 206–230, 1987.

[57] I. Romanovski , P.E. Caines, "On the supervisory control of multi-agent product systems," *Proc. of IEEE Conference on Decision and Control*, Las Vegas, Nevada, pp. 1181-1186, 2002.

[58] I. Romanovski , P.E. Caines, "Multi-agent product system: Controllability and non-blocking properties," *Proc. of WODES 2006*, Ann Arbor, MI, 2006.

[59] P.S. Roop, A. Girault, R. Sinha, and G. Goessler. "Specification Enforcing Refinement for Convertibility Verification," Proc. of ACSD 2009, pp. 148–157, IEEE, 2009.

[60] T.J.J. van den Boom and B. de Schutter. "Modelling and control of discrete event systems using switching max-plus-linear systems," *Control Engineering Practice*, vol. 14 N.10, pp. 1199-1211, 2006.

[61] F. Wenck, J.H. Richter, "On composition oriented perspective on controllability of large DES," *Proc. of WODES 2004*, Reims, France, pp. 271–276, 2004.

[62] W.M. Wonham and P.J. Ramadge. "Modular supervisory control of discrete-event systems," In *Mathematics of Control, Signals and Systems*, vol.1 (1) pp. 13–30, 1988.

[63] W.M. Wonham, "Notes on Discrete Event Systems," Dept. of ECE, University of Toronto, Toronto, Canada, 2004.

[64] P. Hubbard and P.E. Caines, "Initial investigations on hierachical supervisory control of multi-agent systems," *Proc. of IEEE Conference on Decision and Control*, Phoenix, AZ, pp. 2218–2223, 1999.

[65] S. Takai and T. Ushio, "Supervisory control of a class of concurrent discrete-event systems," *Proc. of ATPN - Workshop on DES Control*, Eindhoven, The Netherlands,, pp. 37?55, June 2003

Appendix A

Appendix 1 – Paper [C.24]:

- [C.24] J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama,
  A new algorithm for translating MLD systems into PWA systems.
  Proc. of IEEE American Control Conference (ACC 2004), June 30 - July 2, 2004, Boston, MA, USA, pp. 1208 –1213.

A New Algorithm for Translating MLD Systems into PWA Systems

J.L. Villa[1,2], M. Duque[2], A. Gauthier[2] and N. Rakoto-Ravalontsalama[1]
[1]Ecole des Mines de Nantes / IRCCyN
4, rue Alfred Kastler, F-44307 Nantes Cedex 03, France
{jvilla, rakoto}@emn.fr
[2]Universidad de los Andes
Kra 65B No 17A-11, Bogotá D.C., Colombia
{maduque, agauthie}@uniandes.edu.co

Abstract – This paper presents a new algorithm for translating Mixed Logical and Dynamical (MLD) systems into PieceWise Affine (PWA) systems. The presented algorithm uses an enumeration technique and solves several linear programming problems in order to obtain the equivalence. The obtained model is equivalent to the MLD model meaning that given an initial state and an input sequence, the trajectory of the state vector and output vector are the same. The technique is applied to three examples. The computation time and the simulation results for these examples are given.

I. INTRODUCTION

Mixed and Logical Dynamical (MLD) models introduced by Bemporad and Morari in [2] arise as a suitable representation for Hybrid Dynamical Systems (HDS), in particular for solving control-oriented problems. MLD models can be used for solving a model predictive control (MPC) problem of a particular class of HDS and it is proved that MLD models are equivalent to PieceWise Affine Models in [6]. In the paper by Heemels and co-workers, the equivalencies among PieceWise Affine (PWA) Systems, Mixed Logical and Dynamical (MLD) systems, Linear Complementarity (LC) systems, Extended Linear Complementarity (ELC) systems and Max-Min-Plus-Scaling (MMPS) systems are proved, these relations are transcribed here in Fig. 1.

This equivalences are based on some propositions (see [6] for details)

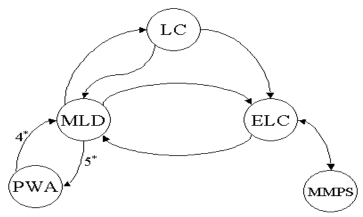


Fig. 1. Equivalence relation between hybrid systems

Every well-posed PWA system can be re-written as an MLD system assuming that the feasible states and inputs are bounded [6, proposition 4*].

A completely well-posed MLD system can be rewritten as a PWA system [6, proposition 5*].

A more formal proof can be found in [3], where an efficient technique for obtaining a PWA representation of a MLD model is proposed.

The technique in [3] describes a methodology for obtaining, in an efficient form, a partition of the state-input space. The algorithm in [3] uses some tools from polytopes theory in order to avoid the enumeration of the all possible combinations of the integer variables contained in the MLD model. However, the technique does not describe the form to obtain a suitable choice of the PWA model, even though this part is introduced in the implementation provided by the author in [4]. The objective of this paper is to propose an algorithm of the suitable choice of the PWA description and use the PWA description for obtaining some analysis and control of Hybrid Dynamical Systems.

II. MLD SYSTEMS AND PWA SYSTEMS

A. Mixed and Logical Dynamical (MLD) Systems

The idea in the MLD framework is to represent logical propositions as equivalent integer expressions. MLD form is obtained by three basic steps [5]. The first step is to associate a binary variable $\delta \in \{0,1\}$ with a proposition S, that may be true or false. δ is 1 if and only if proposition S is true. A composed proposition of elementary propositions $S_1,\ldots,S_q$ combined using the boolean operators like AND(^), OR ($\vee$), NOT(~) may be expressed like integer inequalities over corresponding binary variables δ_i, $i=1,\ldots,q$.

The second step is to replace the products of linear functions and logic variables by a new auxiliary variable $z = \delta a^T x$ where a^T is a constant vector. The z value is obtained by mixed linear inequalities evaluation.

The third step is to describe the dynamical system, binary variables and auxiliary variables in a linear time invariant (LTI) system.

A hybrid system MLD described in general form is represented by (1).

$$\begin{cases} x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\ y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \\ E_2\delta(k) + E_3z(k) \le E_1u(k) + E_4x(k) + E_5 \end{cases} \quad (1)$$

where $x = [x_C^T x_l^T] \in \mathbf{R}^{n_c} \times \{0,1\}^{n_l}$ are the continuous and binary states, $u = [u_C^T u_l^T] \in \mathbf{R}^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $y = [y_C^T y_l^T] \in \mathbf{R}^{p_c} \times \{0,1\}^{p_l}$ the outputs, and $\delta \in \{0,1\}^{r_l}$, $z \in \mathbf{R}^{r_c}$, represent the binary and continuous auxiliary variables, respectively. The constraints over state, input, output, z and δ variables are included in the third term in (1).

B. PieceWise Affine Systems

A particular class of hybrid dynamical systems is the system described as follows,

$$\begin{cases} \dot{x} = A_i x + a_i + B_i u \\ y = C_i x + c_i + D_i u \end{cases}, \quad x \times u \in X_i,\ i \in I,\ t \in \mathbb{R}^+ \qquad (2)$$

where $\mathcal{I}$ is a set of indexes, X_i is a sub-space of the real space $\mathbb{R}^n$, and $\mathbb{R}^+$ is the set of positive real numbers including the zero element.

In addition to this equation it is necessary to define the form as the system switches among its several modes. This equation is affine in the state space x and the systems described in this form are called PieceWise Affine Systems (PWA). In the literature of hybrid dynamical systems the systems described by the autonomous version of this representation are called Switched Systems.

If the system vanishes when x brings near to zero, i.e. a_i and b_i are zero, then the representation is called PieceWise Linear (PWL) system.

The discrete-time version of this equation will be used in this work and can be described as follows,

$$\begin{cases} x(k+1) = A_i x(k) + b_i + B_i u(k) \\ y(k) = C_i x(k) + d_i + D_i u(k) \end{cases}, \quad x \times u \in X_i,\ i \in \mathcal{I},\ k \in \mathbf{Z}^+ \qquad (3)$$

where $\mathcal{I}$ is a set of indexes, X_i is a sub-space of the real space $\mathbb{R}^n$.

III. MLD SYSTEMS INTO PWA SYSTEMS

The MLD framework is a powerful structure for representing hybrid systems in an integrated form. Although E_1, E_2, E_3, E_4 and E_5 matrices are, in general, large matrices, they can be obtained automatically. An example is the HYSDEL compiler [10].

However, some analysis of the system with the MLD representation are computationally more expensive with respect to some tools developed for PWA representations. Exploiting the MLD and PWA equivalencies, it is possible to obtain analysis and control of a system using this equivalent representations. Nevertheless, as it is underlined in [3], this procedure is more complex with respect to the PWA into MLD conversion, and there exist more assumptions. To our knowledge, the only previous approach has been proposed by Bemporad [3]. We propose then a new approach of translating MLD into PWA systems.

The MLD structure can be rewritten as follows,

$$\begin{cases} x(k+1) = Ax(k) + \begin{bmatrix} B_{1c} & B_{1l} \end{bmatrix} \begin{bmatrix} u_c(k) \\ u_l(k) \end{bmatrix} + B_2\delta(k) + B_3 z(k) \\ y(k) = Cx(k) + \begin{bmatrix} D_{1c} & D_{1l} \end{bmatrix} \begin{bmatrix} u_c(k) \\ u_l(k) \end{bmatrix} + D_2\delta(k) + D_3 z(k) \\ E_2\delta(k) + E_3 z(k) \le E_2 x(k) + \begin{bmatrix} E_{1c} & E_{1l} \end{bmatrix} \begin{bmatrix} u_c(k) \\ u_l(k) \end{bmatrix} + E_5 \end{cases} \qquad (4)$$

Here, the binary inputs are distinguished from the continuous inputs, because they induce switching modes in the system, in general.

Supposing that the system is well posed, $z(k)$ has only one possible value for a given $x(k)$ and $u(k)$, and can be rewritten as:

$$z(k) = k_1 x(k) + k_2 u_c(k) + k_3 \ |m[x^T, u^T]^T \le b \qquad (5)$$

Replacing this value in the original equations the system can be represented as,

$$\begin{cases} x(k+1) = (A + B_3 k_1)x(k) + (B_{1c} + B_3 k_2)u(k) + B_3 k_3 \\ y(k+1) = (C + D_3 k_1)x(k) + (D_{1c} + D_3 k_2)u(k) + D_3 k_3 \\ (-E_4 + E_3 k_1)x + (-E_1 + E_3 k_2)u \le E_5 - E_2\delta - E_3 k_3 \end{cases} \qquad (6)$$

If an enumeration technique is used for generating all the feasible binary states of the $[u_l^T\ \delta^T]^T$ vector, the first problem is to find a value of $[x^T\ u^T]^T$ feasible for the problem, that can be obtained solving the linear programming problem,

$$\begin{cases} \min & X = [u^T\ z^T\ x^T]^T \\ s.t. & -E_{1c}u_c + E_3 z - E_4 x \le E_5 - E_2\delta + E_{1l}u_l \end{cases} \qquad (7)$$

The solution is a feasible value $[x^{*T}\ u^{*T}]^T$. The next problem is to find k_1, k_2 and k_3.

The inequalities can be rewritten as,

$$E_3 z \le E_4 x + E_{1c}u_c + E_{1l}u_l - E_2\delta + E_5 = \bar{E}_4\bar{k}_1 x + \bar{E}_{1c}\bar{k}_2 u_c + \bar{E}_5\bar{k}_3 \qquad (8)$$

where $\bar{E}_5$ includes every constant in the problem, i.e. u_l and δ. On the other hand, the E_3 matrix reflects the interaction among the z variables, and we can write:

$$F \times z \le \bar{k}_1 x + \bar{k}_2 u + \bar{k}_3 \qquad (9)$$

The matrix F represents the interaction among the z variables, if the system is well posed F^{-1} should exist.

With this last equation, for finding $\bar{k}_3$ the next linear programming problem is solved,

$$\begin{cases} \max & \bar{k}_3 \\ s.t. & E_3\bar{k}_3 \le E_5 - E_2\delta^* + E_{1l}u_l^* \end{cases} \qquad (10)$$

The solution to this problem is $\bar{k}_3$, in this case we assume that all components in $\bar{E}_5$ are the maximum and minimum values of z and the only solution for the problem is $\bar{k}_3$. With $\bar{k}_3$ we can obtain the other matrices.

For obtaining $\bar{k}_1$ it is necessary to solve n_x, i.e. the length of the state vector, linear programming problems,

$$\begin{cases} \max & \overline{k}_i \\ s.t. & E_3\overline{k}_i \le E_5 - E_2\delta^* + E_{1l}u_l^* + E_{4i} \end{cases} \quad (11)$$

where E_{4i} represents the column i of the E_4 matrix and $\overline{k}_{1i} = \overline{k}_i - \overline{k}_3$ is the column i of the matrix $\overline{k}_1$.

For obtaining $\overline{k}_2$ it is necessary to solve n_u, i.e. the length of the continuous input vector, linear programming problems,

$$\begin{cases} \max & \overline{k}_i \\ s.t. & E_3\overline{k}_i \le E_5 - E_2\delta^* + E_{1l}u_l^* + E_{1ci} \end{cases} \quad (12)$$

where E_{1ci} represents the column i of the E_{1c} matrix and $\overline{k}_{2i} = \overline{k}_i - \overline{k}_3$ is the column i of the matrix $\overline{k}_2$.

The matrix F should be found solving n_z, i.e. the length of the z vector, linear programming problems,

$$\begin{cases} \max & \overline{k}_i \\ s.t. & E_3\overline{k}_i \le E_5 - E_2\delta^* + E_{1l}u_l^* + E_{3i} \end{cases} \quad (13)$$

where E_{3i} represents the column i of the E_3 matrix and $F_i = \overline{k}_i - \overline{k}_3$ is the column i of the matrix F.

Finally, k_1, k_2, and k_3, can be computed as,

$$\begin{cases} k_1 = F^{-1}\overline{k}_1 \\ k_2 = F^{-1}\overline{k}_2 \\ k_3 = F^{-1}\overline{k}_3 \end{cases} \quad (14)$$

With these equations, the algorithm for translating the MLD model into PWA model is given as follows,

Algorithm 1

1. Find a feasible point for the binary vector, composed by the binary inputs and binary auxiliary variables.
2. Compute $\overline{k}_3$ using Eq. (10).
3. Compute $\overline{k}_1$, $\overline{k}_2$ and F using Eq. (11), (12) and (13).
4. Compute k_1, k_2, and k_3 using Eq. (14).
5. Using Eq. (6), compute A_i, B_i, f_i, C_i, D_i and g_i and the valid region for this representation.
6. If there exists another feasible point go to step 1.
7. End.

Some gains in the algorithm performance can be obtained if the vector z is evaluated after step one, using a linear program for finding the maximum and the minimum in z, if the z_{min} and z_{max} solutions are the same, it is not necessary to calculate steps 3, and 4, and $z = z_{min} = z_{max}$ can be assigned directly.

IV. EXAMPLES

A. *The Three-Tank Benchmark Problem*

The three-tank benchmark problem has been proposed as an interesting hybrid dynamical system. This Benchmark was proposed in [7] and [8]. See [13] and references there in for some control results using MLD framework in this system. The algorithm described in the last section is used for obtaining a PWA representation of this system.

This system has three tanks each of them interconnected with another as depicted in Fig. 2.

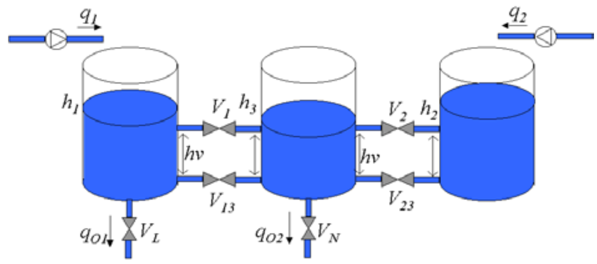


Fig. 2. Three Tank System

The model is written using binary variables (δ_i) and relational expressions,

$$\begin{cases} \delta_1 = 1 \leftrightarrow h_1 > h_v \\ \delta_2 = 1 \leftrightarrow h_2 > h_v \\ \delta_3 = 1 \leftrightarrow h_3 > h_v \end{cases} \qquad \begin{cases} Z_{01} = (h_1 - h_v)\delta_1 \\ Z_{02} = (h_2 - h_v)\delta_2 \\ Z_{03} = (h_3 - h_v)\delta_3 \end{cases}$$

$$\begin{cases} Z_1 = (Z_{01} - Z_{03})V_1 \\ Z_2 = (Z_{02} - Z_{03})V_2 \\ Z_{13} = (h_1 - h_3)V_{13} \\ Z_{23} = (h_2 - h_3)V_{23} \end{cases}$$

$$h_1(k+1) = h_1(k) + T_s * (\frac{q_1(k)}{C_1} - \frac{h_1(k)}{R_L C_1} - \frac{Z_{13}(k)}{R_{13}C_1} - \frac{Z_1(k)}{R_1 C_1})$$

$$h_2(k+1) = h_2(k) - Ts * (\frac{q_2(k)}{C_2} - \frac{Z_{23}(k)}{R_{23}C_2} - \frac{Z_2(k)}{R_2 C_2})$$

$$h_3(k+1) = h_3(k) + Ts * (-\frac{h_3(k)}{R_N C_3} + \frac{Z_{13}(k)}{R_{13}C_3} + \frac{Z_{23}(k)}{R_{23}C_2} + \frac{Z_1(k)}{R_1 C_3} + \frac{Z_2(k)}{R_2 C_3})$$

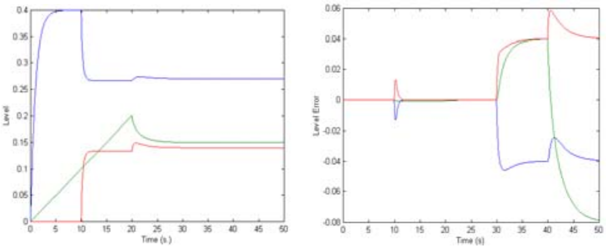
The simulation of the system using the MLD framework and a Mixed Integer Quadratic Programming MIQP algorithm running in an Intel Celeron 2GHz processor and 256MB of RAM was 592.2s, using the PWA representation the same simulation was 1.33s. The time for obtaining the PWA model using the technique described in this work is 72.90s and the algorithm found 128 regions. Using the algorithm in [4] the computation time of the PWA form was 93.88s and the total regions found was 100 and the simulation took 5.89s. These results are summarized in Table I.

Where Computation Time is the time taken by the computer for computing the PWA model based in the MLD model, and Simulation Time is the time taken by the computer for computing a trajectory given a model, an initial state and an input sequence.

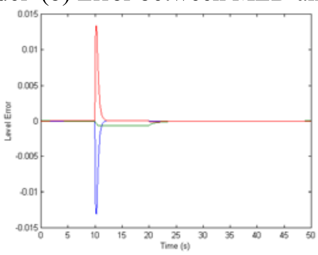
Table I. Computation and Simulation Times

Representation	Computation Time (s.)	Simulation Time (s.)
MLD	-	592.20
PWA-[4]	93.88	5.89
PWA-This work	72.90	1.33

The simulation results with MLD model and the error between PWA simulation results and MLD simulation results, for the same input are shown in Fig. 3,



(a) MLD Model (b) Error between MLD and PWA [4]



(C) Error between MLD and PWA– This Work

Fig. 3. Simulation Results for the Three-Tank System

In this case, at t=30s, the simulation with the PWA system in the Figure 3.b produces a switching to an invalid operation mode.

B. Car with Robotized Manual Gear Shift

The example of a Hybrid Model of a Car with Robotized Manual Gear Shift was reported in [9] and is used in [3] as example. The car dynamics is driven by the following equation,

$$m\ddot{x} = F_e - F_b - \beta\dot{x} \qquad (15)$$

where m is the car mass, $\dot{x}$ and $\ddot{x}$ is the car speed and acceleration, respectively, F_e is the traction force, F_b is the brake force and β is the friction coefficient. The Transmission Kinematics are given by,

$$\omega = \frac{R_g(i)}{k_s}\dot{x}$$

$$F_e = \frac{R_g(i)}{k_s}M$$

where ω is the engine speed, M the engine torque and i is the gear position.

The engine torque M is restricted to belongs between the minimum engine torque $C_e^-(\omega)$ and the maximum engine torque $C_e^+(\omega)$.

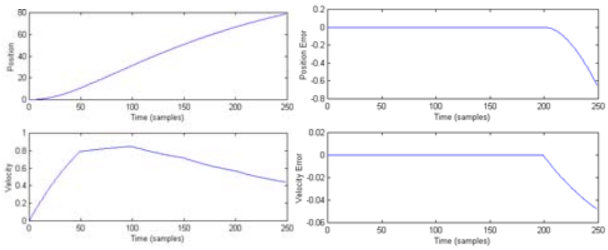
The model has two continuous states, position and velocity of car, two continuous inputs, engine torque and breaking force, and six binary inputs, the gear positions. The MLD model was obtained using the HYSDEL tool.

The translation of the MLD model took 155.73 s and the PWA model found 30 sub-models, using the algorithm proposed in this work, and the PWA model using the algorithm proposed in [3] took 115.52 s and contains 18 sub-models. The simulation time with MLD model and a MIQP algorithm for 250 iterations took 296.25s, using the PWA model obtained with the algorithm proposed here took 0.17s, and using the PWA model obtained using the algorithm in [4] the simulation took 0.35s. These results are summarized in Table II,

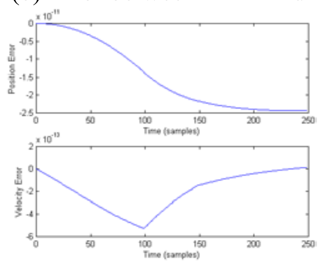
Table II. Computation and Simulation Times

Representation	Computation Time (s.)	Simulation Time (s.)
MLD	-	296.25
PWA-[4]	115.52	0.35
PWA-This work	155.73	0.17

The simulation results with MLD model and the error between PWA simulation results and MLD simulation results, for the same input are shown in Fig. 4,



(a) MLD Model (b) Error between MLD and PWA [4]



(c) Error between MLD and PWA– This Work

Fig. 4. Simulation results for robotized gear shift

C. *The Drinking Water Treatment Plant*

The example of a Drinking Water Treatment Plant has been reported in [11] and [12]. This plant was modeled using identification techniques for hybrid dynamical systems, and its behavior includes autonomous jumps.

The plant modeled is based in the current operation of drinking water plant Francisco Wiesner situated at the periphery of Bogotá D.C. city (Colombia), which treats on average $12m^3/s$. The volume of water produced by this plant is near to 60% of consumption by the Colombian capital. In this plant, there exist two water sources: Chingaza and San Rafael reservoirs which can provide till $22m^3/s$ of water.

The process mixes inlet water with a chemical solution in order to generate aggregated particles that can be caught in a filter. The dynamic of the filter is governed by the differential pressure across the filter and the outlet water turbidity. An automaton associated to the filter executes a back-washing operation when the filter performance is degraded. Because of process non-linearity, the behavior of the system is different with two water sources, that is the case for the particular plant modeled.

The model for each water source includes a dynamic for the aggregation particle process which dynamical variable is called Streaming Current (SC) and is modeled using two state variables, a dynamic for the differential pressure called Head Loss (HL) with only one state variable, a dynamic for the outlet turbidity (T_o) with two state variables.

The identified model consists of four affine models, two for each water source in normal operation, one model in maintenance operation, one model representing the jump produced at the end of the maintenance operation.

$$\begin{cases} x(k+1) = A_i x(k) + B_i u(k) + f_i \\ y(k) = C_i x(k) + D_i u(k) + g_i \end{cases}$$

$$i \in \{1, \text{if water source 1 and normal operation}, \\ 2, \text{if water source 2 and normal operation}, \\ 3, \text{if maintenance operation}, \\ 4, \text{change from maintenance operation to normal operation}\}$$

where *water source* is an input variables, *maintenance operation* is executed if outlet turbidity (T_o) is greater than a predefined threshold, or, Head Loss (HL) is greater than a predefined threshold, or, operation time is greater than a predefined threshold.

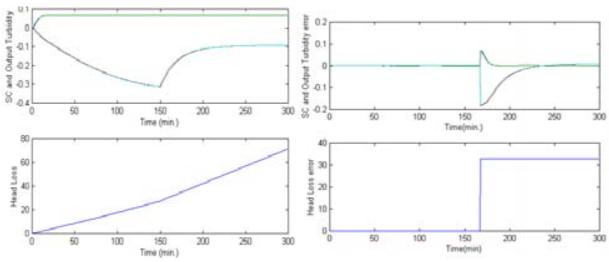
The MLD model has 7 continuous states (including two variables for two timers in the automaton), 4 continuous inputs (dosage, water flow, inlet turbidity and pH), 3 binary inputs (water source, back-washing operation and normal operation), 8 auxiliary binary variables, and 51 auxiliary variables. The complete model can be obtained by mail from the corresponding authors.

The translation from the MLD model into PWA model took 572.19 s, with the algorithm proposed here, generating 127 sub-models. The translation into PWA model took 137.37s, with the algorithm in [3], generating 14 sub-models. The simulation time for 300 iterations with the MLD model and a MIQP algorithm took 4249.301s, the same simulation with the PWA model obtained with the algorithm proposed here took 0.14s, and the same simulation with the PWA model obtained using the algorithm in [4] took 0.31s. These results are summarized in Table III,

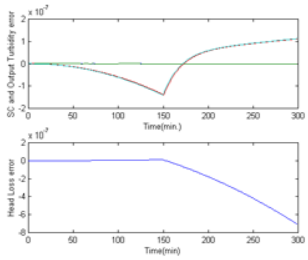
Table III. Computation and Simulation Times.

Representation	Computation Time (s.)	Simulation Time (s.)
MLD	-	4249.30
PWA-[4]	137.37	0.31
PWA-This work	572.20	0.14

The simulation results for the same input are shown in Fig. 5,



(a) MLD Model (b) Error between MLD and PWA [4]



(c) Error between MLD and PWA– This Work

Fig. 5. Simulation results for a water plant model.

In this case, at t=168min, the simulation with the PWA system in the Figure 5.b is not valid because there exist no mode in the PWA representation that belongs to the state-input vector reached in this point. Some other results can be found in [14].

V. CONCLUSIONS

This work presents new algorithm for obtaining a suitable choice of the PWA description from a MLD representation. The results are applied to the three-tank benchmark problem, to a car with robotized gear shift and to a drinking water plant, the three examples have been reported in the literature as examples of hybrid dynamical systems modeled with MLD formalism. The simulation results show that the PWA models obtained have the same behavior with respect to the MLD models. However in some cases the obtained PWA model does not have a valid solution for some state-input sub-spaces.

As a consequence of the enumeration procedure, our PWA models have more submodels/regions than the algorithm in [3], however we show that the procedure does not spent much more computation time because of the simplicity in its formulation, and it ensures the covering of all regions included in the original MLD model.

Ongoing work concerns the analysis of MLD Systems with some results from PWA systems.

VI. ACKNOWLEDMENT

This work has been supported in part by « Contrat Etat - Région No STIC 9-18036, 2000-2006 », Nantes, France. The authors are thankful to the reviewers for their valuable comments and suggestions.

VII. REFERENCES

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P. H. Ho. Hybrid Automata : An Algorithmic Approach To The Specification and Verification of Hybrid Systems. *In Hybrid Systems I*, Lecture Notes in Computer Science 736, Springer-Verlag, pp. 209-229, 1993

[2] A. Bemporad and M. Morari. Control of Systems Integrating Logic, Dynamics, and Constraints. Automatica 35, pp. 407-427, 1999.

[3] A. Bemporad. An Efficient Technique for Translating Mixed Logical Dynamical Systems into PieceWise Affine Systems. Proceedings of the 41st IEEE Conference on Decision and Control. Las Vegas, Nevada USA, December 2002.

[4] A. Bemporad. An Efficient Technique for Translating Mixed Logical Dynamical Systems into PieceWise Affine Systems. Implementation. http://www.dii.unisi.it/~hybrid/tools/mld2pwa

[5] G. Ferrari-Trecate, D. Mignone, D. Castagnoli, and M. Morari. Hybrid Modeling and Control of a Hydroelectric Power Plant. Technical Report AUT00-11, Automatic Control Laboratory, ETH Zurich, Switzerland, 2000.

[6] W.P.M.H. Heemels, B. de Schutter, and A. Bemporad. Equivalence of Hybrid Dynamical Models. Automatica, 37(7), pp. 1085-1091, July 2001.

[7] B. Heiming, J. Lunze Definition of the Three-Tank Benchmark Problem for Controller Reconfiguration. European Control Conference, 1999.

[8] J. Lunze. Benchmark for the Reconfiguration Problem: Laboratory 3 Tank System. Technical University Hamburg-Harbur. 1998. http://www.tu-arburg.de/rts/software/cosy

[9] F.D. Torrisi and A. Bemporad. Discrete-Time Hybrid Modeling and Verification. In *Proc. 40th IEEE Conf. On Decision and Control,* pp. 2899-2904 Orlando, Florida, 2001.

[10] F.D. Torrisi and A. Bemporad. HYSDEL Modeling and Simulation of Hybrid Dynamical Systems. Proceedings of MATHMOD Conference, Vienna University of Technology, Viena, Austria, February 5-7, 2003.

[11] J.L. Villa, M. Duque, A. Gauthier and N. Rakoto-Ravalontsalama Hybrid Modeling of Potable Water Treatment Plant. *In : Cabrera and Cabrera Jr. (Eds.) Pumps, Electromechanical Devices and Systems : Applied to Urban Water Management*. A.A. Balkema Publishers. Lisse, pp. 909-917. 2003.

[12] J.L. Villa, M. Duque, A. Gauthier and N. Rakoto-Ravalontsalama. Modeling and Control of a Water Treatment Plant. In: IEEE International Conference on Systems, Man & Cybernetics (SMC2003) October 5–8, Washington D.C., USA, pp. 171-176 (CD-ROM), 2003.

[13] J.L. Villa, M. Duque, A. Gauthier and N. Rakoto-Ravalontsalama. MLD Control of Hybrid Systems : Application to the Three-Tank Benchmark Problem. In : IEEE International Conference on Systems, Man & Cybernetics (SMC2003) October 5–8, Washington D.C., USA, pp. 666-671 (CD-ROM) 2003.

[14] J.L. Villa, M. Duque, A. Gauthier and N. Rakoto-Ravalontsalama. A New Algorithm for Translating MLD Systems into PWA Systems. Technical Report No 03/8/AUTO Ecole des Mines de Nantes, France. Sept. 2003.

Appendix B

Appendix 2 – Paper [J.5]:

- [J.5] E. Mojica, N. Quijano, and N. Rakoto-Ravalontsalama.
  A polynomial approach for optimal control of switched nonlinear systems,
  Int. Journal of Robust and Nonlinear Control, Wiley, 2014, 24 (12), pp.1797-1808.

INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust. Nonlinear Control (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.2964

A polynomial approach for optimal control of switched nonlinear systems

Eduardo Mojica-Nava[1,*,†], Nicanor Quijano[1] and Naly Rakoto-Ravalontsalama[2]

[1]*Universidad de Los Andes, Cra 1 No. 18A-10, Bogotá, Colombia*
[2]*Ecole des Mines de Nantes and IRCCyN, 44307 Nantes Cedex 03, France*

SUMMARY

Optimal control problems for switched nonlinear systems are investigated. We propose an alternative approach for solving the optimal control problem for a nonlinear switched system based on the theory of moments. The essence of this method is the transformation of a nonlinear, nonconvex optimal control problem, that is, the switched system, into an equivalent optimal control problem with linear and convex structure, which allows us to obtain an equivalent convex formulation more appropriate to be solved by high-performance numerical computing. Consequently, we propose to convexify the control variables by means of the method of moments obtaining semidefinite programs. Copyright © 2013 John Wiley & Sons, Ltd.

Received 4 May 2012; Revised 7 November 2012; Accepted 21 December 2012

KEY WORDS: method of moment; optimal control; switched systems

1. INTRODUCTION

Switched nonlinear control systems are characterized by a set of several continuous nonlinear state dynamics with a logic-based controller, which determines simultaneously a sequence of switching times and a sequence of modes. As performance and efficiency are key issues in modern technological system such as automobiles, robots, chemical processes, power systems among others, the design of optimal logic-based controllers, covering all those functionalities while satisfying physical and operational constraints, plays a fundamental role. In the last years, several researchers have considered the optimal control of switched systems. An early work on the problem is presented in [1], where a class of hybrid-state continuous-time dynamic system is investigated. Later, a generalization of the optimal control problem and algorithms of hybrid systems is presented [2]. The particular case of the optimal control problem of switched systems is presented in [3] and [4]. However, most of the efforts have been typically focused on linear subsystems [5]. In general, the optimal control problem of switched system is often computationally hard as it encloses both elements of optimal control as well as combinatorial optimization [6]. In particular, necessary optimality conditions for hybrid systems have been derived using general versions of the Maximum Principle [7, 8] and more recently in [9]. In the case of switching systems [4] and [6], the switched system has been embedded into a larger family of systems, and the optimization problem is formulated. For general hybrid systems, with nonlinear dynamics in each location and with autonomous and controlled switching, necessary optimality conditions have recently been presented in [10]; and using these conditions, algorithms based on the hybrid Maximum Principle have been derived. Focusing on real-time applications, an optimal control problem for switched dynamical systems is considered,

*Correspondence to: Eduardo Mojica-Nava, Universidad de Los Andes, Cra 1 No. 18A-10, Bogotá, Colombia.
†E-mail: ea.mojica70@uniandes.edu.co

where the objective is to minimize a cost functional defined on the state, and where the control variable consists of the switching times [11]. It is widely perceived that the best numerical methods available for hybrid optimal control problems involve mixed integer programming (MIP) [12, 13]. Even though great progress has been made in recent years in improving these methods, the MIP is an NP-hard problem, so scalability is problematic. One solution for this problem is to use the traditional nonlinear programming techniques such as sequential quadratic programming, which reduces dramatically the computational complexity over existing approaches [6].

The main contribution of this paper is an alternative approach to solve effectively the optimal control problem for an autonomous nonlinear switched system based on the probability measures introduced in [14], and later used in [15] and [16] to establish existence conditions for an infinite-dimensional linear program over a space of measure. Then, we apply the theory of moments, a method previously introduced for global optimization with polynomials in [17, 18], and later extended to nonlinear $0-1$ programs using an explicit equivalent positive semidefinite program in [19]. We also use some results recently introduced for optimal control problems with the control variable expressed as polynomials [20–22]. The first approach relating switched systems and polynomial representations can be found in [23]. The moment approach for global polynomial optimization based on semidefinite programming (SDP) is consistent, as it simplifies and/or has better convergence properties when solving convex problems. This approach works properly when the control variable (i.e., the switching signal) can be expressed as a polynomial. Essentially, this method transforms a nonlinear, nonconvex optimal control problem (i.e., the switched system) into an equivalent optimal control problem with linear and convex structure, which allows us to obtain an equivalent convex formulation more appropriate to be solved by high-performance numerical computing. In other words, we transform a given controllable switched nonlinear system into a controllable continuous system with a linear and convex structure in the control variable.

This paper is organized as follows. In Section 2, we present some definitions and preliminaries. A semidefinite relaxation using the moment approach is developed in Section 3. An algorithm is developed on the basis of the semidefinite approach in Section 4 with a numerical example to illustrate our approach, and finally in Section 5, some conclusions are drawn.

2. THE SWITCHED OPTIMAL CONTROL PROBLEM

2.1. Switched systems

The switched system adopted in this work has a general mathematical model described by

$$\dot{x}(t) = f_{\sigma(t)}(x(t)), \tag{1}$$

where $x(t)$ is the state, $f_i : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the $i-th$ vector field, $x(t_0) = x_0$ are fixed initial values, and $\sigma : [t_0, t_f] \mapsto \mathcal{Q} \in \{0, 1, 2, ..., q\}$ is a *piecewise constant* function of time, with t_0 and t_f as the initial and final times, respectively. Every mode of operation corresponds to a specific subsystem $\dot{x}(t) = f_i(x(t))$, for some $i \in \mathcal{Q}$, and the *switching signal* σ determines which subsystem is followed at each point of time into the interval $[t_0, t_f]$. The control input σ is a measurable function. In addition, we consider a non-Zeno behavior, that is, we exclude an infinite switching accumulation points in time. Finally, we assume that the state does not have jump discontinuities. Moreover, for the interval $[t_0, t_f]$, the control functions must be chosen so that the initial and final conditions are satisfied.

Definition 1
A control for the switched system in (1) is a duplet consisting of

(i) a finite sequence of modes, and
(ii) a finite sequence of switching times such that, $t_0 < t_1 < \cdots < t_q = t_f$.

DOI: 10.1002/rnc

2.2. Switched optimal control problem

Let us define the optimization functional in Bolza form to be minimized as

$$J = \varphi(x(t_{\mathrm{f}})) + \int_{t_0}^{t_{\mathrm{f}}} L_{\sigma(t)}(t, x(t))\mathrm{d}t, \tag{2}$$

where $\varphi(x(t_{\mathrm{f}}))$ is a real-valued function, and the running switched costs $L_{\sigma(t)} : \mathbb{R}^+ \times \mathbb{R}^n \mapsto \mathbb{R}$ are continuously differentiable for each $\sigma \in \mathcal{Q}$.

A switched optimal control problem (SOCP) can be stated in a general form as follows.

Definition 2
Given the switched system in (1) and a Bolza cost functional J as in (2), the SOCP is given by

$$\min_{\sigma(t)\in\mathcal{Q}} J(t_0, t_{\mathrm{f}}, x(t_0), x(t_{\mathrm{f}}), x(t), \sigma(t)) \tag{3}$$

subject to the state $x(\cdot)$ satisfying Equation (1).

The SOCP can have the usual variations of fixed or free initial or terminal state, free terminal time, and so forth.

2.3. A Polynomial representation

The starting point is to rewrite (1) as a continuous non-switched control system as it has been shown in [24]. The polynomial expression in the control variable able to mimic the behavior of the switched system is developed using a variable v, which works as a control variable.

A polynomial expression in the new control variable $v(t)$ can be obtained through Lagrange polynomial interpolation and a constraint polynomial as follows. First, let the Lagrange polynomial interpolation quotients be defined as [25],

$$l_k(v) = \prod_{\substack{i=0\\ i\neq k}}^{q} \frac{(v-i)}{(k-i)}. \tag{4}$$

The control variable is restricted by the set $\Omega = \{v \in \mathbb{R}\,|\,g(v) = 0\}$, where $g(v)$ is defined by

$$g(v) = \prod_{k=0}^{q} (v-k). \tag{5}$$

General conditions for the subsystems functions should be satisfied.

Assumption 3
The nonlinear switched system satisfies growth, Lipschitz continuity, and coercivity qualifications concerning the mappings

$$f_i : \mathbb{R}^n \mapsto \mathbb{R}^n$$

$$L_i : \mathbb{R}^n \mapsto \mathbb{R}$$

to ensure existence of solutions of (1).

The solution of this system may be interpreted as an explicit ODE on the manifold Ω. A related continuous polynomial system of the switched system (1) is constructed in the following proposition [24].

DOI: 10.1002/rnc

Proposition 4
Consider a switched system of the form given in (1). There exists a unique continuous state system with polynomial dependence in the control variable v, $\mathcal{F}(x, v)$ of degree q in v, with $v \in \Omega$ as follows:

$$\dot{x} = \mathcal{F}(x, v) = \sum_{k=0}^{q} f_k(x) l_k(v). \tag{6}$$

Then, this polynomial system is an equivalent polynomial representation of the switched system (1).

Similarly, we define a polynomial equivalent representation for the running cost $L_{\sigma(t)}$ by using the Lagrange's quotients as follows.

Proposition 5
Consider a switched running cost of the form given in (2). There exists a unique polynomial running cost equation $\mathcal{L}(x, v)$ of degree q in v, with $v \in \Omega$ as follows:

$$\mathcal{L}(x, v) = \sum_{k=0}^{q} L_k(x) l_k(v) \tag{7}$$

with $l_k(v)$ defined in (4). Then, this polynomial system is an equivalent polynomial representation of the switched running cost in (2).

The equivalent optimal control problem (EOCP), which is based on the equivalent polynomial representation is described next.

The functional using Equation (7) is defined by

$$J = \varphi(x(t_f)) + \int_{t_0}^{t_f} \mathcal{L}(x, v)\mathrm{d}t, \tag{8}$$

subject to the system defined in (6), with $x \in \mathbb{R}^n$, $v \in \Omega$, and $x(t_0) = x_0$, where $l_k(v)$, Ω, and $\mathcal{L}$ are defined earlier. Note that this control problem is a continuous polynomial system with the input constrained by a polynomial $g(v)$. This polynomial constraint is nonconvex with a disjoint feasible set, and traditional optimization solvers perform poorly on such equations, as the necessary constraint qualification is violated. This makes this problem intractable directly by traditional nonlinear optimization solvers. Next, we propose a convexification of the EOCP using the special structure of the control variable v, which improves the optimization process.

3. SEMIDEFINITE RELAXATION USING A MOMENTS APPROACH

3.1. Relaxation of the optimal control problem

We describe the relaxation of the polynomial optimal control problem, for which, regardless of convexity assumptions, existence of optimal solutions can be achieved. Classical relaxation results establish, under some technical assumptions, that the infimum of any functional does not change when we replace the integrand by its convexification. In the previous section, a continuous representation of the switched system has been presented. This representation has a polynomial form in the control variable, which implies that this system is nonlinear and nonconvex with a disjoint feasible set. Thus, traditional optimization solvers have a disadvantaged performance, either by means of the direct methods (i.e., nonlinear programming) or indirect methods (i.e., Maximum Principle). We propose then, an alternative approach to deal with this problem. The main idea of this approach is to convexify the control variable in polynomial form by means of the method of moments. This method has been recently developed for optimization problems in polynomial form (see [17, 18], among others). Therefore, a linear and convex relaxation of the polynomial problem (8) is presented next. The relaxed version of the problem is formulated in terms of probability measures associated with sequences of admissible controls [15].

DOI: 10.1002/rnc

Let Ω be the set of admissible controls $v(t)$. The set of probability measures associated to the admissible controls in Ω is

$$\Lambda = \left\{\mu = \{\mu_t\}_{t\in[t_0,t_f]} : \mathrm{supp}(\mu_t) \subset \Omega, \text{a.e.}, t \in [t_0, t_f]\right\},$$

where μ is a probability measure supported in Ω. The functional $J(x, v)$ defined on Λ is now given by

$$J(x, v) = \varphi(x(t_f)) + \int_{t_0}^{t_f} \int_{\Omega} \mathcal{L}(x(t), v)\mathrm{d}\mu_t(v)\mathrm{d}t,$$

where $x(t)$ is the solution of

$$\dot{x}(t) = \int_{\Omega} \mathcal{F}(x, v)\mathrm{d}\mu_t(v), \quad x(t_0) = x_0.$$

We have obtained a reformulation of the problem that is an infinite dimensional linear program and thus not tractable as it stands. However, the polynomial dependence in the control variable allows us to obtain a semidefinite program or linear matrix inequality relaxation, with finitely many constraints and variables. By means of moments variables, an equivalent convex formulation more appropriate to be solved by numerical computing can be rendered. The method of moments takes a proper formulation in probability measures of a nonconvex optimization problem ([18, 23], and references therein). Thus, when the problem can be stated in terms of polynomial expressions in the control variable, we can transform the measures into algebraic moments to obtain a new convex program defined in a new set of variables that represent the moments of every measure [17, 18, 22].

We define the space of moments as

$$\Gamma = \left\{m = \{m_k\} : m_k = \int_{\Omega} v^k \mathrm{d}\mu(v), \ \ \mu \in P(\Omega)\right\},$$

where $P(\Omega)$ is the convex set of all probability measures supported in Ω. In addition, a sequence $m = \{m_k\}$ has a representing measure μ supported in Ω only if these moments are restricted to be entries on positive semidefinite moments and localizing matrices [17, 19]. For this particular case, when the control variable is of dimension one, the moment matrix is a Hankel matrix with $m_0 = 1$, that is, for a moment matrix of degree d, we have

$$M_d(m) = \begin{bmatrix} m_0 & m_1 & \cdots & m_d \\ m_1 & m_2 & \cdots & m_{d+1} \\ \vdots & \vdots & \cdots & \vdots \\ m_d & m_{d+1} & \cdots & m_{2d} \end{bmatrix}.$$

The localizing matrix is defined on the basis of corresponding moment matrix, whose positivity is directly related to the existence of a representing measure with support in Ω as follows. Consider the set Ω defined by the polynomial $\beta(v) = \beta_0 + \beta_1 v + \cdots \beta_d v^\eta$. It can be represented in moment variables as $\beta(m) = \beta_0 + \beta_1 m_1 + \cdots \beta_\eta m_\eta$, or in compact form as $\beta(m) = \sum_{\gamma=0}^{\eta} \beta_\gamma m_\gamma$. Suppose that the entries of the corresponding moment matrix are m_ρ, with $\rho \in [0, 1, \ldots, 2d]$. Thus, every entry of the localizing matrix is defined as $l_\rho = \sum_{\gamma=0}^{d} \beta_\gamma m_{\gamma+\rho}$. Note that the localizing matrix has the same dimension of the moment matrix, that is, if $d = 1$ and the polynomial $\beta = v + 2v^2$, then the moment and localizing matrices are

$$M_1(m) = \begin{bmatrix} 1 & m_1 \\ m_1 & m_2 \end{bmatrix}, \ M_1(\beta m) = \begin{bmatrix} m_1 + 2m_2 & m_2 + 2m_3 \\ m_2 + 2m_3 & m_3 + 2m_4 \end{bmatrix}.$$

More details on the method of moments can be found in [19, 26].

Because J is a polynomial in v of degree q, the criterion $\int \mathcal{L}\mathrm{d}\mu$ involves only the moments of μ up to order q and is linear in the moment variables. Hence, we replace μ with the finite sequence

DOI: 10.1002/rnc

$m = \{m_k\}$ of all its moments up to order q. We can then express the linear combination of the functional J and the space of moments Γ as follows

$$\min_{v\in\Omega} J(x,v) \rightarrow \min_{\mu\in P(\Omega)} \int_\Omega J(x,v)\mathrm{d}\mu(v)$$

$$= \min_{m_k\in\Gamma} \int_{t_0}^{t_\mathrm{f}} \sum_i \sum_k L_i(x)\alpha_{ik} m_k, \tag{9}$$

where α_{ik} are the coefficients resulting of the factorization of Equation (4). Similarly, we obtain the convexification of the state equation

$$\dot{x}(t) = \int_\Omega \mathcal{F}(x,v)\mathrm{d}\mu(v) = \sum_i \sum_k f_i(x)\alpha_{ik} m_k. \tag{10}$$

We have now a problem in moment variables, which can be solved by efficient computational tools as it is shown in the next section.

3.2. Semidefinite programs for the EOCP

We can use the functional and the state equation with moment structure to rewrite the relaxed formulation as a SDP. First, we need to redefine the control set Ω to be coherent with the definitions of localizing matrix and representation results. We treat the polynomial $g(v)$ as two opposite inequalities, that is, $g_1(v) = g(v) \geqslant 0$ and $g_2(v) = -g(v) \geqslant 0$, and we redefine the compact set to be $\Omega = \{g_i(v) \geqslant 0, \;\; i = 1,2\}$. Also, we define also a prefixed order of relaxation, which is directly related to the number of subsystems.

Let w be the degree of the polynomial $g(v)$, which is equivalent to the degree of the polynomials g_1 and g_2. Considering its parity, we have that if w is even (odd) then $r = w/2$ ($r = (w+1)/2$). In this case, r corresponds to the prefixed order of relaxation. We use a direct transcription method to obtain an SDP to be solved through a nonlinear programming (NLP) algorithm [27]. Using a discretization method, the first step is to split the time interval $[t_0, t_\mathrm{f}]$ into N subintervals as $t_0 < t_1 < t_2 < \ldots < t_N = t_\mathrm{f}$, with a time step h predefined by the user. The integral term in the functional is implicitly represented as an additional state variable, transforming the original problem in Bolza form into a problem in Mayer form, which is a standard transformation [27]. Therefore, we obtain a set of discrete equations in moment variables. In this particular case, we have used a trapezoidal discretization, but we could have used a more elaborated discretization scheme. Thus, the optimal control problem can be formulated as an SDP.

Consider a fixed t in the time interval $[t_0, t_\mathrm{f}]$ and let Assumption 3 holds. We can state the following SDP of relaxation order r (SDP_r).

Semidefinite program-SDP_r: For every $j = \{1, 2, \ldots, N\}$, a semidefinite program SDP_r can be described by

$$\begin{aligned} J_r^* &= \min_{m(t_j)} \frac{h}{2} \sum_{j=0}^{N-1} \mathcal{L}(x(t_j), m(t_j)) \\ &\text{s.t.} \\ x(t_{j+1}) &= x(t_j) + h \sum_i \sum_k f_i(x(t_j))\alpha_{ik} m_k(t_j), \quad x(t_0) = x_0, \\ &M_r(m(t_j)) \succeq 0, \; M_0(g_1 m(t_j)) \succeq 0, \; M_0(g_2 m(t_j)) \succeq 0. \end{aligned} \tag{11}$$

Notice that in this case, the localizing matrices are linear. Let us consider the two subsystems case, that is, we have $g = v^2 - v$ that leads to polynomials $g_1 = v^2 - v$ and $g_2 = v - v^2$, thus $w = \deg g = 2$. The localizing matrices are $M_0(g_1 m) = m_2 - m_1$, so $M_0(g_2 m) = m_1 - m_2$.

DOI: 10.1002/rnc

This happens because we are using the minimum order of relaxation, $r = w/2$ or $r = (w+1)/2$ depending on its parity. It is also known that the optimum J_r^* is not always an optimal solution. However, in this case, a suboptimal solution is obtained, which corresponds to a lower bound on the global optimum J^* of the original problem. If we are interested in searching for an optimal solution, we can use a higher order of relaxation, that is, $r \geqslant w/2$, but the number of moment variables will increase, which can make the problem numerically inefficient. However, in many cases, low order relaxations will provide the optimal value J^* as shown in the next section, where we use a criterion to test whether the SDP_r relaxation achieves the optimal value J^* for a fixed time. Still, suboptimal solutions of the original problem are obtained in the iteration that can be used. In order to solve a traditional NLP, we use the characteristic form of the moment and localizing matrices. We know that the moment matrices, and so the localizing matrices, are symmetric positive definite, which implies that every principal subdeterminant is positive [21]. Then, we use the set of subdeterminants of each matrix as algebraic constraints.

3.3. Analysis of solutions

Once a solution has been obtained in a subinterval $[t_{j-1}, t_j]$, we obtain a vector of moments $m^*(t_j) = [m_1^*(t_j), m_2^*(t_j), \dots, m_r^*(t_j)]$. Then, we need to verify if we have attained an optimal solution. On the basis of a rank condition of the moment matrix [26], we can test if we have obtained a global optimum at a relaxation order r. Also, on the basis of the same rank condition, we can check whether the optimal solution is unique or if it is a convex combination of several minimizers. The next result is based on an important result presented in [26] and used in [19] for optimization of $0-1$ problems.

Proposition 6
For a fixed time t_j in the interval $[t_0, t_f]$, the SDP_r (11) is solved with an optimal vector solution $m^*(t_j)$, if

$$\nu_r = \operatorname{rank} M_r\left(m^*(t_j)\right) = \operatorname{rank} M_0\left(m^*(t_j)\right), \tag{12}$$

then the global optimum has been reached and the problem for the fixed time t_j has ν_r optimal solutions.

Note that the rank condition (12) is a sufficient condition, which implies that the global optimum could be reached at some relaxation of order r and still the $\operatorname{rank} M_r > \operatorname{rank} M_0$. It should be noted that for the particular case of minimum order of relaxation, the rank condition yields $\nu_r = \operatorname{rank} M_r(m(t_j)) = \operatorname{rank} M_0(m(t_j)) = 1$, because $M_0 = 1$. Then, the $\operatorname{rank} M_0 = 1$, which implies that when $\nu_r > 1$, that is, several solutions arise. In this case, we obtain a suboptimal switching solution.

Using the previous result, we can state some relations between solutions that can be used to obtain the switching signal in every t_j. First, we state the following result valid for the unique solution case.

Theorem 7
If Problem (11) is solved for a fixed $t_j \in [t_0, t_f]$ and the rank condition in (12) is verified with $\nu_r = \operatorname{rank} M_r(m^*(t_j)) = 1$, then the vector of moments $m^*(t_j)$ has attained a unique optimal global solution; and therefore, the optimal switching signal of the switched problem (3) for the fixed time t_j is obtained as

$$\sigma^*(t_j) = m_1^*(t_j), \tag{13}$$

where $m_1^*(t_j)$ is the first moment of the vector of moments $m^*(t_j)$.

DOI: 10.1002/rnc

Proof

Suppose the problem (11) has been solved for a fixed t_j, and a solution has been obtained. Let $m^*(t_j)$ be the solution obtained and the rank condition (12) has been verified. From a result presented in [19], it follows that

$$\min_{\mu \in P(\Omega)} \int_\Omega J(x,v)\mathrm{d}\mu(v) = \min_{m_k \in \Gamma} \int_{t_0}^{t_\mathrm{f}} \sum_i \sum_k L_i(x)\alpha_{ik} m_k,$$

where $m^*(t_j) = \left[m_1^*, \ldots, m_r^*\right]$ is the vector of moments of some measure μ_m. But then, as μ_m is supported on Ω, it also follows that $m^*(t_j)$ is an optimal solution and because of rank $M_r\left(m^*(t_j)\right) = 1$, this solution is unique and it is the solution of the polynomial problem (8). Then, we know that every optimal solution v^* corresponds to

$$m^*(t_j) = \left(v^*(t_j), \left(v^*(t_j)\right)^2, \ldots, \left(v^*(t_j)\right)^{2d}\right),$$

which implies that $m_1^*(t_j) = v^*(t_j)$. Now, using the equivalence stated in Proposition 5, we know that the solutions of the polynomial Problem (8) are solutions of the switching system; and in this case, it is only one. Hence, we obtain $\sigma^*(t_j) = v^*(t_j)$, which implies that $\sigma^*(t_j) = v^*(t_j) = m_1^*(t_j)$, where m_1^* is the first moment of the vector of moments. □

Remark 8

Switched linear systems case. When we have a switched linear system, that is, when each subsystem is defined by a linear system, results presented in Theorem (7) can be directly applied, because Assumption (3) is satisfied for linear systems because the Lipschitz condition is satisfied globally [28]. Also, we can notice that if the switched linear system has one and only one switching solution, it corresponds to the first moment solution of the SDP_r program for all $t \in [t_0, t_\mathrm{f}]$, that is, $m_1^*(t_j) = \sigma^*(t_j)$, for all $t_j \in [t_0, t_\mathrm{f}]$. This can be verified by means of the rank condition (12), which should be $\nu_r = 1$, for all $t \in [t_0, t_\mathrm{f}]$.

This result states a correspondence between the minimizer of the original switched problem and the minimizer of the SDP_r, and it can be used to obtain a switching signal directly from the solution of the SDP_r. However, it is not always the case. Sometimes, we obtain a non-optimal solution that arises when the rank condition is not satisfied, that is, $\nu_r > 1$. But, we still can use information from the solution to obtain a switching suboptimal solution. In [29], a sum up rounding strategy is presented to obtain a suboptimal switched solution from a relaxed solution in the case of mixed-integer optimal control. We use a similar idea but extended to the case when the relaxed solution is any integer instead of the binary case.

Consider the first moment $m_1(\cdot) : [t_0, t_\mathrm{f}] \mapsto [0, q]$, which is a relaxed solution of the NLP problem for t_j when the rank condition is not satisfied. We can state a correspondence between the relaxed solution and a suboptimal switching solution, which is close to the relaxed solution in average and is given by

$$\sigma(t_j) = \begin{cases} \lceil m_1(t_j) \rceil & \text{if } \displaystyle\int_{t_0}^{t_j} m_1(\tau)\mathrm{d}\tau - \delta t \sum_{k=0}^{j-1} \sigma(t_k) \geqslant 0.5\delta t \\ \lfloor m_1(t_j) \rfloor & \text{otherwise} \end{cases} \tag{14}$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are the ceiling and floor functions, respectively.

4. A SWITCHED OPTIMIZATION ALGORITHM

The ideas presented earlier are summarized in the following algorithm, which is implemented in Section 4.2 on a simple numerical example presented as a benchmark in [30]. The core of the algorithm is the inter-relationship of three main ideas:

DOI: 10.1002/rnc

(i) **The equivalent optimal control problem**
The EOCP is formulated as in Section 2, where the equivalent representation of the switched system and the running cost are used to obtain a polynomial continuous system.

(ii) **The relaxation of the EOCP – the theory of moments**
The EOCP is now transformed into an SDP of order of relaxation r, which can be solved numerically efficiently. We obtain an equivalent linear convex formulation in the control variable.

(iii) **The relationship between the solutions of the original switched problem and the SDP solutions**
The solutions of the SDP_r for each $t_j \in [t_0, t_f]$ are obtained; and through an extracting algorithm, the solutions of the original problem are obtained.

4.1. *Algorithm SDP_r – SOCP*

The optimal control pseudo-code algorithm for the switched systems is shown in Algorithm 1.

Algorithm 1 Algorithm SDP_r - SOCP

1: **Algorithm initialization.** Obtain the equivalent representation of the optimal switched problem using Propositions 4 and 5. Redefine the EOCP in moments variables using Equations (9) and (10), obtaining the corresponding moment and localizing matrices. Use a direct transcription method to obtain a SDP to be solved.
2: Split the time interval $[t_0, t_f]$ into N subintervals with points $t_0 < t_1 < \cdots < t_N = t_f$, with a time step h and set $j = 1$.
3: Set the minimum relaxation order r as explained in Section 3.1.
4: Solve the SDP_r described by (11).
5: Verify the rank condition expressed in Equation (12) and Proposition 6.
6: **if** $w_r = 1$ **then** $\sigma^*(t_j) = m_1^*(t_j)$.
7: **else**
8:     use Equation (14) and fix $\sigma^*(t_j) = \lceil m_1^*(t_j) \rceil$ or $\sigma^*(t_j) = \lfloor m_1^*(t_j) \rfloor$ as it corresponds.
9: **end if**
10: Set $j = j + 1$
11: **repeat**
12:     From Step 3
13: **until** $j = N$
14: Set $J^* = \sum_j J(t_j)$.
15: **return** J^*.

In the next section, we present a numerical example to illustrate the results presented in this work.

4.2. *Numerical example: Lotka–Volterra problem*

We present an illustrative example of a switched nonlinear optimal control problem reformulated as a polynomial optimal control problem. Then, this reformulation allows us to apply the semidefinite relaxation based on the theory of moments. We illustrate an efficient computational treatment to study the optimal control problem of switched systems reformulated as a polynomial expression.

We deal with the Lotka–Volterra fishing problem. Basically, the idea is to find an optimal strategy on a fixed time horizon to bring the biomass of both predator as prey fish to a prescribed steady-state. The system has two operation modes and a switching signal as a control variable. The optimal integer control shows chattering behavior, which makes this problem a benchmark to test different types of algorithms[‡].

[‡]The problem has been used as a small-scale benchmark problem for the evaluation of algorithms [30].

DOI: 10.1002/rnc

The Lotka–Volterra model, also known as the predator–prey model, is a coupled nonlinear differential equations where the biomasses of two fish species are the differential states x_1 and x_2, the binary control is the operation of a fishing fleet, and the objective is to penalize deviation from a steady-state. The optimal control problem is described as follows:

$$
\begin{aligned}
&\min_u \int_{t_0}^{t_f} (x_1-1)^2+(x_2-1)^2\mathrm{d}t\\
&\text{s.t.}\\
&\dot{x}_1 = x_1 - x_1x_2 - 0.4x_1u\\
&\dot{x}_2 = -x_2 + x_1x_2 - 0.2x_2u\\
&x(0) = (0.5, 0.7)^\top,\ \ u(t)\in\{0,1\},\ \ t\in[0,12].
\end{aligned}
\tag{15}
$$

The problem can be represented by the approach described earlier. Consider a subsystem f_0 when the control variable takes value 0, and a subsystem f_1 when the control variable takes value 1. This leads to a two operation modes and a switching control variable $\sigma(\cdot) : [0,12] \mapsto \{0,1\}$. Thus, by means of the algorithm SDP_r–EOCP, an SDP program can be stated. First, we define the order of relaxation as $r = w/2 = 1$; the constraint control set as $\Omega = \{g_i(v) \geqslant 0, g_1(v) = v^2 - v,\ g_2 = v - v^2\}$; the moment matrix with $r = 1$, $M_1(m)$; and the localizing matrices, $M_0(g_1m)$ and $M_0(g_2m)$. Using the set Ω and the moment and localizing matrices, we set the problem in moment variables obtaining the positive semidefinite program (SDP_r). Solving the SDP_r program for each $t \in [0,12]$, with a step time h, we obtain an optimal trajectory, and the moment sequence allows us to calculate the switching signal.

Figure 1 shows the trajectories, the relaxed moment solution, and the switching signal obtained for an order of relaxation $r = 1$. It can be appreciated that when the relaxed solution has a unique optimal solution, that is, when the rank condition is satisfied, the relaxed solution has an exact unique solution that is integer and corresponds to the switching signal, which shows the validity of Theorem 7. Also, it is shown that when the rank condition is not satisfied, the algorithm proposed gives a suitable solution, that in average is close to the relaxed solution. The algorithm has shown that even if there is no global optimal solution, a local suboptimal solution is found. Furthermore, for the intervals where there is no optimal solution, a suboptimal solution has been found using the relaxed solution. In comparison with traditional algorithms, where a global suboptimal solution based on a relaxation is found, the proposed algorithm is able to detect whether an optimal solution is found in a time interval, which implies that if the system is composed by convex functions,

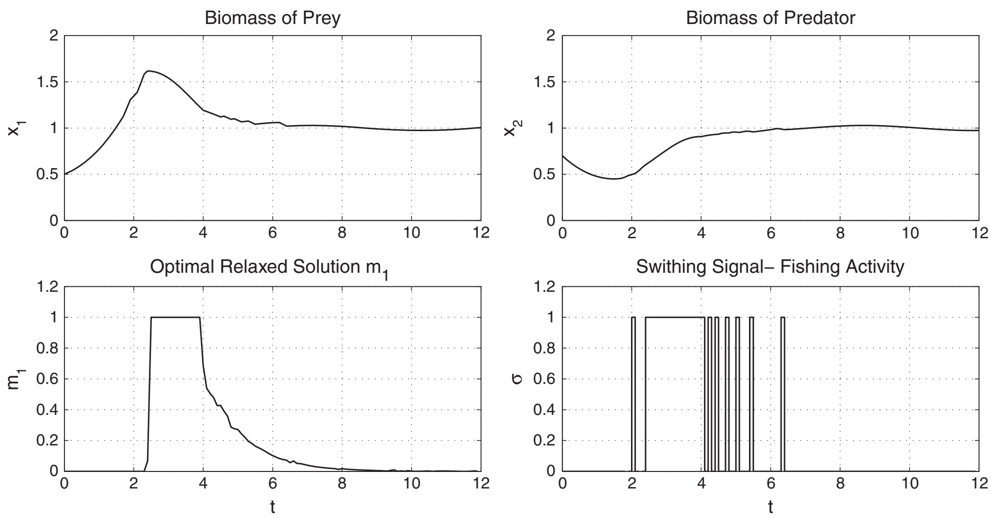


Figure 1. States and switching signal for the Lotka–Volterra example.

DOI: 10.1002/rnc

a global optimal solution is found. The computational efficiency is based on the semidefinite methods of solutions.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have developed a new method for solving the optimal control problem of switched nonlinear systems based on a polynomial approach. First, we transform the original problem into a polynomial system, which is able to mimic the switching behavior with a continuous polynomial representation. Next, we transform the polynomial problem into a relaxed convex problem using the method of moments. From a theoretical point of view, we have provided sufficient conditions for the existence of the minimizer by using particular features of the relaxed, convex formulation. Even in the absence of classical minimizers of the switched system, the solution of its relaxed formulation provides minimizers. We have introduced the moment approach as a computational useful tool to solve this problem, which has been illustrated by means of a classical example used in switched systems. As a future work, the algorithm can be extended to the case when an external control input and the switching signal should be obtained.

ACKNOWLEDGEMENTS

This study was supported by Proyecto CIFI 2011, Facultad de Ingeniería, Universidad de Los Andes.

REFERENCES

1. Witsenhausen H. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control* 1966; **11**(2):161–167.
2. Branicky MS, Borkar VS, Mitter SK. A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control* 1998; **43**(1):31–45.
3. Xu X, Antsaklis PJ. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Transactions on Automatic Control* 2004; **49**(1):2–16.
4. Bengea SC, DeCarlo RA. Optimal control of switching systems. *Automatica* 2005; **41**(1):11–27.
5. Das T, Mukherjee R. Optimally switched linear systems. *Automatica* 2008; **44**(5):1437–1441.
6. Wei S, Uthaichana K, Žefran M, DeCarlo R, Bengea S. Applications of numerical optimal control to nonlinear hybrid systems. *Nonlinear Analysis: Hybrid Systems* 2007; **1**(2):264–279.
7. Sussmann H. A maximum principle for hybrid optimal control problems. *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, 1999; **1**: 425–430.
8. Riedinger P, Daafouz J, Iung C. Suboptimal switched controls in context of singular arcs. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, 2003; 6254–6259.
9. Garavello M, Piccoli B. Hybrid necessary principle. *Proceeding of the 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, 2005; 723–728.
10. Shaikh MS, Caines PE. On the hybrid optimal control problem: theory and algorithms. *IEEE Transactions on Automatic Control* 2007; **52**(9):1587–1603.
11. Egerstedt M, Wardi Y, Delmotte F. Optimal control of switching times in switched dynamical systems. *Proceedings of the 42nd IEEE Conference on Decision and Control*, Hawaii, 2003; **3**: 2138 –2143.
12. Bemporad A, Morari M. Control of systems integrating logic, dynamics, and constraints. *Automatica* 1999; **35**(3):407–427.
13. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit solution of model predictive control via multiparametric quadratic programming. *Proceedings of the American Control Conference*, Chicago, 2000; 872–876.
14. Young LC. *Lectures on the Calculus of Variations and Optimal Control Theory*. Chelsea Publishing Co: New York, 1969.
15. Pedregal P. *Parametrized Measures and Variational Principles*. Birkhäuser: Basel, 1997.
16. Muñoz J, Pedregal P. A refinement on existence results in nonconvex optimal control. *Nonlinear Analysis* 2001; **46**(3):381–398.
17. Lasserre J. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* 2001; **11**:796–817.
18. Meziat R. The method of moments in global optimization. *Journal of Mathematics Science* 2003; **116**:3303–3324.
19. Lasserre JB. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization* 2002; **12**:756–769.
20. Lasserre JB. A semidefinite programming approach to the generalized problem of moments. *Mathematical Programming* 2008; **112**(1):65–92.

21. Meziat R, Patiño D, Pedregal P. An alternative approach for non-linear optimal control problems based on the method of moments. *Computational Optimization and Applications* 2007; **38**:147–171.
22. Pedregal P, Tiago J. Existence results for optimal control problems with some special nonlinear dependence on state and control. *SIAM Journal on Control and Optimization* 2009; **48**:415–437.
23. Mojica-Nava E, Meziat R, Quijano N, Gauthier A, Rakoto-Ravalontsalama N. Optimal control of switched systems: a polynomial approach. *Proceedings of the 17th IFAC World Congress*, Seoul, 2008; 7808–7813.
24. Mojica-Nava E, Quijano N, Rakoto-Ravalontsalama N, Gauthier A. A polynomial approach for stability analysis of switched systems. *Systems & Control Letters* 2010; **59**(2):98–104.
25. Burden R, Faires JD. *Numerical Analysis*. PWS: Boston, 1985.
26. Curto RE, Fialkow LA. The truncated complex K-moment problem. *Transactions of the American Mathematical Society* 2000; **352**(6):2825–2856.
27. Bertsekas DP. *Nonlinear Programming*, (2nd edn). Athena Scientific: Belmont, MA, 1999.
28. Khalil H. *Nonlinear Systems*, (3rd edn). Prentice Hall: Upper Saddle River, NJ, 2002.
29. Sager S. Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. *Journal of Process Control* 2009; **19**(8):1238–1247.
30. Sager S. *Numerical Methods for Mixed-integer Optimal Control Problems*. Der Andere Verlag: Tönning, 2005.

Appendix C

Appendix 3 – Paper [J.9]:

- [J.9] G. Obando, N. Quijano, and N. Rakoto-Ravalontsalama.
  A Center-Free Approach for Resource Allocation with Lower Bounds,
  Int. Journal of Control, 2016. DOI: 10.1080/00207179.2016.1225167.

INTERNATIONAL JOURNAL OF CONTROL, 2016
http://dx.doi.org/10.1080/00207179.2016.1225167

A centre–free approach for resource allocation with lower bounds

Germán Obando[a,b], Nicanor Quijano[a] and Naly Rakoto–Ravalontsalama[b]

[a]Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Colombia; [b]Ecole des Mines de Nantes, and IRCCyN, 44307 Nantes Cedex 03, France

ABSTRACT
Since complexity and scale of systems are continuously increasing, there is a growing interest in developing distributed algorithms that are capable to address information constraints, specially for solving optimisation and decision-making problems. In this paper, we propose a novel method to solve distributed resource allocation problems that include lower bound constraints. The optimisation process is carried out by a set of agents that use a communication network to coordinate their decisions. Convergence and optimality of the method are guaranteed under some mild assumptions related to the convexity of the problem and the connectivity of the underlying graph. Finally, we compare our approach with other techniques reported in the literature, and we present some engineering applications.

ARTICLE HISTORY
Received 31 December 2015
Accepted 11 August 2016

KEYWORDS
Distributed optimisation; resource allocation; multi-agent systems

1. Introduction

The increasing scale and complexity of systems have motivated the development of distributed methods to deal with situations where optimisation and decision-making are required. An important issue within this field is the optimal resource allocation over networks of agents, a problem that is closely related to network utility maximisation (NUM) problems (Palomar & Chiang, 2006; Tan, Zhu, Ge, & Xiong, 2015). Resource allocation arises when there is a limited amount of a certain resource (e.g. electric power, computing capacity or execution time), and it is necessary to establish an optimal distribution policy between some entities (e.g. loads, processors or controllers) that are connected by a communication network. This kind of problems has a large number of applications in economics (Ayesta, Erausquin, Ferreira, & Jacko, 2016; Conrad, 2010), smart energy systems (Hansen, Roche, Suryanarayanan, Maciejewski, & Siegel, 2015; Pantoja & Quijano, 2012), cloud computing (Pietrabissa et al., 2016; Pillai & Rao, 2016), and communications (H. Lee, K.J. Lee, Kim, Clerckx, & I. Lee, 2016; Tan et al., 2015).

Although there exists an extensive literature regarding distributed methods for solving resource allocation problems, this field still attracts considerable research attention (Cherukuri & Cortés, 2015; Obando, Pantoja, & Quijano, 2014; Pantoja, Quijano, & Passino, 2014; Poveda & Quijano, 2015; Ramirez-Llanos & Martinez, 2015; Tan, Yang, & Xu, 2013). Most of the solution methods are based on multi-agent systems (e.g. a survey that deals with the general class of NUM problems can be found in Palomar & Chiang, 2006), where the agents make decisions based on local information in order to obtain a desirable global behaviour. Appropriate coordination of agents is crucial because it avoids converging to suboptimal solutions. In order to ensure this coordination, a large number of methods require either the inclusion of a centralised agent or the use of restrictive information structures (as it is pointed out in Mosk-Aoyama, Roughgarden, & Shah, 2010). For instance, in classic decomposition techniques (Bemporad, Heemels, & Johansson, 2010; Boyd, Parikh, Chu, Peleato, & Eckstein, 2010; Palomar & Chiang, 2006), the Lagrange multiplier related to the 'price' of the resource is centrally adjusted to reach the optimum. By contrast, other methods are fully decentralised (e.g. Barreiro-Gomez, Obando, & Quijano, 2016; Xiao & Boyd, 2006; Zhu & Martinez, 2012). These methods exploit the communication capabilities of the agents to coordinate their decisions based on the information received from their neighbours. Fully decentralised methodologies have important advantages, among which we highlight the increase of the autonomy and resilience of the whole system since the dependence on a central authority is avoided.

In this paper, we propose a distributed resource allocation algorithm that does not require a central coordinator. An important characteristic of our method is the capability of handling lower bounds on the decision variables. This feature is crucial in a large number of

CONTACT Germán Obando ge-oband@uniandes.edu.co

practical applications, e.g. in Conrad (2010), Pantoja and Quijano (2012), and Lee et al. (2016), where it is required to capture the non-negativity of the resource allocated to each entity. We use a Lyapunov-based analysis in order to prove that the proposed algorithm asymptotically converges to the optimal solution under some mild assumptions related to the convexity of the cost function, and the connectivity of the graph that represents the communication topology. In order to illustrate our theoretical results, we perform some simulations and compare our method with other techniques reported in the literature. Finally, we present two engineering applications of the proposed algorithm. The first one seeks to improve the energy efficiency in large-scale air-conditioning systems. The second one is related to the distributed computation of the Euclidean projection onto a given set.

Our approach is based on a continuous time version of the *centre–free* algorithm presented in Xiao and Boyd (2006). The key difference is that the method in Xiao and Boyd (2006) does not allow the explicit inclusion of lower bounds on the decision variables, unless they are added by means of barrier functions (either logarithmic or exact; Cherukuri & Cortés, 2015). The problem of using barrier functions is that they can adversely affect the convergence time (in the case of using exact barrier functions) and the accuracy of the solution (in the case of using classic logarithmic barrier functions), especially for large-scale problems (Jensen & Bard, 2003). There are other methods that consider lower bound constraints in the problem formulation. For instance, Dominguez-Garcia, Cady, and Hadjicostis (2012) and Tan et al. (2013) have developed a decentralised technique based on broadcasting and consensus to optimally distribute a resource considering capacity constraints on each entity in the network. Nonetheless, compared to our algorithm, the approach in Dominguez-Garcia et al. (2012) and Tan et al. (2013) is only applicable to quadratic cost functions. On the other hand, Pantoja and Quijano (2012) propose a novel methodology based on population dynamics. The main drawback of this technique is that its performance is seriously degraded when the number of communication links decreases. We point out the fact that other distributed optimisation algorithms can be applied to solve resource allocation problems, as those presented in Nedic, Ozdaglar, and Parrilo (2010), Yi, Hong, and Liu (2015), and Johansson and Johansson (2009). Nevertheless, the underlying idea in these methods is different from the one used in our work, i.e. Nedic et al. (2010), Yi et al. (2015), and Johansson and Johansson (2009) use consensus steps to refine an estimation of the system state, while in our approach, consensus is used to equalise a quantity that depends on both the marginal cost perceived by each agent in the network and the Karush–Kuhn–Tucker (KKT) multiplier related to the corresponding resource's lower bound. In this regard, it is worth noting that the method studied in this paper requires less computational capability than the methods mentioned above. Finally, there are other techniques based on game theory and mechanism design (Kakhbod & Teneketzis, 2012; Sharma & Teneketzis, 2009) that decompose and solve resource allocation problems. Nonetheless, those techniques need that each agent broadcasts a variable to all the other agents, i.e. a communication topology given by a complete graph is required. In contrast, the method developed in this paper only uses a communication topology given by a connected graph, which generally requires lower infrastructure.

The remainder of this paper is organised as follows. Section 2 shows preliminary concepts related to graph theory. In Section 3, the resource allocation problem is stated. Then, in Section 4, we present our distributed algorithm and the main results on convergence and optimality. A comparison with other techniques reported in the literature is performed in Section 5. In Section 6, we describe two applications of the proposed method: (i) the optimal chiller loading problem in large-scale air-conditioning systems, and (ii) the distributed computation of Euclidean projections. Finally, in Sections 7 and 8, arguments and conclusions of the developed work are presented.

2. Preliminaries

First, we describe the notation used throughout the paper and presents some preliminary results on graph theory that are used in the proofs of our main contributions.

In the multi-agent framework considered in this article, we use a graph to model the communication network that allows the agents to coordinate their decisions. A graph is mathematically represented by the pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges connecting the nodes. $\mathcal{G}$ is also characterised by its adjacency matrix $\mathcal{A} = [a_{ij}]$. The adjacency matrix $\mathcal{A}$ is an $n \times n$ non-negative matrix that satisfies: $a_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$, and $a_{ij} = 0$ if and only if $(i, j) \notin \mathcal{E}$. Each node of the graph corresponds to an agent of the multi-agent system, and the edges represent the available communication channels (i.e. $(i, j) \in \mathcal{E}$ if and only if agents i and j can share information). We assume that there is no edges connecting a node with itself, i.e. $a_{ii} = 0$, for all $i \in \mathcal{V}$; and that the communication channels are bidirectional, i.e. $a_{ij} = a_{ji}$. The last assumption implies that $\mathcal{G}$ is undirected. Additionally, we denote by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$, the set of neighbours of node i, i.e. the set of nodes that are able to receive/send information from/to node i.

Let us define the $n \times n$ matrix $L(\mathcal{G}) = [l_{ij}]$, known as the graph Laplacian of $\mathcal{G}$, as follows:

$$l_{ij} = \begin{cases} \sum_{j \in \mathcal{V}} a_{ij} & \text{if } i = j \\ -a_{ij} & \text{if } i \neq j. \end{cases} \tag{1}$$

Properties of $L(\mathcal{G})$ are related to connectivity characteristics of $\mathcal{G}$ as shown in the following theorem. We remark that a graph $\mathcal{G}$ is said to be connected if there exists a path connecting any pair of nodes.

Theorem 2.1 (adapted from Godsil & Royle, 2001)**:** *An undirected graph $\mathcal{G}$ of order n is connected if and only if rank$(L(\mathcal{G})) = n - 1$.*

From Equation (1), it can be verified that $L(\mathcal{G})\mathbb{1} = \mathbf{0}$, where $\mathbb{1} = [1, \ldots, 1]^\top$, $\mathbf{0} = [0, \ldots, 0]^\top$. A consequence of this fact is that $L(\mathcal{G})$ is a singular matrix. However, we can modify $L(\mathcal{G})$ to obtain a nonsingular matrix as shown in the following lemma.

Lemma 2.1: *Let $L^{k_r}(\mathcal{G}) \in \mathbb{R}^{(n-1)\times n}$ be the submatrix obtained by removing the kth row of the graph Laplacian $L(\mathcal{G})$, and let $L^k(\mathcal{G}) \in \mathbb{R}^{(n-1)\times(n-1)}$ be the submatrix obtained by removing the kth column of $L^{k_r}(\mathcal{G})$. If $\mathcal{G}$ is connected, then $L^k(\mathcal{G})$ is positive definite. Furthermore, the inverse matrix of $L^k(\mathcal{G})$ satisfies $(L^k(\mathcal{G}))^{-1}\mathbf{l}_k^{k_r} = -\mathbb{1}$, where $\mathbf{l}_k^{k_r}$ is the kth column of the matrix $L^{k_r}(\mathcal{G})$.*

Proof: First, notice that $L(\mathcal{G})$ is a symmetric matrix because $\mathcal{G}$ is an undirected graph. Moreover, notice that according to Equation (1), $L(\mathcal{G})$ is diagonally dominant with non-negative diagonal entries. The same holds for $L^k(\mathcal{G})$ since this is a sub-matrix obtained by removing the kth row and column of $L(\mathcal{G})$. Thus, to show that $L^k(\mathcal{G})$ is positive definite, it is sufficient to prove that $L^k(\mathcal{G})$ is nonsingular.

According to Theorem 2.1, since $\mathcal{G}$ is connected, $L(\mathcal{G})$ has exactly $n-1$ linearly independent columns (resp. rows). Let us show that the kth column (resp. row) of $L(\mathcal{G})$ can be obtained by a linear combination of the other columns (resp. rows), i.e. the kth column (resp. row) is not linearly independent of the rest of the columns (resp. rows).

Since $L(\mathcal{G})\mathbb{1} = 0$, notice that $l_{ik} = -\sum_{j \in \mathcal{V}, j \neq k} l_{ij}$, for all $i \in \mathcal{V}$, i.e. the kth column can be obtained by a linear combination of the rest of the columns. Furthermore, since $L(\mathcal{G})$ is a symmetric matrix, the same occurs with the kth row. Therefore, the submatrix $L^k(\mathcal{G})$ is nonsingular since its $n-1$ columns (resp. rows) are linearly independent.

Now, let us prove that $(L^k(\mathcal{G}))^{-1}\mathbf{l}_k^{k_r} = -\mathbb{1}$. To do so, we use the fact that $(L^k(\mathcal{G}))^{-1}L^k(\mathcal{G}) = I$, where I is the identity matrix. Hence, by the definition of matrix multiplication, we have that

$$\sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mj}^k = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \tag{2}$$

where l_{ij}^k and $\bar{l}_{ij}^k$ are the elements located in the ith row and jth column of the matrices $L(\mathcal{G})$ and $\left(L^k(\mathcal{G})\right)^{-1}$, respectively. Thus,

$$\sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mi}^k = 1, \text{ for all } i = 1, \ldots n-1. \tag{3}$$

Let $l_{k_m}^{k_r}$ be the mth entry of the vector $\mathbf{l}_k^{k_r}$. Notice that, according to the definition of $L^k(\mathcal{G})$ and since $L(\mathcal{G})\mathbb{1} = \mathbf{0}$, $l_{mi}^k = -\sum_{j=1, j\neq i}^{n-1} l_{mj}^k - l_{k_m}^{k_r}$. Replacing this value in Equation (3), we obtain

$$-\sum_{j=1, j\neq i}^{n-1} \sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mj}^k - \sum_{m=1}^{n-1} \bar{l}_{im}^k l_{k_m}^{k_r} = 1, \text{ for all } i = 1, \ldots n-1.$$

According to Equation (2), $\sum_{j=1, j\neq i}^{n-1} \sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mj}^k = 0$. This implies that $\sum_{m=1}^{n-1} \bar{l}_{im}^k l_{k_m}^{k_r} = -1$, for all $i = 1, \ldots, n-1$. Therefore, $\left(L^k(\mathcal{G})\right)^{-1} \mathbf{l}_k^{k_r} = -\mathbb{1}$. ■

Theorem 2.1 and Lemma 2.1 will be used in the analysis of the method proposed in this paper.

3. Problem statement

In general terms, a resource allocation problem can be formulated as follows (Patriksson, 2008; Patriksson & Strömberg, 2015):

$$\min_x \phi(x) := \sum_{i=1}^n \phi_i(x_i) \tag{4a}$$

$$\text{subject to} \sum_{i=1}^n x_i = X \tag{4b}$$

$$x_i \geq \underline{x}_i, \text{ for all } i = 1, \ldots, n, \tag{4c}$$

where $x_i \in \mathbb{R}$ is the resource allocated to the ith zone; $x = [x_1, \ldots, x_n]^\top$; $\phi_i : \mathbb{R} \mapsto \mathbb{R}$ is a strictly convex and differentiable cost function; X is the available resource; and $\underline{x}_i$, is the lower bound of x_i, i.e. the minimum amount of resource that has to be allocated in the ith zone.

Given the fact that we are interested in distributed algorithms to solve the problem stated in Equation (4), we consider a multi-agent network, where the ith agent is responsible for managing the resource allocated to the ith zone. Moreover, we assume that the agents have limited communication capabilities, so they can only share information with their neighbours. This constraint can be represented by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as it was explained in Section 2.

Avoiding the individual inequality constraints (4c), KKT conditions establish that at the optimal solution $x^* = [x_1^*, \ldots, x_n^*]^\top$ of the problem given in Equation (4a–4b), the marginal costs $\phi_i'(x_i) = \frac{d\phi_i}{dx_i}$ must be equal, i.e. $\phi_i'(x_i^*) = \lambda$, for all $i = 1, \ldots, n$, where $\lambda \in \mathbb{R}$. Hence, a valid alternative to solve (4a–4b) is the use of consensus methods. For instance, we can adapt the algorithm presented in Xiao and Boyd (2006), which is described as follows:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \left(\phi_j'(x_j) - \phi_i'(x_i) \right), \text{ for all } i \in \mathcal{V}. \tag{5}$$

This algorithm has two main properties: (i) at equilibrium, $\phi_i'(x_i^*) = \phi_j'(x_j^*)$ if the nodes i and j are connected by a path; (ii) $\sum_{i=1}^n x_i^* = \sum_{i=1}^n x_i(0)$, where $x_i(0)$ is the initial condition of x_i. Therefore, if the graph $\mathcal{G}$ is connected and the initial condition is feasible (i.e. $\sum_{i=1}^n x_i(0) = X$), x asymptotically reaches the optimal solution of (4a–4b) under (5). However, the same method cannot be applied to solve (4) (the problem that considers lower bounds in the resource allocated to each zone) since some feasibility issues related with the constraints (4c) arise.

In the following section, we propose a novel method that extends the algorithm in Equation (5) to deal with the individual inequality constraints given in Equation (4c).

4. Centre-free resource allocation algorithm

4.1 Resource allocation among a subset of nodes in a graph

First, we consider the following subproblem: let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph comprised by a subset of active nodes $\mathcal{V}_a$ and a subset of passive nodes $\mathcal{V}_p$, such that $\mathcal{V}_a \bigcup \mathcal{V}_p = \mathcal{V}$. A certain amount of resource X has to be split among those nodes to minimise the cost function $\phi(x)$ subject to each passive node is allocated with its corresponding lower bound $\underline{x}_i$. Mathematically, we formulate this subproblem as:

$$\min_x \phi(x) \tag{6a}$$

$$\text{subject to } \sum_{i=1}^n x_i = X \tag{6b}$$

$$x_i = \underline{x}_i, \text{ for all } i \in \mathcal{V}_p. \tag{6c}$$

Feasibility of (6) is guaranteed by making the following assumption.

Assumption 4.1: *At least one node is active, i.e. $\mathcal{V}_a \neq \emptyset$.*

According to KKT conditions, the active nodes have to equalise their marginal costs at the optimal solution. Therefore, a consensus among the active nodes is required to solve (6). Nonetheless, classic consensus algorithms, as the one given in Equation (5), cannot be used directly. For instance, if all the nodes of $\mathcal{G}$ apply (5) and $\mathcal{G}$ is connected, the marginal costs of both passive and active nodes are driven to be equal in steady state. This implies that the resource allocated to passive nodes can violate the constraint (6c). Besides, if the resource allocated to passive nodes is forced to satisfy (6c) by setting $x_i^* = \underline{x}_i$, for all $i \in \mathcal{V}_p$, there is no guarantee that the new solution satisfies (6b). Another alternative, is to apply (5) to only active nodes (in this case, the neighbourhood of node $i \in \mathcal{V}_a$ in Equation (5) has to be taken as $\{j \in \mathcal{V}_a : (i, j) \in \mathcal{E}\}$, and the initial condition must satisfy $\sum_{i \in \mathcal{V}_a} x_i(0) = X - \sum_{i \in \mathcal{V}_p} \underline{x}_i$). However, the sub-graph formed by the active nodes is not necessarily connected although $\mathcal{G}$ is connected. Hence, marginal cost of active nodes are not necessarily equalised at equilibrium, which implies that the obtained solution is sub-optimal. In conclusion, modification of (5) to address (6) is not trivial. In order to deal with this problem, we propose the following algorithm:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} (y_j - y_i), \text{ for all } i \in \mathcal{V} \tag{7a}$$

$$\dot{\hat{x}}_i = (x_i - \underline{x}_i) + \sum_{j \in \mathcal{N}_i} (y_j - y_i), \text{ for all } i \in \mathcal{V}_p \tag{7b}$$

$$y_i = \begin{cases} \phi_i'(x_i) & \text{if } i \in \mathcal{V}_a \\ \phi_i'(x_i) + \hat{x}_i & \text{if } i \in \mathcal{V}_p. \end{cases} \tag{7c}$$

In the same way as in (5), the variables $\{x_i, i \in \mathcal{V}\}$ in Equation (7) correspond to the resource allocated to both active and passive nodes. Notice that we have added auxiliary variables $\{\hat{x}_i, i \in \mathcal{V}_p\}$ that allow the passive nodes to interact with their neighbours taking into account the constraint (6c). On the other hand, the term $\sum_{j \in \mathcal{N}_i} (y_j - y_i)$, in Equations (7a)–(7b), leads to a consensus among the elements of the vector $y = [y_1, \ldots, y_n]^\top$, which are given in Equation (7c). For active nodes, y_i only depends on the marginal cost $\phi_i'(x_i)$, while for passive nodes, y_i depends on both the marginal cost and the state of the auxiliary variable $\hat{x}_i$. Therefore, if the ith node is passive, it has to compute both variables x_i and $\hat{x}_i$. Furthermore, it can be seen that, if all the nodes are active, i.e. ($\mathcal{V}_a = \mathcal{V}$), then the proposed algorithm becomes the one stated in Equation (5).

Notice that the ith node only needs to know y_i and the values $\{y_j : j \in \mathcal{N}_i\}$ to compute $\sum_{j\in\mathcal{N}_i}(y_j - y_i)$ in (7a)–(7b). In other words, $L(\mathcal{G})y = [\sum_{j\in\mathcal{N}_1}(y_j - y_1), \ldots, \sum_{j\in\mathcal{N}_n}(y_j - y_n)]^\top$ is a distributed map over the graph $\mathcal{G}$ (Cortés, 2008). This implies that the dynamics given in Equation (7) can be computed by each node using only local information. In fact, the message that the ith node must send to its neighbours is solely composed by the variable y_i.

4.1.1 Feasibility

Let us prove that, under the multi-agent system proposed in Equation (7), $x(t)$ satisfies the first constraint of the problem given by Equation (6), for all $t \geq 0$, provided that $\sum_{i=1}^n x_i(0) = X$.

Lemma 4.1: *The quantity $\sum_{i=1}^n x_i(t)$ is invariant under Equation (7), i.e. if $\sum_{i=1}^n x_i(0) = X$, then $\sum_{i=1}^n x_i(t) = X$, for all $t \geq 0$.*

Proof: It is sufficient to prove that $\dot{\Delta} = 0$, where $\Delta = \sum_{i=1}^n x_i$. Notice that $\dot{\Delta} = \sum_{i=1}^n \dot{x}_i = \mathbb{1}^\top \dot{x}$, where $\dot{x} = [\dot{x}_1, \ldots, \dot{x}_n]^\top$. Moreover, according to Equation (7), $\mathbb{1}^\top \dot{x} = -\mathbb{1}^\top L(\mathcal{G})y$. Since $\mathcal{G}$ is undirected, $\mathbb{1}^\top L(\mathcal{G}) = L(\mathcal{G})\mathbb{1} = 0$. Therefore, $\dot{\Delta} = 0$. ■

The above lemma does not guarantee that $x(t)$ is always feasible because of the second constraint in Equation (6), i.e. $x_i = \underline{x}_i$, for all $i \in \mathcal{V}_p$. However, it is possible to prove that, at equilibrium, this constraint is properly satisfied.

4.1.2 Equilibrium point

The next proposition characterises the equilibrium point of the multi-agent system given in Equation (7).

Proposition 4.1: *If $\mathcal{G}$ is connected, the system in Equation (7) has an equilibrium point x^*, $\{\hat{x}_i^*, i \in \mathcal{V}_p\}$, such that: $\phi_i'(x_i^*) = \lambda$, for all $i \in \mathcal{V}_a$, where $\lambda \in \mathbb{R}$ is a constant; and $x_i^* = \underline{x}_i$, for all $i \in \mathcal{V}_p$. Moreover, $\hat{x}_i^* = \lambda - \phi_i'(x_i^*)$, for all $i \in \mathcal{V}_p$.*

Proof: Let x^*, $\{\hat{x}_i^*, i \in \mathcal{V}_p\}$ be the equilibrium point of Equation (7). Since $\mathcal{G}$ is connected by assumption, it follows from Equation (7a) that $y_i^* = \lambda$, for all $i \in \mathcal{V}$, where λ is a constant. Thus, $y_i^* = \phi_i'(x_i^*)$ if $i \in \mathcal{V}_a$, and $y_i^* = \phi_i'(x_i^*) + \hat{x}_i^*$, if $i \in \mathcal{V}_p$. Hence, $\phi_i'(x_i^*) = \lambda$, for all $i \in \mathcal{V}_a$, and $\hat{x}_i^* = \lambda - \phi_i'(x_i^*)$, for all $i \in \mathcal{V}_p$. Moreover, given the fact that $\sum_{j\in\mathcal{N}_i}(y_j^* - y_i^*) = 0$, it follows from Equation (7b) that $x_i^* = \underline{x}_i$, for all $i \in \mathcal{V}_p$ ■

Remark 4.1: Proposition 4.1 states that, at the equilibrium point of (7), the active nodes equalise their marginal costs, while each passive node is allocated with an amount of resource equal to its corresponding lower bound. In conclusion, if $\sum_{i=1}^n x_i^* = X$, then it follows from Proposition 4.1, that x^* minimises the optimisation problem given in Equation (6). Additionally, notice that the values $\{\hat{x}_i^*, i \in \mathcal{V}_p\}$ are equal to the KKT multipliers associated with the constraint (6c).

4.1.3 Convergence

Let us prove that the dynamics in Equation (7) converge to x^*, $\{\hat{x}_i^*, i \in \mathcal{V}_p\}$, provided that each $\phi_i(x_i)$ is strictly convex.

Proposition 4.2: *Assume that $\phi_i(x_i)$ is a strictly convex cost function, for all $i \in \mathcal{V}$. If $\mathcal{G}$ is connected, $\sum_{i=1}^n x_i(0) = X$, and Assumption 4.1 holds, then $x(t)$ converges to x^* under Equation (7), where x^* is the solution of the optimisation problem stated in Equation (6), i.e. x^* is the same given in Proposition 4.1. Furthermore, $\hat{x}_i$ converges to $\hat{x}_i^*$, for all $i \in \mathcal{V}_p$.*

Proof: According to Lemma 4.1, since $\sum_{i=1}^n x_i(0) = X$, then $x(t)$ satisfies the first constraint of the problem stated in Equation (6), for all $t \geq 0$. Therefore, it is sufficient to prove that the equilibrium point x^*, $\{\hat{x}_i^*, i \in \mathcal{V}_p\}$ (which is given in Proposition 4.1) of the system proposed in Equation (7) is asymptotically stable (AS). In order to do that, let us express our multi-agent system in error coordinates, as follows:

$$
\begin{aligned}
\dot{e} &= -L(\mathcal{G})e_y \\
\dot{\hat{e}}_i &= e_i - \big(L(\mathcal{G})e_y\big)_i, \qquad \text{for all } i \in \mathcal{V}_p \\
e_{y_i} &= \begin{cases} \phi_i'(x_i) - \phi_i'(x_i^*) & \text{if } i \in \mathcal{V}_a \\ \phi_i'(x_i) - \phi_i'(x_i^*) + \hat{e}_i & \text{if } i \in \mathcal{V}_p, \end{cases}
\end{aligned} \tag{8}
$$

where $L(\mathcal{G})$ is the graph Laplacian of $\mathcal{G}$; $e_i = x_i - x_i^*$, and $e_{y_i} = y_i - y_i^*$, for all $i \in \mathcal{V}$; $\hat{e}_i = \hat{x}_i - \hat{x}_i^*$, for all $i \in \mathcal{V}_p$; $e = [e_1, \ldots, e_n]^\top$; $e_y = [e_{y_1}, \ldots, e_{y_n}]^\top$; and $\big(L(\mathcal{G})e_y\big)_i$ represents the ith element of the vector $L(\mathcal{G})e_y$.

Since Assumption 4.1 holds, $\mathcal{V}_a \neq \emptyset$. Let k be an active node, i.e. $k \in \mathcal{V}_a$, and let e^k, e_y^k be the vectors obtained by removing the kth element from vectors e and e_y, respectively. We notice that, according to Lemma 4.1, $e_k(t) = -\sum_{i\in\mathcal{V}, i\neq k} e_i(t)$, for all $t \geq 0$. Therefore, Equation (8) can be expressed as

$$
\begin{aligned}
\dot{e}^k &= -L^k(\mathcal{G})e_y^k - \mathbf{l}_k^{k_r} e_{y_k} \\
e_k &= -\textstyle\sum_{i\in\mathcal{V}, i\neq k} e_i \\
\dot{\hat{e}}_i &= e_i - \Big(L^k(\mathcal{G})e_y^k + \mathbf{l}_k^{k_r} e_{y_k}\Big)_i, \quad \text{for all } i \in \mathcal{V}_p \\
e_{y_i} &= \begin{cases} \phi_i'(x_i) - \phi_i'(x_i^*) & \text{if } i \in \mathcal{V}_a \\ \phi_i'(x_i) - \phi_i'(x_i^*) + \hat{e}_i & \text{if } i \in \mathcal{V}_p, \end{cases}
\end{aligned} \tag{9}
$$

where $L^k(\mathcal{G})$ and $\mathbf{l}_k^{k_r}$ are defined in Lemma 2.1. In order to prove that the origin of the above system is AS, let us define the following Lyapunov function (adapted from

Obando, Quijano, & Rakoto-Ravalontsalama, 2014):

$$V = \frac{1}{2} e^{k\top} \left(L^k(\mathcal{G})\right)^{-1} e^k + \frac{1}{2} \sum_{i \in \mathcal{V}_p} (e_i - \hat{e}_i)^2. \quad (10)$$

The function V is positive definite since $\mathcal{G}$ is connected (the reason of this fact is that, according to Lemma 2.1, $L^k(\mathcal{G})$ and its inverse are positive definite matrices if $\mathcal{G}$ is connected). The derivative of V along the trajectories of the system stated in Equation (9) is given by,

$$\dot{V} = -e^{k\top} e_y^k - e^{k\top} (L(\mathcal{G}))^{-1} \mathbf{l}_k^{k_r} e_{y_k} - \sum_{i \in \mathcal{V}_p} e_i(e_i - \hat{e}_i)$$

Taking into account that $\left(L^k(\mathcal{G})\right)^{-1} \mathbf{l}_k^{k_r} = -\mathbb{1}$ (cf. Lemma 2.1), we obtain

$$\begin{aligned}\dot{V} &= -e^{k\top} e_y^k + e_{y_k} \sum_{i \in \mathcal{V}, i \neq k} e_i - \sum_{i \in \mathcal{V}_p} e_i(e_i - \hat{e}_i) \\ &= -\sum_{i=1}^{n} \left(e_i(\phi_i'(x_i) - \phi_i'(x_i^*))\right) - \sum_{i \in \mathcal{V}_p} e_i \hat{e}_i \\ &\quad + \sum_{i \in \mathcal{V}_p} e_i(\hat{e}_i - e_i) \\ &= -\sum_{i=1}^{n} \left((x_i - x_i^*)(\phi_i'(x_i) - \phi_i'(x_i^*))\right) - \sum_{i \in \mathcal{V}_p} e_i^2,\end{aligned}$$

where ϕ_i' is strictly increasing given the fact that ϕ_i is strictly convex, for all $i \in \mathcal{V}$. Therefore, $(x_i - x_i^*)(\phi_i'(x_i) - \phi_i'(x_i^*)) \geq 0$, for all $i \in \mathcal{V}$, and thus $\dot{V} \leq 0$.

Since $\dot{V}$ does not depend on $\{\hat{e}_i, i \in \mathcal{V}_p\}$, it is negative semidefinite. Let $S = \{\{e_i, i \in \mathcal{V}\}, \{\hat{e}_i, i \in \mathcal{V}_p\} : \dot{V} = 0\}$, i.e. $S = \left\{\{e_i, i \in \mathcal{V}\}, \{\hat{e}_i, i \in \mathcal{V}_p\} : e_i = 0, \text{for all } i \in \mathcal{V}\right\}$. Given the fact that $\mathcal{G}$ is connected and $\mathcal{V} \neq \mathcal{V}_p$ (by Assumption 4.1), then $\dot{e} = \mathbf{0}$ iff $e_y = \mathbf{0}$ (see Equation (8)). Therefore, the only solution that stays identically in S is the trivial solution, i.e. $e_i(t) = 0$, for all $i \in \mathcal{V}$, $\hat{e}_i(t) = 0$, for all $i \in \mathcal{V}_p$. Hence, we can conclude that the origin is AS by applying the Lasalle's invariance principle. ■

In summary, we have shown that the algorithm described in Equation (7) asymptotically solves the subproblem in Equation (6), i.e. (7) guarantees that the resource allocated to each passive node is equal to its corresponding lower bound, while the remaining resource $X - \sum_{i \in \mathcal{V}_p} \underline{x}_i$ is optimally allocated to active nodes.

4.2 Optimal resource allocation with lower bounds

Now, let us consider our original problem stated in Equation (4), i.e. the resource allocation problem that includes lower bound constraints. Let $x^* = [x_1^*, \ldots, x_n^*]^\top$ be the optimal solution of this problem. Notice that, if we know in advance which nodes will satisfy the constraint (4c) with strict equality after making the optimal resource allocation process, i.e. $\underline{\mathcal{I}} := \{i \in \mathcal{V} : x_i^* = \underline{x}_i\}$, we can mark these nodes as passive and reformulate (4) as a subproblem of the form (6). Based on this idea, we propose a solution method for (4), which is divided in two stages: in the first one, the nodes that belong to $\underline{\mathcal{I}}$ are identified and marked as passive; in the second one, the resulting subproblem of the form (6) is solved by using (7).

Protocol (7) can be also used in the first stage of the method as follows: in order to identify the nodes that will satisfy (4c) with strict equality at the optimal allocation, we start marking all nodes as active and apply the resource allocation process given by (7). The nodes that are allocated with an amount of resource below their lower bounds at equilibrium are marked as passive, and then (7) is newly applied (in this way, passive nodes are forced to meet (4c)). This iterative process is performed until all nodes satisfy their lower bound constraints. Notice that the last iteration of this procedure corresponds to solve a subproblem of the form (6) where the set of passive nodes is equal to the set $\underline{\mathcal{I}}$. Therefore, this last iteration is equivalent to the second stage of the proposed method.

Summarising, our method relies on an iterative process that uses the continuous-time protocol (7) as a subroutine. The main idea of this methodology is to identify in each step the nodes that have an allocated resource out of their lower bounds. These nodes are marked as passive, so they are forced to satisfy their constraints in subsequent iterations, while active nodes seek to equalise their marginal costs using the remaining resource. In the worst case scenario, the classification between active and passive nodes requires $|\mathcal{V}|$ iterations, where $|\mathcal{V}|$ is the number of nodes in the network. This fact arises when only one active node becomes passive at each iteration.

The proposed method is formally described in Algorithm 1. Notice that this algorithm is fully decentralised since Steps 4–6 can be computed by each agent using only local information. Step 4 corresponds to solve Equation (7), while Steps 5 and 6 describe the conditions for converting an active node into passive. Let us note that Steps 4–6 have to be performed $|\mathcal{V}|$ times since we are considering the worst case scenario. Therefore, each agent needs to know the total number of nodes in the network. This requirement can be computed in a distributed way by using the method proposed in Garin and Schenato (2010, p. 90). We also notice the fact that the agents have to be synchronised (as usual in several distributed algorithms; Cortés, 2008; Garin & Schenato, 2010; Xiao & Boyd, 2006) in order to apply the Step 4 of Algorithm 1, i.e. all agents must start solving Equation (7) at the same time.

Algorithm 1: Resource allocation with lower bounds

Input: – Parameters of the problem in Equation (4).
– An initial value $x^{(0)}$, such that $\sum_{i=1}^{n} x_i^{(0)} = X$.

Output: Optimal allocation $\tilde{x}^*$

1 Mark all nodes as active, i.e. $\tilde{\mathcal{V}}_{a,0} \leftarrow \mathcal{V}$, $\tilde{\mathcal{V}}_{p,0} \leftarrow \emptyset$.;

2 $\tilde{x}_{i,0} \leftarrow x_i^{(0)}$, for all $i \in \mathcal{V}$.;

3 **for** $l \leftarrow 1$ ***to*** $|\mathcal{V}|$ **do**

4 $\quad$ $\tilde{x}_{i,l} \leftarrow x_i(t_l)$, for all $i \in \mathcal{V}$, where $x_i(t_l)$ is the solution of Equation (7a) at time t_l, with initial conditions $x(0) = [\tilde{x}_{1,l-1}, \ldots, \tilde{x}_{n,l-1}]^\top$, $\mathcal{V}_a = \tilde{\mathcal{V}}_{a,l-1}$, $\mathcal{V}_p = \tilde{\mathcal{V}}_{p,l-1}$, and $\{\hat{x}_i(0) = 0, \forall i \in \mathcal{V}_p\}$.;

5 $\quad$ $\tilde{\mathcal{V}}_{p,l} \leftarrow \tilde{\mathcal{V}}_{p,l-1} \bigcup \{i \in \tilde{\mathcal{V}}_{a,l-1} : \tilde{x}_i < \underline{x}_i\}$, and $\tilde{\mathcal{V}}_{a,l} \leftarrow \tilde{\mathcal{V}}_{a,l-1} \backslash \{i \in \tilde{\mathcal{V}}_{a,l-1} : \tilde{x}_i < \underline{x}_i\}$.;

6 $\quad$ $\tilde{x}^* \leftarrow [\tilde{x}_{1,l}, \ldots, \tilde{x}_{n,l}]^\top$.;

7 **return** $\tilde{x}^*$;

According to the reasoning described at the beginning of this subsection, we ideally require to know the steady-state solution of Equation (7) at each iteration of Algorithm 1 (since we need to identify which nodes are allocated with an amount of resource below their lower bounds in steady state). This implies that the time t_l in Step 4 of Algorithm 1 goes to infinity. Under this requirement, each iteration would demand infinite time and the algorithm would not be implementable. Hence, to relax the infinite time condition, we state the following assumption on the time t_l.

Assumption 4.2: *Let $x^*_{i,l}$ be the steady state of $x_i(t)$ under Equation (7), with initial conditions $x(0) = \tilde{x}_{i,l-1}$, $\mathcal{V}_a = \tilde{\mathcal{V}}_{a,l-1}$, $\mathcal{V}_p = \tilde{\mathcal{V}}_{p,l-1}$, and $\{\hat{x}_i(0) = 0, \forall i \in \mathcal{V}_p\}$[1]. For each $l = 1, \ldots, |\mathcal{V}| - 1$, the time t_l satisfies the following condition: $x_i(t_l) < \underline{x}_i$ if and only if $x^*_{i,l} < \underline{x}_i$, for all $i \in \mathcal{V}$.*

According to assumption 4.2, for the first $|\mathcal{V}| - 1$ iterations, we only need a solution of (7) that is close enough to the steady-state solution. We point out the fact that, if the conditions of Proposition 4.2 are met in the lth iteration of Algorithm 1, then $x_i(t)$ asymptotically converges to $x^*_{i,l}$, for all $i \in \mathcal{V}$, under Equation (7). Therefore, Assumption 4.2 is satisfied for large values of $t_1, \ldots, t_{|\mathcal{V}|-1}$.

Taking into account all the previous considerations, the next theorem states our main result regarding the optimality of the output of Algorithm 1.

Theorem 4.1: *Assume that $\mathcal{G}$ is a connected graph. Moreover, assume that ϕ_i is a strictly convex function for all $i = 1, \ldots, n$. If $t_1, \ldots, t_{|\mathcal{V}|-1}$ satisfy Assumption 4.2, and the problem stated in Equation (4) is feasible, then the output of Algorithm 4 tends to the optimal solution of the problem given in Equation (4) as $t_{|\mathcal{V}|} \to \infty$.*

Proof: The ith component of the output of Algorithm 1 is equal to $\tilde{x}_{i,|\mathcal{V}|} = x_i(t_{|\mathcal{V}|})$, where $x_i(t_{|\mathcal{V}|})$ is the solution of Equation (7a) at time $t_{|\mathcal{V}|}$, with initial conditions $[\tilde{x}_{1,|\mathcal{V}|-1}, \ldots, \tilde{x}_{n,|\mathcal{V}|-1}]^\top$, $\mathcal{V}_a = \tilde{\mathcal{V}}_{a,|\mathcal{V}|}$, and $\mathcal{V}_p = \tilde{\mathcal{V}}_{p,|\mathcal{V}|}$. Hence, it is sufficient to prove that $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}\}$ solves the problem in Equation (4). In order to do that, let us consider the following premises (proof of each premise is written in brackets).

P1: $\{\tilde{x}_{1,l}, \ldots, \tilde{x}_{n,l}\}$ satisfies (4b), for all $l = 1, \ldots, |\mathcal{V}|$ (this follows from Lemma 4.1, and form the fact that $\sum_{i=1}^{n} \tilde{x}_{i,0} = X$).

P2: $x^*_{i,l} = \underline{x}_i$, for all $i \in \tilde{\mathcal{V}}_{p,l-1}$, and for all $l = 1, \ldots, |\mathcal{V}|$ (this follows directly from Proposition 4.2).

P3: $\tilde{\mathcal{V}}_{p,l} = \tilde{\mathcal{V}}_{p,l-1} \bigcup \{i \in \tilde{\mathcal{V}}_{a,l-1} : x^*_{i,l} < \underline{x}_i\}$, and $\tilde{\mathcal{V}}_{a,l} = \tilde{\mathcal{V}}_{a,l-1} \backslash \{i \in \tilde{\mathcal{V}}_{a,l-1} : x^*_{i,l} < \underline{x}_i\}$, for all $l = 1, \ldots, |\mathcal{V}|$ (this follows from Step 5 of Algorithm 1, and from Assumption 4.2).

P4: If for some l, $\tilde{\mathcal{V}}_{p,l} = \tilde{\mathcal{V}}_{p,l-1}$, then $\tilde{\mathcal{V}}_{p,l+j} = \tilde{\mathcal{V}}_{p,l-1}$, for all $j = 0, \ldots, |\mathcal{V}| - l$ (this can be seen from the fact that if the set of passive nodes does not change from one iteration to the next, the steady state of Equation (7a) is the same for both iterations).

P5: $\tilde{\mathcal{V}}_{a,l} \bigcup \tilde{\mathcal{V}}_{p,l} = \mathcal{V}$, for all $l = 1, \ldots, |\mathcal{V}|$ (from P3, we know that $\tilde{\mathcal{V}}_{a,l} \bigcup \tilde{\mathcal{V}}_{p,l} = \tilde{\mathcal{V}}_{a,l-1} \bigcup \tilde{\mathcal{V}}_{p,l-1}$, for all $l = 1, \ldots, |\mathcal{V}|$. Moreover, given the fact that $\tilde{\mathcal{V}}_{p,0} = \emptyset$, and $\tilde{\mathcal{V}}_{a,0} = \mathcal{V}$, (see step 1 of Algorithm 1) we can conclude P5).

P6: Since the problem in Equation (4) is feasible by assumption, then $|\tilde{\mathcal{V}}_{p,l}| < |\mathcal{V}|$, for all $l = 1, \ldots, |\mathcal{V}|$ (the fact that $|\tilde{\mathcal{V}}_{p,l}| \leq |\mathcal{V}|$, for all $l = 1, \ldots, \mathcal{V}$, follows directly from P5. Let us prove that $|\tilde{\mathcal{V}}_{p,l}| \neq |\mathcal{V}|$, for all $l = 1, \ldots, \mathcal{V}$. We proceed by contradiction: Assume that there exists some l, such that $|\tilde{\mathcal{V}}_{p,l-1}| < |\mathcal{V}|$ and $|\tilde{\mathcal{V}}_{p,l}| = |\mathcal{V}|$. Hence, from P2 and P3, we know that $x^*_{i,l} \leq \underline{x}_i$, for all $i \in \mathcal{V}$; moreover, $\{i \in \tilde{\mathcal{V}}_{a,l-1} : x^*_{i,l} < \underline{x}_i\} \neq \emptyset$. Therefore, $\sum_{i=1}^{n} x^*_{i,l} < \sum_{i=1}^{n} \underline{x}_i$. According to P1, we know that $\sum_{i=1}^{n} x^*_{i,l} = X$; thus, $X < \sum_{i=1}^{n} \underline{x}_i$, which contradicts the feasibility assumption).

P7: $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}\}$ satisfies the constraints (4c) (in order to prove P7, we proceed by contradiction: assume that $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}\}$ does not satisfy the constraints (4c). Since P2 holds, this assumption implies that $\{i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}-1|} : x^*_{i,|\mathcal{V}|} < \underline{x}_i\} \neq \emptyset$. Therefore, $\tilde{\mathcal{V}}_{p,|\mathcal{V}|} \neq \tilde{\mathcal{V}}_{p,|\mathcal{V}|-1}$ (see P3). Using P4, we can conclude that $\tilde{\mathcal{V}}_{p,|\mathcal{V}|} \neq \tilde{\mathcal{V}}_{p,|\mathcal{V}|-1} \neq \cdots \neq \tilde{\mathcal{V}}_{p,0} = \emptyset$, i.e. $\{i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|-j} : x^*_{i,|\mathcal{V}|-j+1} < \underline{x}_i\} \neq \emptyset$, for all $j = 1, \ldots, |\mathcal{V}|$. Thus, according to P3, $|\tilde{\mathcal{V}}_{p,|\mathcal{V}|}| > |\tilde{\mathcal{V}}_{p,|\mathcal{V}|-1}| > \cdots > |\tilde{\mathcal{V}}_{p,1}| > 0$. Hence, $|\tilde{\mathcal{V}}_{p,|\mathcal{V}|}| \geq |\mathcal{V}|$, which contradicts P6).

P8: $\sum_{i \in \tilde{\mathcal{V}}_{a,l}} x^*_{i,l} \geq \sum_{i \in \tilde{\mathcal{V}}_{a,l}} x^*_{i,l+1}$ (we prove P8 as follows: using P1 and the result in Lemma

4.1, we know that $\sum_{i\in\mathcal{V}} x^*_{i,l} = \sum_{i\in\mathcal{V}} x^*_{i,l+1} = X$. Moreover, according to P5, $\mathcal{V}$ can be expressed as $\mathcal{V} = \tilde{\mathcal{V}}_{a,l} \bigcup \tilde{\mathcal{V}}_{p,l}$, where $\tilde{\mathcal{V}}_{p,l-1} \subset \tilde{\mathcal{V}}_{p,l}$ (see P3). Thus, we have that $\sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l} + \sum_{i\in\tilde{\mathcal{V}}_{p,l}, i\notin\tilde{\mathcal{V}}_{p,l-1}} x^*_{i,l} + \sum_{i\in\tilde{\mathcal{V}}_{p,l-1}} x^*_{i,l} = \sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l+1} + \sum_{i\in\tilde{\mathcal{V}}_{p,l}, i\notin\tilde{\mathcal{V}}_{p,l-1}} x^*_{i,l+1} + \sum_{i\in\tilde{\mathcal{V}}_{p,l-1}} x^*_{i,l+1}$. Furthermore, since P2 holds, we have that $\sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l} + \sum_{i\in\tilde{\mathcal{V}}_{p,l}, i\notin\tilde{\mathcal{V}}_{p,l-1}} x^*_{i,l} + \sum_{i\in\tilde{\mathcal{V}}_{p,l-1}} \underline{x}_i = \sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l+1} + \sum_{i\in\tilde{\mathcal{V}}_{p,l}, i\notin\tilde{\mathcal{V}}_{p,l-1}} \underline{x}_i + \sum_{i\in\tilde{\mathcal{V}}_{p,l-1}} \underline{x}_i$. Therefore, $\sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l} = \sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l+1} + \sum_{i\in\tilde{\mathcal{V}}_{p,l}, i\notin\tilde{\mathcal{V}}_{p,l-1}} (\underline{x}_i - x^*_{i,l})$, where $\underline{x}_i - x^*_{i,l} > 0$, for all $i \in \tilde{\mathcal{V}}_{p,l}$, $i \notin \tilde{\mathcal{V}}_{p,l-1}$ (according to P3). Hence, we can conclude P8).

P9: There exists k, such that $k \in \tilde{\mathcal{V}}_{a,l}$, for all $l = 1, \ldots, |\mathcal{V}|$ (in order to prove P9, we use the fact that, if $k \in \tilde{\mathcal{V}}_{a,l}$, then $k \in \tilde{\mathcal{V}}_{a,l-j}$, for all $j = 1, \ldots, l$ (this follows from P3). Moreover, according to P5 and P6, $|\tilde{\mathcal{V}}_{a,|\mathcal{V}|}| \neq 0$; hence, there exists k, such that $k \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|}$. Therefore, P9 holds). P9 guarantees that Assumption 4.1 is satisfied at each iteration.

P10: $\phi'_i(x^*_{i,l}) \geq \phi'_i(x^*_{i,l+1})$, for all $i \in \tilde{\mathcal{V}}_{a,l}$ (we prove P10 by contradiction: assume that $\phi'_i(x^*_{i,l}) < \phi'_i(x^*_{i,l+1})$, for some $i \in \tilde{\mathcal{V}}_{a,l}$. According to Proposition 4.2, and since P1 and P9 hold, $x^*_{i,l}$ has the characteristics given in Proposition 4.1, for all $i \in \mathcal{V}$, and for all $l = 1, \ldots, |\mathcal{V}|$. Hence, $\phi'_i(x^*_{i,l})$ has the same value for all $i \in \tilde{\mathcal{V}}_{a,l-1}$, and $\phi'_i(x^*_{i,l+1})$ has the same value for all $i \in \tilde{\mathcal{V}}_{a,l}$. Moreover, since $\tilde{\mathcal{V}}_{a,l} \subset \tilde{\mathcal{V}}_{a,l-1}$ (according to P3), we have that $\phi'_i(x^*_{i,l}) < \phi'_i(x^*_{i,l+1})$, for all $i \in \tilde{\mathcal{V}}_{a,l}$. Thus, $x^*_{i,l} < x^*_{i,l+1}$, for all $i \in \mathcal{V}_{a,l}$, because ϕ'_i is strictly increasing (this follows from the fact that ϕ_i is strictly convex by assumption). Therefore, $\sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l} < \sum_{i\in\tilde{\mathcal{V}}_{a,l}} x^*_{i,l+1}$, which contradicts P8).

Now, let us prove that $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}\}$ solves the Problem in Equation (4). First, the solution $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}\}$ is feasible according to P1 and P7. On the other hand, from P9, it is known that $\exists k : k \in \tilde{\mathcal{V}}_{a,l}$, for all $l = 1, \ldots, |\mathcal{V}|$. Let $\phi'_k(x^*_{k,|\mathcal{V}|}) = \lambda$, where $\lambda \in \mathbb{R}$. Moreover, let us define $\mathcal{V}^0 = \{j \in \mathcal{V} : x^*_{i,|\mathcal{V}|} > \underline{x}_i\}$, and $\mathcal{V}^1 = \{j \in \mathcal{V} : x^*_{i,|\mathcal{V}|} = \underline{x}_i\}$.

If $i \in \mathcal{V}^0$, then $i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|-1}$ (given the fact that, if $i \notin \tilde{\mathcal{V}}_{a,|\mathcal{V}-1|} \Rightarrow i \in \tilde{\mathcal{V}}_{p,|\mathcal{V}-1|} \Rightarrow x^*_{i,|\mathcal{V}|} = \underline{x}_i \Rightarrow i \notin \mathcal{V}^0$). Hence, $\phi'_i(x^*_{i,|\mathcal{V}|}) = \phi'_k(x^*_{k,|\mathcal{V}|}) = \lambda$ (this follows from the fact that $\phi'_j(x^*_{j,l})$ has the same value for all $j \in \tilde{\mathcal{V}}_{a,l-1}$, which in turn follows directly from step 4 of Algorithm 1, and Proposition 4.2).

If $i \in \mathcal{V}^1$, then either $i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|-1}$ or $i \in \tilde{\mathcal{V}}_{p,|\mathcal{V}|-1}$. In the first case, $\phi'_i(x^*_{i,|\mathcal{V}|}) = \phi'_k(x^*_{k,|\mathcal{V}|}) = \lambda$ (following the reasoning used when $i \in \mathcal{V}^0$). In the second case, $\exists l : i \in (\tilde{\mathcal{V}}_{p,l} \backslash \tilde{\mathcal{V}}_{p,l-1})$; hence, $\phi'_i(x^*_{i,l}) = \phi'_k(x^*_{k,l})$ (this follows from the fact that, if $i \in (\tilde{\mathcal{V}}_{p,l} \backslash \tilde{\mathcal{V}}_{p,l-1})$, then $i \in \tilde{\mathcal{V}}_{a,l-1}$). Furthermore, since $i \in (\tilde{\mathcal{V}}_{p,l} \backslash \tilde{\mathcal{V}}_{p,l-1})$, $x^*_{i,l} < \underline{x}_i$ (see P3), and given the fact that ϕ_i is strictly increasing, we have that $\phi'_i(x^*_{i,l}) < \phi'_i(\underline{x}_i)$. Moreover, according to P10, $\phi'_k(x^*_{k,l}) \geq \phi'_k(x^*_{k,|\mathcal{V}|})$. Hence, $\phi'_i(\underline{x}_i) > \phi'_k(x^*_{k,|\mathcal{V}|}) = \lambda$. In conclusion, if $i \in \mathcal{V}^1$, then $\phi'_i(x^*_{i,|\mathcal{V}|}) \geq \lambda$.

Thus, we can choose $\mu_i \geq 0$, for all $i \in \mathcal{V}$, such that $\phi'_i(x^*_{i,|\mathcal{V}|}) - \mu_i = \lambda$, where $\mu_i = 0$ if $i \in \mathcal{V}^0$. Hence, let us note that $\frac{\partial \phi}{\partial x_i}|_{x_i = x^*_{i,|\mathcal{V}|}} - \mu_i - \lambda = 0$, for all $i \in \mathcal{V}$, where $\frac{\partial \phi}{\partial x_i}|_{x_i = x^*_{i,|\mathcal{V}|}} = \phi'_i(x^*_{i,|\mathcal{V}|})$. Therefore, $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}, \mu_1, \ldots, \mu_n, -\lambda\}$ satisfies the KKT conditions for the problem given in Equation (4). Furthermore, since $\phi(x)$ is a strictly convex function by assumption, then $\{x^*_{1,|\mathcal{V}|}, \ldots, x^*_{n,|\mathcal{V}|}\}$ is the optimal solution to that problem. ■

Early stopping criterion

Notice that, if the set of passive nodes does not change in the kth iteration of Algorithm 1 because all active nodes satisfy the lower bound constraints (see step 5), then the steady state solutions $x^*_{i,k}$ and $x^*_{i,k+1}$ are the same, for all $i \in \mathcal{V}$, which implies that the set of passive nodes also does not change in the $(k+1)$th iteration. Following the same reasoning, we can conclude that $x^*_{i,k} = x^*_{i,k+1} = \cdots = x^*_{i,|\mathcal{V}|}$, for all $i \in \mathcal{V}$. Therefore, in this case, $\{x^*_{1,k}, \ldots, x^*_{n,k}\}$ is the solution of our resource allocation problem. Practically speaking, this implies that Algorithm 1 does not need to perform more iterations after the kth one. Thus, it is possible to implement a flag z^*_i (in a distributed way) that alerts the agents if all active nodes satisfy the lower bound constraints after step 4 of Algorithm 1. A way to do that is by applying a *min–consensus* protocol (Cortés, 2008) with initial conditions $z_i(0) = 0$ if the node i is active and does not satisfy its lower bound constraint, and $z_i(0) = 1$ otherwise. Hence, notice that our flag z^*_i (i.e. the result of the *min-consensus* protocol) is equal to one, for all $i \in \mathcal{V}$, only if all the active nodes satisfy the lower bound constraints, which corresponds to the early stopping criterion described above.

5. Simulation results and comparison

In this section, we compare the performance of our algorithm with other continuous-time distributed techniques found in the literature. We have selected three techniques that are capable to address nonlinear problems and can handle lower bound constraints: (i) a distributed interior point method (Xiao & Boyd, 2006), (ii) the local replicator equation (Pantoja & Quijano, 2012), and (iii) a distributed interior point method with exact barrier functions (Cherukuri & Cortés, 2015). The first one is a traditional methodology that uses barrier functions; the second one is a novel technique based on population dynamics; and the third one is a recently proposed method that follows the same ideas as the first one, but replaces classic logarithmic barrier functions by exact penalty functions. Below, we briefly describe the aforementioned algorithms.

5.1 Distributed interior point (DIP) method

This algorithm is a variation of the one presented in Equation (5) that includes strictly convex barrier functions to prevent the solution to flow outside the feasible region. The barrier functions $b_i(x_i)$ are added to the original cost function as follows:

$$\phi_b(x) = \phi(x) + \epsilon \sum_{i=1}^{n} b_i(x_i)$$
$$b_i(x_i) = -\ln\left(x_i - \underline{x}_i\right), \qquad \text{for all } i \in \mathcal{V},$$

where $\phi_b(x)$ is the new cost function, and $\epsilon > 0$ is a constant that minimises the effect of the barrier function when the solution is far from the boundary of the feasible set. With this modification, the distributed algorithm is described by the following equation:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \left(\phi'_{b_j}(x_j) - \phi'_{b_i}(x_i)\right), \text{ for all } i \in \mathcal{V}, \tag{11}$$

where $\phi'_{b_i}(x_i) = \frac{d\phi_i}{dx_i} - \epsilon \frac{db_i}{dx_i}$, i.e. $\phi'_{b_i}(x_i)$ is equal to the marginal cost plus a penalty term induced by the derivative of the corresponding barrier function.

5.2 Local replicator equation (LRE)

This methodology is based on the classical replicator dynamics from evolutionary game theory. In the LRE, the growth rate of a population that plays a certain strategy only depends on its own fitness function and on the fitness of its neighbours. Mathematically, the LRE is given by

$$\begin{aligned} \dot{x}_i &= \sum_{j \in \mathcal{N}_i} (x_i - \underline{x}_i)(x_j - \underline{x}_j)(v_i(x_i) - v_j(x_j)), \\ v_i &= -\phi'_i(x_i), \text{ for all } i \in \mathcal{V}, \end{aligned} \tag{12}$$

where v_i is the fitness perceived by the individuals that play the ith strategy. In this case, the strategies correspond to the nodes of the network, and the fitness functions to the negative marginal costs (the minus appears because replicator dynamics are used to maximise utilities instead of minimise costs). On the other hand, it can be shown that, if the initial condition $x(0)$ is feasible for the problem given in Equation (4), then $x(t)$ remains feasible for all $t \geq 0$, under the LRE.

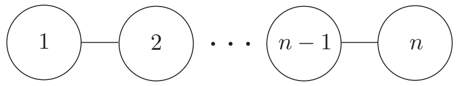


Figure 1. Single path topology for n nodes.

5.3 Distributed interior point method with exact barrier functions (DIPe)

This technique follows the same reasoning of the DIP algorithm. The difference is that DIPe uses exact barrier functions (Bertsekas, 1975) to guarantee satisfaction of the lower bound constraints. The exact barrier function for the ith node is given by:

$$b_i^e(x_i) = \frac{1}{\varepsilon}[\underline{x}_i - x_i]_+,$$

where $[\cdot]_+ = \max(\ \cdot, 0)$, $0 < \varepsilon < \frac{1}{2\max_{x\in\mathcal{F}} \|\nabla\phi(x)\|_\infty}$, and $\mathcal{F} = \{x \in \mathbb{R}^n : \sum_{i=1}^{n} x_i = 1, x_i \geq \underline{x}_i\}$ is the feasible region of x for the problem (4). Using these exact barrier functions, the augmented cost function can be expressed as:

$$\phi_b^e(x) = \phi(x) + \sum_{i=1}^{n} b_i^e(x_i).$$

The DIPe algorithm is given in terms of the augmented cost function and its generalised gradient $\partial\phi_b^e(x) = [\partial_1\phi_b^e(x), \ldots, \partial_n\phi_b^e(x)]^\top$ as follows:

$$\dot{x}_i \in \sum_{j \in \mathcal{N}_i} \left(\partial_j\phi_b^e(x) - \partial_i\phi_b^e(x)\right), \text{ for all } i \in \mathcal{V}, \tag{13}$$

where

$$\partial_i\phi_b^e(x) = \begin{cases} \{\phi'_i(x_i) - \frac{1}{\varepsilon}\} & \text{if } x_i < \underline{x}_i \\ \left[\phi'_i(x_i) - \frac{1}{\varepsilon}, \phi'_i(x_i)\right] & \text{if } x_i = \underline{x}_i \\ \{\phi'_i(x_i)\} & \text{if } x_i > \underline{x}_i \end{cases}$$

In Cherukuri and Cortés (2015), the authors show that the differential inclusion (13) converges to the optimal solution of the problem (4), provided that $x(0)$ is feasible.

5.4 Comparison

In order to compare the performance of our algorithm with the three methods described above, we use the following simulation scenario: a set of n nodes connected as in Figure 1 (we use this topology to verify the behaviour of the different algorithms in the face of few communication channels since previous studies have shown that algorithms' performance decreases with the number of

Table 1. Distributed algorithms' performance.

	Percentage decrease, computation time			
Number of nodes	Proposed approach	DIP	LRE	DIPe
$n = 5$	100%, 0.08 s	100%, 0.04 s	91%, 0.04 s	100%, 7 s
$n = 20$	100%, 0.7 s	99%, 0.4 s	69%, 0.2 s	100%, 153 s
$n = 50$	100%, 3.2 s	98%, 1.6 s	57%, 1.3 s	100%, 841 s
$n = 100$	100%, 17.8 s	96%, 9.1 s	50%, 6.5 s	–
$n = 200$	100%, 181.2 s	94%, 68.7 s	46%, 41.6 s	–

available communication links); a nonlinear cost function $\phi(x) = \sum_{i=1}^{n} e^{a_i(x_i - b_i)} + e^{-a_i(x_i - b_i)}$, where a_i and b_i are random numbers that belong to the intervals [1, 2] and $[-\frac{1}{2}, \frac{1}{2}]$, respectively; a resource constraint $X = 1$; and a set of lower bounds $\{\underline{x}_i = 0 : i \in \mathcal{V}\}$.

For each n, we generate 50 problems with the characteristics described above. The four distributed methods are implemented in Matlab employing the solver function ode23s. Moreover, we use the solution provided by a centralised technique as reference. The results on the average percentage decrease in the cost function reached with each algorithm and the average computation time (time taken by each algorithm for solving a problem[2]) are summarised in Table 1. Results of DIPe for 100 and 200 nodes were not computed for practicality since the time required by this algorithm to solve a 100/200-nodes problem is very high.

We notice that the algorithm proposed in this paper always reaches the maximum reduction, regardless of the number of nodes that comprise the network. The same happens with the DIPe algorithm. This is an important advantage of our method compared to other techniques. In contrast, the algorithm based on the LRE performs far from the optimal solution. This unsatisfactory behaviour is due to the small number of links of the considered communication network. In Pantoja and Quijano (2012), the authors prove the optimality of the LRE in problems involving well connected networks; however, they also argue that this technique can converge to suboptimal solutions in other cases. On the other hand, the DIP method provides solutions close to the optimum. Nonetheless, its performance decreases when the number of nodes increases. This tendency is due to the influence of barrier functions on the original problem. Notice that, the larger the number of nodes, the bigger the effect of the barrier functions in Equation (11).

Regarding the computation time, although convergence of the proposed method is slower than the one shown by LRE and DIP, it is faster than the convergence of the method based on exact barrier functions, i.e. DIPe. Therefore, among the methods that guarantee optimality of the solution, our technique shows the best convergence speed. Computation time taken by DIPe is affected by the use of penalty terms that generate strong changes in the value of the cost function near to the boundaries of the feasible set. The drastic variations of the generalised gradient of exact barrier functions produces oscillations of numerical solvers around the lower bounds (a visual inspection of the results given in Figure 3 of Cherukuri and Cortés (2015) confirms this claim). These oscillations are the main responsible for the low convergence speed shown by DIPe. On the other hand, LRE and DIP exhibit the fastest convergence. Hence, LRE and DIP are appealing to be implemented in applications that require fast computation and tolerate suboptimal solutions.

6. Applications

This section describes the use of the approach developed in this paper to solve two engineering problems. First, we present an application for sharing load in multiple chillers plants. Although this is not a large-scale application (multi-chiller plants are typically comprised of less than ten chillers; Yu & Chan, 2007), it aims to illustrate the essence of the proposed method and shows algorithm's performance in small-size problems. One of the reasons to use a distributed approach in small-/medium-size systems is due to the need of enhancing systems resilience in the face of central failures (e.g. in multiple chiller plants, central failures can occur due to cyber-attacks (Manic, Wijayasekara, Amarasinghe, & Rodriguez-Andina, 2016) against building management systems (Yu & Chan, 2007)). The second application deals with the distributed computation of the Euclidean projection of a vector onto a given set. Particularly, we use the proposed algorithm as part of a distributed technique that computes optimal control inputs for plants composed of a large number of sub-systems. This application aims to illustrate the performance of the method proposed in this paper when coping with large-scale problems.

6.1 Optimal chiller loading

The optimal chiller loading problem in multiple chiller systems arises in decoupled chilled–water plants, which

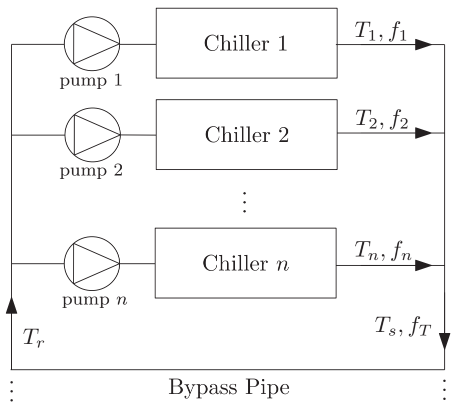


Figure 2. Decoupled chilled–water plant with n chillers.

are widely used in large air–conditioning systems (Chang & Chen, 2009). The goal is to distribute the cooling load among the chillers that comprise the plant for minimising the total amount of power used by them. For a better understanding of the problem, below we present a brief description of the system.

A decoupled chilled-water plant comprised by n chillers is depicted in Figure 2. The purpose of this plant is to provide a water flow f_T at a certain temperature T_s to the rest of the air-conditioning system. In order to do this task the plant needs to meet a cooling load C_L that is given by the following expression:

$$C_{\rm L} = m f_{\rm T}(T_{\rm r} - T_{\rm s}), \tag{14}$$

where $m > 0$ is the specific heat of the water, and $T_{\rm r}$ is the temperature of the water returning to the chillers. Since there are multiple chillers, the total cooling load $C_{\rm L}$ is split among them, i.e. $C_{\rm L} = \sum_{i=1}^{n} Q_i$, where Q_i is the cooling power provided by the ith chiller, which, in turn, is given by

$$Q_i = m f_i(T_{\rm r} - T_i), \tag{15}$$

where $f_i > 0$ and T_i are, respectively, the flow rate of chilled water and the water supply temperature of the ith chiller. As it is shown in Figure 2, we have that $f_{\rm T} = \sum_{i=1}^{n} f_i$. In order to meet the corresponding cooling load, the ith chiller consumes a power P_i that can be calculated using the following expression (Chang & Chen, 2009):

$$\begin{aligned} P_i = &\left(k_{0,i} + k_{1,i} m f_i T_r + k_{2,i}(m f_i T_r)^2 +\right) \\ &+ \left(k_{3,i} - k_{1,i} m f_i - k_{4,i} m f_i T_r - 2k_{2,i}(m f_i)^2 T_r\right) T_i \\ &+ \left(k_{5,i} + k_{6,i} m f_i + k_{2,i}(m f_i)^2\right) T_i^2, \end{aligned} \tag{16}$$

where $k_{j,\,i}$, for $j = 0, \ldots, 6$, are constants related to the ith chiller. If we assume that the flow rate f_i of each chiller is constant, then P_i is a quadratic function of the temperature T_i. The optimal chiller loading problem involves the calculation of the chillers' water supply temperatures that meet the total cooling load given in Equation (14), and minimise the total amount of power consumed by the chillers, i.e. $\sum_{i=1}^{n} P_i$. Moreover, given the fact that each chiller has a maximum cooling capacity, we have to consider the following additional constraints:

$$m f_i(T_r - T_i) \le \overline{Q}_i \quad \text{for all } i = 1, \ldots, n, \tag{17}$$

where $\overline{Q}_i$ is the maximum capacity (rated value) of the ith chiller.

Summarising, the optimal chiller loading problem can be expressed as follows:

$$\begin{aligned} \min_{T_1,\ldots,T_n} \quad & \textstyle\sum_{i=1}^{n} P_i(T_i) \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} m f_i(T_r - T_i) = C_{\rm L} \\ & T_i \ge T_r - \frac{\overline{Q}_i}{m f_i}, \quad \text{for all } i = 1, \ldots, n. \end{aligned} \tag{18}$$

Now, let us consider that we want to solve the aforementioned problem in a distributed way by using a multi-agent system, in which each chiller is managed by an agent that decides the value of the water supply temperature. We assume that the ith agent knows (e.g. by measurements) the temperature of the water returning to the chillers, i.e. T_r, and the flow rate of chilled water, i.e. f_i. Moreover, agents can share their own information with their neighbours through a communication network with a topology given by the graph $\mathcal{G}$. If each $P_i(T_i)$ is a convex function, then the problem can be solved by using the method proposed in Algorithm 1 (we take, in this case, $x_i = f_i T_i$). The main advantage of this approach is to increase the resilience of the whole system in the face of possible failures, due to the fact that the plant operation does not rely on a single control centre but on multiple individual controllers without the need for a centralised coordinator.

6.1.1 Illustrative example

We simulate a chilled-water plant comprised by 7 chillers.[3] The cooling capacity and the water flow rate of each chiller are, respectively, $\overline{Q}_i = 1406.8$ kW, and $f_i = 65$ kg.s^{-1}, for $i = 1, \ldots, 7$; the specific heat of the water is $m = 4.19$ kW.s.kg^{-1}. degC^{-1}; the supply temperature of the system is $T_s = 11$ degC; and the coefficients $k_{j,\,i}$ of Equation (16) are given in Table 2. We operate the system at two different cooling loads, the first one is 90% of the total capacity, i.e. $C_{\rm L} = 0.9 \sum_{i=1}^{n} \overline{Q}_i$, and the second one is 60% of the total capacity, i.e. $C_{\rm L} = 0.6 \sum_{i=1}^{n} \overline{Q}_i$. The P_i–T_i curves are shown in Figure 3(a) for both cases, it can be noticed that all functions are convex. In order to apply

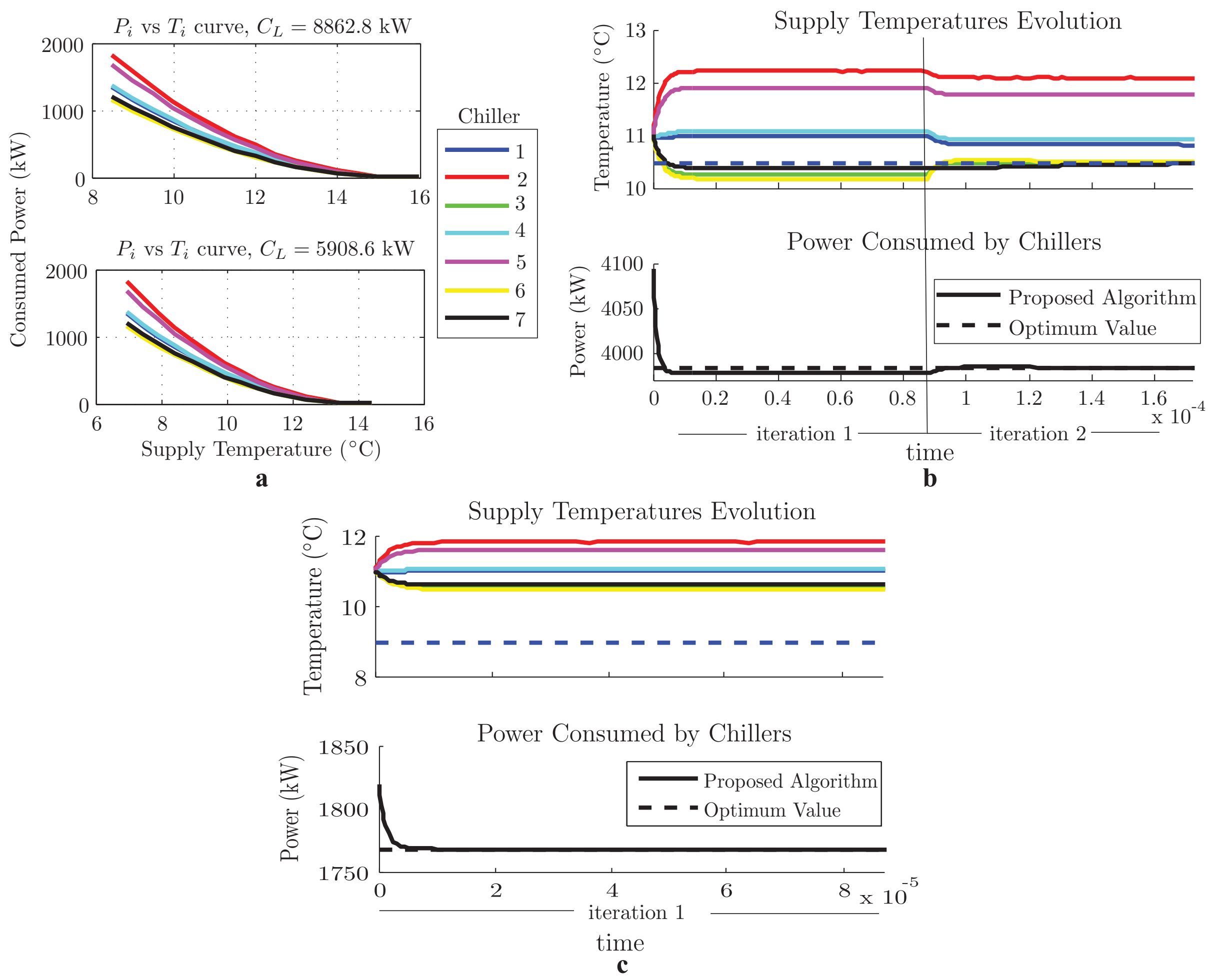


Figure 3. (a) P_i–T_i curves for each chiller, (b) evolution of supply temperatures and total power consumed by the chillers, $C_L = 8862.8$ kW, (c) evolution of supply temperatures and total power consumed by the chillers, $C_L = 5908.6$ kW.

Table 2. Chillers' parameters.

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$
$k_{0,i}$	113.51	71.70	62.75	112.68	74.13	61.98	76.54
$k_{1,i}$	0.21	−0.45	0.49	0.18	−0.44	0.55	0.34
$k_{2,i}$	0.35	0.48	0.30	0.36	0.44	0.30	0.31
$k_{3,i}$	−8.19	−5.13	−4.53	−8.13	−5.30	−4.48	−5.527
$k_{4,i}$	0.43	−0.14	0.71	0.40	−0.13	0.76	0.54
$k_{5,i}$	0.14	0.09	0.08	0.14	0.09	0.08	0.10
$k_{6,i}$	0.01	−0.01	0.02	0.01	−0.01	0.02	0.01

the distributed solution presented in Algorithm 1, we use an agent per chiller (i.e. the ith agent controls the supply temperature T_i of the ith chiller) and the communication network shown in Figure 1. In all cases the initial conditions of the chillers' supply temperatures are $T_i(0) = T_s$, for $i = 1, \ldots, 7$. The results for the first cooling load, i.e. $C_L = 8862.8$ kW, are depicted in Figure 3(b), where it is shown that the cooling load is properly allocated among the chillers by adjusting the supply temperatures. More efficient chillers (i.e. chiller 3, chiller 6, and chiller 7 in Figure 3(a)) are more loaded than the less efficient ones (i.e. chiller 2 and chiller 5). This can be noticed from the fact that their supply temperatures, in steady state, reach the minimum value. Furthermore, the energy consumption is minimised and power saving reaches to 2.6%. The results for the second cooling load, i.e. $C_L = 5908.6$ kW, are shown in Figure 3(c), where it can be noticed a similar performance to that obtained with the first cooling load. However, in this case, it is not necessary that the supply temperatures reach the minimum value to meet the required load. Newly, energy consumption is minimised and power saving reaches to 2.8%. As it is stated in Section 4, convergence and optimality of the method is guaranteed under the conditions given in Theorem 4.1. In both cases we use the early stopping criterion given in Section 4.

Although other techniques have been applied to solve the optimal chiller loading problem, e.g. the ones in Chang and Chen (2009), they require centralised information. In this regard, it is worth noting that the same objective is properly accomplished by using our approach, which is fully distributed.

6.2 Distributed computation of the Euclidean projection

Several applications require computing the Euclidean projection of a vector in a distributed way. These applications include matrix updates in quasi-Newton methods, balancing of traffic flows, and decomposition techniques for stochastic optimisation (Patriksson, 2008). The problem of finding the Euclidean projection of the vector $\hat{\xi}$ onto a given set $\mathcal{X}$ is formulated as follows:

$$\begin{aligned} \min_{\xi} \ & \|\hat{\xi} - \xi\|_2^2 \\ \text{s.t.} \ & \xi \in \mathcal{X}, \end{aligned} \tag{19}$$

where $\|\cdot\|_2$ is the Euclidean norm. The vector that minimises the above problem, which is denoted by ξ^*, is the Euclidean projection. Roughly speaking, ξ^* can be seen as the closest vector to $\hat{\xi}$ that belongs to the set $\mathcal{X}$.

In Barreiro-Gomez, Obando, Ocampo-Martinez, and Quijano (2015), the authors use a distributed computation of the Euclidean projection to decouple large-scale control problems. Specifically, they propose a discrete time method to address problems involving plants comprised of a large number of decoupled sub-systems whose control inputs are coupled by a constraint. The control inputs are associated with the power applied to the sub-systems, and the constraint limits the total power used to control the whole plant. At each time iteration, local controllers that manage the sub-systems compute optimal control inputs ignoring the coupled constraint (each local controller uses a model predictive control scheme that does not use global information since the sub–systems' dynamics are decoupled). Once this is done, the coupled constraint is addressed by finding the Euclidean projection of the vector of local control inputs (i.e. the vector formed by all the control inputs computed by the local controllers) onto a domain that satisfies the constraint associated with the total power applied to the plant.

For a better explanation of the method, consider a plant comprised of n sub-systems. Let $\hat{u}_i(k) \geq 0$ be the control input computed by the ith local controller at the kth iteration ignoring the coupled constraint (non-negativity of $\hat{u}_i(k)$ is required since the control signals correspond to an applied power). Let $\hat{u}(k) = [\hat{u}_1(k), \ldots, \hat{u}_n(k)]^\top$ be the vector of local control inputs, and let $u^*(k)$ be the vector of control signals that are finally applied to the sub-systems. If the maximum allowed power to control the plant is $U > 0$, the power constraint that couples the control signals is given by $\sum_{i=1}^{n} u_i^*(k) \leq U$. The vector $u^*(k)$ is calculated by using the Euclidean projection of $\hat{u}(k)$ onto a domain that satisfies the power constraint, i.e. $u^*(k)$ is the solution of the following optimisation problem (cf. Equation (19)):

$$\min_{u(k)} \|\hat{u}(k) - u(k)\|_2^2 \tag{20a}$$

$$\text{s. t.} \sum_{i=1}^{n} u_i(k) \leq U \tag{20b}$$

$$u_i(k) \geq 0, \ \text{for all } i = 1, \ldots, n, \tag{20c}$$

where $u_i(k)$ denotes the ith entry of the vector $u(k)$. Notice that $u^*(k)$ satisfies the power constraint and minimises the Euclidean distance with respect to the control vector $\hat{u}_k$ that is initially calculated by the local controllers. Computation of $u^*(k)$ can be performed by using the approach proposed in this paper because the problem stated in Equation (20) is in the standard form given in Equation (4) except for the inequality constraint (20b). However, this constraint can be addressed by adding a slack variable.

6.2.1 Illustrative example

Consider a plant composed of 100 sub-systems. Assume that, at the kth iteration of the discrete time method presented in Barreiro-Gomez, Obando, Ocampo-Martinez, and Quijano (2015), the control inputs that are initially computed by the local controllers are given by the entries of the vector $\hat{u}(k) = [\hat{u}_1(k), \ldots, \hat{u}_{100}(k)]^\top$, where $\hat{u}_i(k)$ is a random number chosen from the interval [0, 1] kW. Furthermore, assume that the maximum allowed power to control the plant is $U = 40$ kW. To satisfy this constraint, the Euclidean projection described in Equation (20) is computed in a distributed way using Algorithm 1 with the early stopping criterion described in Section 4. The results under a communication network with path topology (see Figure 1) are depicted in Figure 4. The curve at the top of Figure 4 describes the evolution of the Euclidean distance. Notice that the proposed algorithm minimises this distance and reaches the optimum value (dashed line), which has been calculated employing a centralised method. On the other hand, the curves at the bottom of Figure 4 illustrate the evolution of the values $\sum_{i=1}^{100} u_i(k)$ (solid line) and $\min\{u_i(k)\}$ (dash-dotted line). These curves show that the constraints of the problem stated in Equation (20) are properly satisfied in steady state, i.e. $\sum_{i=1}^{100} u_i^*(k) = 40$ kW and $\min\{u_i^*(k)\} = 0$ kW. As a final observation, our algorithm exhibits a suitable

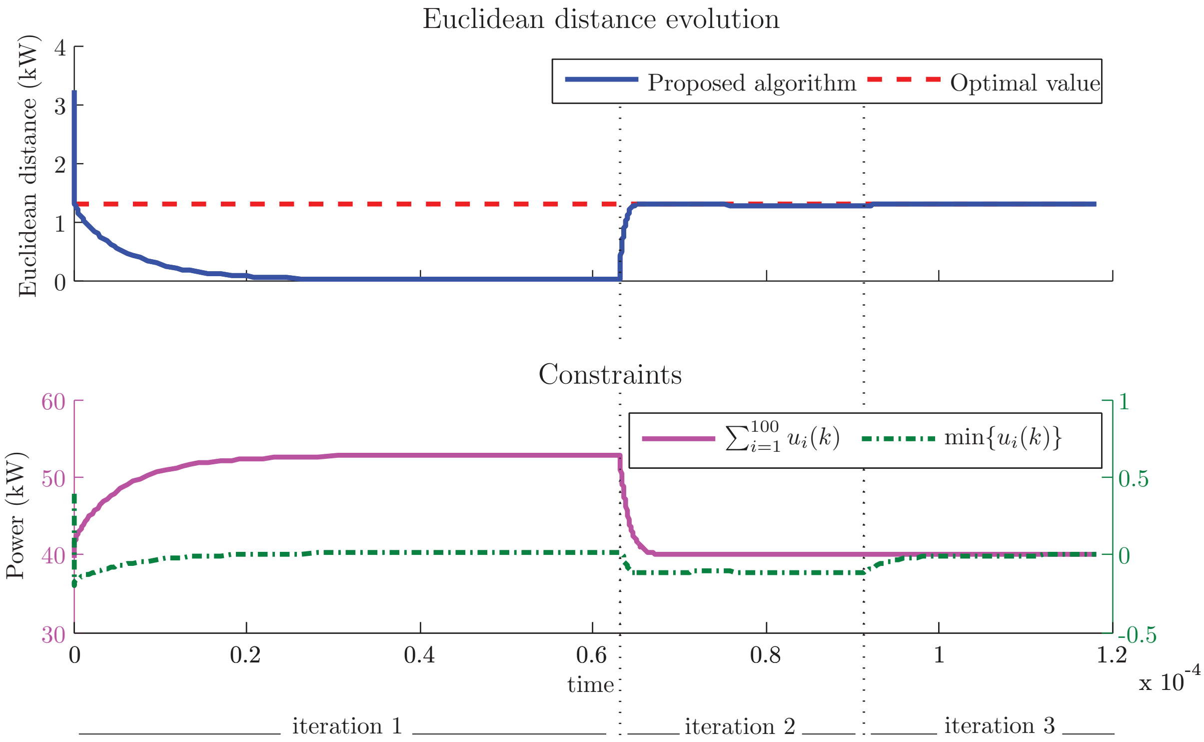


Figure 4. Evolution of the Euclidean distance and constraint satisfaction using the proposed algorithm. Right *y*-axis corresponds to the dash-dotted line.

performance even considering that the communication graph is sparse and the optimal solution is not in the interior of the feasible domain. As shown in Section 5, this characteristic is an advantage of Algorithm 1 over population dynamics techniques as the one proposed in Barreiro-Gomez, Obando, Ocampo-Martinez, and Quijano (2015) to compute the Euclidean projection in a distributed way.

7. Discussion

The method developed in this paper solves the problem of resource allocation with lower bounds given in Equation (4). The main advantage of the proposed technique is its distributed nature; indeed, our approach does not need the implementation of a centralised coordinator. This characteristic is appealing, especially in applications where communications are strongly limited. Moreover, fully distributed methodologies increase the autonomy and resilience of the system in the face of possible failures. In Section 5, we show by means of simulations that the performance of the method presented in this paper does not decrease when the number of nodes (which are related to the decision variables of the optimisation problem) is large, or the communication network that allows the nodes to share information has few channels. In these cases, the behaviour of our approach is better than the behaviour of other techniques found in the literature, such as the DIP method, or the LRE. Moreover, it is worth noting that our technique addresses the constraints as hard. This fact has two important consequences: (i) in all cases, the solution satisfies the imposed constraints, and (ii) the objective function (and therefore the optimum) is not modified (contrary to the DIP method that includes the constraints in the objective function decreasing the quality of the solution as shown in Section 5.4).

Other advantage of the method proposed in this paper is that it does not require an initial feasible solution of the resource allocation problem (4). Similarly to the DIPe technique, our method only requires that the starting point satisfies the resource constraint (4b), i.e. we need that $\sum_{i=1}^{n} x_i(0) = X$. Notice that an initial solution $x(0)$ that satisfies (4b) is not hard to obtain in a distributed manner. For instance, if we assume that only the kth node has the information of the available resource X, we can use $(x_k(0) = X, \{x_i(0) = 0 : i \in \mathcal{V}, i \neq k\})$ as our starting point. Thus, an initialisation phase is not required. In contrast, other distributed methods, such as DIP and LRE needs an initial feasible solution of the problem (4), i.e. a solution that satisfies (4b) and (4c). Finding this starting point is not a trivial problem for systems involving a large number of variables. Therefore, for these methods, it is necessary to employ distributed constraint satisfaction algorithm (as the one described in Domınguez-Garcıa & Hadjicostis, 2011) as a first step.

On the other hand, we notice that to implement the early stopping criterion presented at the end of Section 4, it is required to perform an additional *min–consensus* step in each iteration. Despite this fact, if the number of nodes is large, this criterion saves computational time, because in most of the cases, all passive nodes are identified during the first iterations of Algorithm 1.

8. Conclusions

In this paper, we have developed a distributed method that solves a class of resource allocation problems with lower bound constraints. The proposed approach is based on a multi–agent system, where coordination among agents is done by using a consensus protocol. We have proved that convergence and optimality of the method is guaranteed under some mild assumptions, specifically, we require that the cost function is strictly convex and the graph related to the communication network that enables the agents to share information is connected. The main advantage of our technique is that it does not need a centralised coordinator, which makes the method appropriate to be applied in large–scale distributed systems, where the inclusion of centralised agents is undesirable or infeasible. As future work, we propose to use a switched approach in order to eliminate the iterations in Algorithm 1. Moreover, we plan to include upper bound constraints in our original formulation.

Notes

1. As well as in Step 4 of Algorithm 1, we have initialised the auxiliary variables $\hat{x}_i$ to zero by convention. If these variables are initialised to other value, convergence of (7) is not affected (cf. Proposition 4.2).
2. Algorithms were implemented in a computer with an Intel Core i5 processor.
3. Simulation parameters are adapted from Chang and Chen (2009).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

G. Obando is supported in part by Convocatoria 528 Colciencias–Colfuturo and in part by OCAD–Fondo de CTel SGR, Colombia (ALTERNAR project, BPIN 20130001000089).

References

Ayesta, U., Erausquin, M., Ferreira, E., & Jacko, P. (2016). Optimal dynamic resource allocation to prevent defaults. *Operations Research Letters, 44*(4), 451–456.

Barreiro-Gomez, J., Obando, G., Ocampo-Martinez, C., & Quijano, N. (2015). Making non-centralized a model predictive control scheme by using distributed smith dynamics. In *Proceedings of the 5th IFAC Conference on Nonlinear Model Predictive Control* (Vol. 48, no. 23, pp. 501–506). IFAC-PapersOnLine.

Barreiro-Gomez, J., Obando, G., & Quijano, N. (2016). Distributed population dynamics: Optimization and control applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, doi:10.1109/TSMC.2016.2523934

Bemporad, A., Heemels, M., & Johansson, M. (2010). *Networked control systems*. London: Springer.

Bertsekas, D.P. (1975). Necessary and sufficient conditions for a penalty method to be exact. *Mathematical Programming, 9*(1), 87–99.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning, 3*(1), 1–122.

Chang, Y., & Chen, W. (2009). Optimal chilled water temperature calculation of multiple chiller systems using Hopfield neural network for saving energy. *Energy, 34*(4), 448–456.

Cherukuri, A., & Cortés, J. (2015). Distributed generator coordination for initialization and anytime optimization in economic dispatch. *IEEE Transactions on Control of Network Systems, 2*(3), 226–237.

Conrad, J. (2010). *Resource economics*. Cambridge: Cambridge University Press.

Cortés, J. (2008). Distributed algorithms for reaching consensus on general functions. *Automatica, 44*(3), 726–737.

Dominguez-Garcia, A., Cady, S., & Hadjicostis, C. (2012). Decentralized optimal dispatch of distributed energy resources. In Edwin K. P. Chong, et al. (Eds.) *Proceedings of the 51st IEEE Conference on Decision and Control* (pp. 3688–3693). Maui: IEEE.

Domınguez-Garcıa, A., & Hadjicostis, C. (2011). Distributed algorithms for control of demand response and distributed energy resources. In Jay Farrell, et al. (Eds.) *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference* (pp. 27–32).

Garin, F., & Schenato, L. (2010). A survey on distributed estimation and control applications using linear consensus algorithms. In Alberto Bemporad, Maurice Heemels, Mikael Johansson (Eds.) *Networked Control Systems* (pp. 75–107). Springer.

Godsil, C., & Royle, G. (2001). *Algebraic graph theory, volume 207 of Graduate Texts in Mathematics*. London: Springer-Verlag.

Hansen, T.M., Roche, R., Suryanarayanan, S., Maciejewski, A.A., & Siegel, H.J. (2015). Heuristic optimization for an aggregator-based resource allocation in the smart grid. *IEEE Transactions on Smart Grid, 6*(4), 1785–1794.

Jensen, P.A., & Bard, J.F. (2003). *Operations research models and methods*. Hoboken: John Wiley & Sons.

Johansson, B., & Johansson, M. (2009). Distributed non-smooth resource allocation over a network. In Lei Guo, et al. (Eds.) *Proceedings of the 48th IEEE Conference on*

Appendix D

Appendix 4 – Paper [C.sub1]:

- [C.sub1] M. Canu and N. Rakoto-Ravalontsalama
  On Switchable Languages of Discrete-Event Systems with Weighted Automata.
  Submitted, March 2017.

On Switchable Languages of Discrete-Event Systems with Weighted Automata

Michael Canu and Naly Rakoto-Ravalontsalama

***Abstract*— The notion of switchable languages has been defined by Kumar, Takai, Fabian and Ushio in [11]. It deals with switching supervisory control, where switching means switching between two specifications. In this paper, we first extend the notion of switchable languages to n languages, ($n \geq 3$). Then we consider a discrete-event system modeled with weighted automata. The use of weighted automata is justified by the fact that it allows us to synthesize a switching supervisory controller based on the cost associated to each event, like the energy for example. Finally the proposed methodology is applied to a simple example.**

Keywords: *Supervisory control; switching control; weighted automata.*

I. Introduction

Supervisory control initiated by Ramadge and Wonham [15] provides a systematic approach for the control of discrete event system (DES) plant. There has been a considerable work in the DES community since this seminal paper. On the other hand, from the domain of continuous-time system, hybrid and switched systems have received a growing interests [12]. The notion of switching is an important feature that has to be taken into account, not only in the continuous-time domain but for the DES area too.

As for non-blocking property, there exist different approaches. The first one is the non-blocking property defined in [15]. Since then other types of non-blocking properties have been defined. The mutually non-blocking property has been proposed in [5]. Other approaches of mutually and globally nonblocking supervision with application to switching control is proposed in [11]. Robust non-blocking supervisory control has been proposed in [1]. Other types of non-blocking include the generalised non-blocking property studied in [13]. Discrete-event modeling with switching max-plus systems is proposed in [17], an example of mode switching DES is described in [6] and finally a modal supervisory control is considered in [7].

In this paper we will consider the notion of switching supervisory control defined by Kumar and Colleagues in [11] where switching means switching between a pair of specifications. Switching (supervisory) control is in fact an application of some results obtained in the same paper [11] about mutually non blocking properties of languages, mutually nonblocking supervisor existence, supremal controllable, relative-closed and mutually nonblocking languages. All these results led to the definition of a *pair of switchable languages* [11].

In this paper, we first extend the notion of switchable languages to n languages, ($n \geq 3$). Then we consider a discrete-event system modeled with weighted automata. The switching supervisory control strategy is based on the cost associated to each event, and it allows us to synthesize an optimal supervisory controller. Finally the proposed methodology is applied to a simple example.

This paper is organized as follows. In Section II, we recall the notation and some preliminaries. Then in Section III the main results on the extension of n switchable languages ($n \geq 3$) are given. An illustrative example of supervisory control of AGVs is proposed in Section IV, and finally a conclusion is given in Section V.

II. Notation and Preliminaries

Let the discrete event system plant be modeled by a finite state automaton [10],[4] to which a cost function is added.

Definition 1: (Weighted automaton). A weighted automaton is defined as a sixtuple

$$G = (Q, \Sigma, \delta, q_0, Q_m, \mathbb{C})$$

where

- Q is the finite set of states,
- Σ is the finite set of events,
- $\delta : Q \times \Sigma \to Q$ is the partial transition function,

M. Canu is with Univ. los Andes, Bogota, Colombia, e-mail: *m.canu134@uniandes.edu.co*

N. Rakoto-Ravalontsalama is with IMT Atlantique and LS2N, France, e-mail: *naly.rakoto@mines-nantes.fr*

- $q_0 \subseteq Q$ is the initial state,
- $Q_m \subseteq Q$ is the set of *marked* states (final states),
- $\mathbb{C} : \Sigma \to \mathbb{N}$ is the cost function.

Let Σ^* be the set of all finite strings of elements in Σ including the empty string ε. The transition function δ can be generalized to $\delta : \Sigma^* \times Q \to Q$ in the following recursive manner:

$\delta(\varepsilon, q) = q$
$\delta(\omega\sigma, q) = \delta(\sigma, \delta(\omega, q))$ for $\omega \in \Sigma^*$

The notation $\delta(\sigma, q)!$ for any $\sigma \in \Sigma^*$ and $q \in Q$ denotes that $\delta(\sigma, q)$ is defined. Let $L(G) \subseteq \Sigma^*$ be the language generated by G, that is,

$$L(G) = \{\sigma \in \Sigma^* | \delta(\sigma, q_0)!\}$$

Let $K \subseteq \Sigma^*$ be a language. The set of all prefixes of strings in K is denoted by $pr(K)$ with $pr(K) = \{\sigma \in \Sigma^* | \exists\ t \in \Sigma^*; \sigma t \in K\}$. A language K is said to be *prefix closed* if $K = pr(K)$. The event set Σ is decomposed into two subsets Σ_c and Σ_{uc} of controllable and uncontrollable events, respectively, where $\Sigma_c \cap \Sigma_{uc} = \emptyset$. A controller, called a supervisor, controls the plant by dynamically disabling some of the controllable events.

A sequence $\sigma_1\sigma_2 \ldots \sigma_n \in \Sigma^*$ is called a *trace* or a *word* in term of language. We call a *valid trace* a path from the initial state to a marked state ($\delta(\omega, q_0) = q_m$ where $\omega \in \Sigma^*$ and $q_m \in Q_m$). The cost is by definition non negative. In the same way, the cost function $\mathbb{C}$ is generalized to the domain Σ^* as follows:

$\mathbb{C}(\varepsilon) = 0$
$\mathbb{C}(\omega\sigma) = \mathbb{C}(\omega) + \mathbb{C}(\sigma)$ for $\omega \in \Sigma^*$

In other words, the cost of a trace is the sum of the costs of each event that composes the trace.

Definition 2: (Controllability) [15]. A language $K \subseteq L(G)$ is said to be *controllable* with respect to (w.r.t.) $L(G)$ and Σ_{uc} if

$$pr(K)\Sigma_{uc} \cap L(G) \subseteq pr(K).$$

Definition 3: (Mutually non-blocking supervisor) [5]. a supervisor $f : L(G) \to 2^{\Sigma - \Sigma_u}$ is said to be (K_1, K_2)-*mutually non-blocking* if

$$K_i \cap L_m(G^f) \subseteq pr(K_j \cap L_m(G^f)), \text{ for } i, j \in \{1, 2\}. \quad (1)$$

In other words, a supervisor S is said to be *mutually non-blocking* w.r.t. two specifications K_1 and K_2 if whenever the closed-loop system has completed a task of one language (by completing a marked trace of that language), then it is always able to continue to complete a task of the other language [5].

Definition 4: (Mutually non-blocking language) [5]. A language $H \subseteq K_1 \cup K_2$ is said to be (K_1, K_2)-*mutually non-blocking* if $H \cap K_i \subseteq pr(H \cap K_j)$ for $i, j \in \{1, 2\}$.

The following theorem gives a necessary and sufficient condition for the existence of a supervisor.

Theorem 1: (Mutually nonblocking supervisor existence) [5]. Given a pair of specifications $K_1, K_2 \subseteq L_m(G)$, there exists a globally and mutually non-blocking supervisor f such that $L_m(G^f) \subseteq K_1 \cup K_2$ if and only if there exists a nonempty, controllable, relative-closed, and (K_1, K_2)-mutually non-blocking sublanguage of $K_1 \cup K_2$.

The largest possible language (the supremal element) that is controllable and mutually non-blocking exists, as stated by the following theorem.

Theorem 2: (SupMRC($K_1 \cup K_2$) existence) [5]. The set of controllable, relative-closed, and mutually non-blocking languages is closed under union, so that the supremal such sublanguage of $K_1 \cup K_2$, denoted $supMRC(K_1 \cup K_2)$ exists.

Recall that a pair of languages K_1, K_2 are *mutually nonconflicting* if $pr(K_1 \cap K_2) = pr(K_1) \cap pr(K_2)$ [18]. K_1, K_2 are called *mutually weakly nonconflicting* if $K_i, pr(K_j)$ $(i \neq j)$ are mutually nonconflicting [5].

Another useful result from [5] is the following. Given a pair of mutually weakly nonconflicting languages $K_1, K_2 \subseteq L_m(G)$, the following holds ([5], Lemma 3). If K_1, K_2 are controllable then $K_1 \cap pr(K_2), K_2 \cap pr(K_1)$ are also controllable.

The following theorem is proposed in [11] and it gives the formula for the supremal controllable, relative-closed, and mutually nonblocking languages.

Theorem 3: (SupMRC($K_1 \cup K_2$)) [11]. For relative-closed specifications $K_1, K_2 \subseteq L_m(G)$, $supMRC(K_1 \cup K_2) = supRC(K_1 \cap K_2)$.

The following theorem, also from [11] gives another expression of the supremal controllable, relative-closed, and mutually nonblocking languages.

Theorem 4: [11] Given a pair of controllable, relative-closed, and mutually weakly nonconflicting languages $K_1, K_2 \subseteq L_m(G)$, it holds that $supMRC(K_1 \cup K_2) = (K_1 \cap K_2)$.

And finally the following theorem gives a third formula of the supremal controllable, relative-closed, and mutually nonblocking languages.

Theorem 5: [11] For specifications $K_1, K_2 \subseteq L_m(G)$, $supMRC(K_1 \cup K_2) = supMC(supRC(K_1 \cap K_2))$.

In order to allow switching between specifications, a pair of supervisors is considered, such that the supervisor is switched when the specification is switched. The supervisor f_i for the specification K_i is designed to enforce a certain sublanguage $H_i \subseteq K_i$. Suppose a switching in specification from K_i to K_j is induced at a point when a trace $s \in H_i$ has been executed in the f_i-controlled plant. Then in order to be able to continue with the new specification K_j *without reconfiguring the plant*, the trace s must be a prefix of $H_j \subseteq K_j$. In other words, the two supervisors should enforce the languages H_i and H_j respectively such that $H_i \subseteq pr(H_j)$. Hence the set of pairs of such languages are defined to be *switchable languages* as follows.

Definition 5: (Pair of switchable languages) [11]. A pair of specifications $K_1, K_2 \subseteq L_m(G)$ are said to be *switchable languages* if
$SW(K_1, K_2) := \{(H_1, H_2) | H_i \subseteq K_i \cap pr(H_j), i \neq j,$ and H_i controllable$\}$.

The supremal pair of switchable languages exists and is given by the following theorem.

Theorem 6: (Supremal pair of switchable languages) [11]. For specifications $K_1, K_2 \subseteq L_m(G)$,
$supSW(K_1, K_2) =$
$(supMC(K_1 \cup K_2) \cap K_1, supMC(K_1 \cup K_2) \cap K_2)$.

III. Main Results

We now give the main results of this paper. First, we define a triplet of switchable languages. Second we derive a necessary and sufficient condition for the transitivity of switchable languages ($n = 3$). Third we generalize this definition to a n-uplet of switchable languages, with $n > 3$. And fourth we derive a necessary and sufficient condition for the transitivity of switchable languages for $n > 3$.

A. Triplet of Switchable Languages

We extend the notion of pair of switchable languages, defined in [11], to a triplet of switchable languages.

Definition 6: (Triplet of switchable languages). A triplet of languages (K_1, K_2, K_3), $K_i \subseteq L_m(G)$ with $H_i \subseteq K_i$, $i = \{1, 2, 3\}$ are said to be a *triplet of switchable languages* if they are pairwise switchable languages, that is,

$SW(K_1, K_2, K_3) := SW(K_i, K_j), \; i \neq j, \; i,j = \{1, 2, 3\}$.

Another expression of the triplet of switchable languages is given by the following lemma.

Lemma 1: (Triplet of switchable languages). A triplet of languages (K_1, K_2, K_3), $K_i \subseteq L_m(G)$ with $H_i \subseteq K_i$, $i = \{1, 2, 3\}$ are said to be a *triplet of switchable languages* if the following holds:

$SW(K_1, K_2, K_3) = \{(H_1, H_2, H_3) \mid H_i \subseteq K_i \cap pr(H_j), i \neq j,$ and H_i controllable$\}$.

B. Transitivity of Switchable Languages ($n = 3$)

The following theorem gives a necessary and sufficient condition for the transitivity of switchable languages.

Theorem 7: (Transitivity of switchable languages, $n = 3$) . Given 3 specifications (K_1, K_2, K_3), $K_i \subseteq L_m(G)$ with $H_i \subseteq K_i$, $i = \{1, 2, 3\}$ such that $SW(K_1, K_2)$ and $SW(K_2, K_3)$.
(K_1, K_3) is a pair of switchable languages, i.e. $SW(K_1, K_3)$, if and only if

1) $H_1 \cap pr(H_3) = H_1$, and
2) $H_3 \cap pr(H_1) = H_3$.

Proof: The proof can be found in [3]. ∎

C. N-uplet of Switchable Languages

We now extend the notion of switchable languages, to a n-uplet of switchable languages, with ($n > 3$).

Definition 7: (N-uplet of switchable languages, $n > 3$). A n-uplet of languages $(K_1, ..., K_n)$, $K_i \subseteq L_m(G)$ with $H_i \subseteq K_i$, $i = \{1, ..., n\}, n > 2$, is said to be a *n-uplet of switchable languages* if the languages are pairwise switchable that is,

$SW(K_1, ..., K_n) := SW(K_i, K_j), \; i \neq j, \; i,j = \{1, ..., n\}, n > 2$.

As for the triplet of switchable languages, an alternative expression of the n-uplet of switchable languages is given by the following lemma.

Lemma 2: (N-uplet of switchable languages, $n > 3$). A n-uplet of languages $(K_1, \ldots, K_n)$, $K_i \subseteq L_m(G)$ with $H_i \subseteq K_i,\ i = \{1, ..., n\},\ n > 3$ are said to be a *n-uplet of switchable languages* if the following holds:

$SW(K_1, ..., K_n) = \{(H_1, ..., H_n) \mid H_i \subseteq K_i \cap pr(H_j), i \neq j$, and H_i controllable$\}$.

D. Transitivity of Switchable Languages $(n > 3)$

We are now able to derive the following theorem that gives a necessary and sufficient condition for the transitivity of n switchable languages.

Theorem 8: (Transitivity of n switchable languages, $n > 3$) . Given n specifications $(K_1, ..., K_n)$, $K_i \subseteq L_m(G)$ with $H_i \subseteq K_i,\ i = \{1, ..., n\}$. Moreover, assume that each language K_i is at least switchable with another language $K_j,\ i \neq j$.
A pair of languages (K_k, K_l) is switchable i.e. $SW(K_k, K_l)$, if and only if

1) $H_k \cap pr(H_l) = H_k$, and
2) $H_l \cap pr(H_k) = H_l$.

Proof: The proof is similar to the proof of Theorem 6 and can be found in [3]. ■
It is to be noted that the assumption that each of the n languages be at least switchable with another language is important, in order to derive the above result.

IV. EXAMPLE: SWITCHING SUPERVISORY CONTROL OF AGVS

The idea of switching supervisory control is now applied to a discrete-event system, modeled with weighted automata. We take as an illustrating example the supervisory control of a fleet of fleet automated guided vehicles (AGVs) that move in a given circuit area. The example is taken from [9]. A circuit is partitioned into sections and intersections. Each time an AGV moves in a new intersection or a new section, then the automaton will move to a new state in the associated automaton. An example of an area with its associated basic automaton is depicted in Figure 1.

The area to be supervised is the square depicted in Figure 1 (left). The flow direction with the arrows are specified the four intersections $\{A, B, C, D\}$ and the associated basic automaton are given in Figure 1 (right). The basic automaton is denoted $G_{basic} = (Q_b, \Sigma_b, \delta_b, \emptyset, \emptyset)$ where the initial state and the final

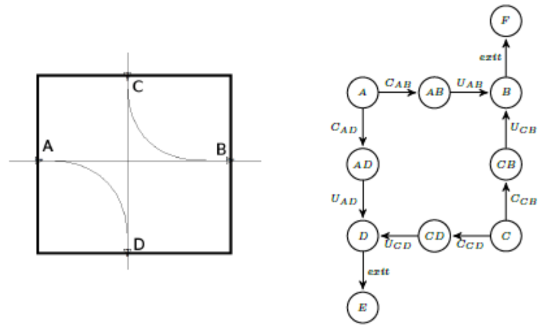


Fig. 1. An AGV circuit (left) and its basic automaton (right)

state are not defined. The initial state is defined according to the physical position of the AGV and the final state is defined according to its mission, that is his position target. A state represents and intersection or a section. Each state corresponding to a section is named XY_i where X is the beginning of the section, Y its end and i the number of the AGV. For each section, there are two transitions, the first transition C_{XY} is an input which is controllable and represents the AGV moving on the section from X to Y. The second transition is an output transition U_Y which is uncontrollable and represents the AGV arriving to the intersection Y.
For example the basic automaton depicted in Figure 1 (right) can be interpreted as follows. If AGV_i arrives at section A, then it has two possibilities, either to go to section B with the event C_{ABi}, or the go section D with the event C_{ADi}. If we choose to go to section B, then the next state is AB_i. From this state, the uncontrollable event U_{AB} is true so that the following state is B_i. And from B_i, the only possibility is to exit to Point F with the uncontrollable event $exit_i$.
Now consider for example that 2 AGVs are moving in the circuit of Figure 1 (left). Assume AGV_1 is in D and AGV_2 is in AB so that the state is in (D_1, AB_2). AGV_1 is leaving the area when the event $exit_1$ is true so that the system will be in state (E_1, AB_2). And since AGV_1 is out of the considered area, then the new state will be $(E_1, AB_2) = (\emptyset_1, AB_2) = (AB_2)$ since AGV_1 is out of the area.

We give here below the synthesis algorithm for calculating the supervisor S_c as it aws proposed by Girault et Colleagues in [9]. For more details on the synthesis algorithm, the reader is referred to the above paper.

Algorithm 1 – Synthesis algorithm of S_C [9]

Data: $G_{w,1}, \dots G_{w,n}$
Result: Supervisor S_C

$G_w \leftarrow \{G_{w,1}, \dots G_{w,n}\}$
$G_u \leftarrow \{\emptyset\}$
forall $G_{w,i} \in G_w$ **do**
  $G_u \leftarrow G_u \cup U_{\gamma_i}(G_{w,i})$
end
$S_C \leftarrow S(G_{u,i})$
$G_u \leftarrow G_u \backslash \{G_{u,1}\}$
while $G_u \neq \emptyset$ **do**
  $x \leftarrow get(G_u)$
  $S_C \leftarrow S(S_C || x)$
  $G_u \leftarrow G_u \backslash \{x\}$
end

V. Conclusions

The notion of switchable languages has been defined by Kumar and Colleagues in [11]. It deals with switching supervisory control, where switching means switching between two specifications. In this paper, we have extended the notion of switchable languages to a triplet of languages ($n = 3$) and we gave a necessary and sufficient condition for the transitivity of two switchable languages. Then we generalized the notion of switchable languages of a n-uplet of languages, $n > 3$ and we gave also necessary and sufficient condition for the transitivity of two (out of n) switchable languages. Finally the proposed methodology is applied to a simple example for the supervisory control of a fleet of AGVs. Ongoing work deals with a) the calculation of the supremal of n-uplet of switchable languages, and b) the optimal switching supervisory control of DES exploiting the cost of the weighted automata for the synthesis strategy.

Acknowledgment

Part of this work was carried out when the second author (N.R.) was visiting Prof. Stephane Lafortune at University of Michigan, Ann Arbor, MI, USA, in Sep. 2013. Grant #EMN-DAP-2013-09 is gratefully acknowledged.

References

[1] S.E. Bourdon, M. Lawford, and W.M. Wonham "Robust nonblocking supervisory control of discrete event systems," In *IEEE Trans. on Automatic Control*, vol. 50, N.12 pp. 2015–2021, 2005.

[2] M. Canu and N. Rakoto-Ravalontsalama, *From mutually non-blocking to switched non-blocking DES*. Presented at MSR'13 Workshop (Poster Session), Rennes, France, Nov 13-15, 2013.

[3] M. Canu and N. Rakoto-Ravalontsalama. *On Switchable Languages of Discrete-Event Systems with Weighted Automata*, Technical Report Mines Nantes, March 2017.

[4] C.G. Cassandras and S. Lafortune, "Introduction to Discrete Event Systems," 2nd Edition, Springer Verlag, 2008.

[5] M. Fabian and R. Kumar. "Mutually non-blocking supervisory control of discrete-event systems," In *Automatica*, 36(12) pp. 1863–1869, 2000.

[6] G. Faraut, L. Pietrac, and E. Niel, "Formal Approach to Multimodal Control Design: Application to Mode Switching", In *IEEE Trans. on Industrial Informatics*, vol.5, N.4 pp. 443–453, Nov 2009.

[7] G. Faraut, L. Pietrac, and E. Niel, "Process Tracking by Equivalent States in Modal Supervisory Control", Proc. of *IEEE ETFA*, Sep. 2011, Toulouse, France.

[8] J. Girault, J.J. Loiseau, O.H. Roux "Synthese en ligne de superviseur compositionnel pour une flotte de robots mobiles." European Journal of Automation, MSR'13, vol 47/1-3, pp. 195–210, 2013.

[9] J. Girault, J.J. Loiseau, O.H. Roux "On-line optimal compositional controller synthesis for AGV by unfolding," Proc. of DCDS 2015, IFAC-PapersOnLine 48-7 (2015) pp. 167–173.

[10] J.E. Hopcroft and J.D. Ullman, "Introduction to Automata Theory, Languages, and Computation," Addison-Wesley, Reading, MA, USA, 1979.

[11] R. Kumar, S. Takai, M. Fabian, and T. Ushio, "Maximally Permissive Mutually and Globally Nonblocking Supervision with Application to Switching Control," In *Automatica*, 41(8) pp. 1299–1312, 2005.

[12] D. Liberzon, "Switching in Systems and Control", ser. Systems and Control: Foundations and Applications. Boston: Birkhauser, 2003.

[13] R. Malik and R. Leduc, "Generalised nonblocking"', in Proc. 9th Int. Workshop on Discrete Event

Systems, WODES 2008, Goteborg, Sweden, May 2008, pp. 340–345.

[14] N. Rakoto-Ravalontsalama. ”Supervisory control of switched discrete-event systems,” in Proc. of 17th Symp. on MTNS 2006, Kyoto, Japan 2006, pp. 2213–2217.

[15] P.J. Ramadge and W.M. Wonham. ”Supervisory control of a class of discrete-event processes,” In *SIAM J. Control and Optimization*, vol.25 pp. 206–230, 1987.

[16] P.S. Roop, A. Girault, R. Sinha, and G. Goessler. ”Specification Enforcing Refinement for Convertibility Verification,” Proc. of ACSD 2009, pp. 148–157, IEEE, 2009.

[17] T.J.J. van den Boom and B. de Schutter. ”Modelling and control of discrete event systems using switching max-plus-linear systems,” *Control Engineering Practice*, vol. 14 N.10, pp. 1199-1211, 2006.

[18] W.M. Wonham and P.J. Ramadge. ”Modular supervisory control of discrete-event systems,” In *Mathematics of Control, Signals and Systems*, vol.1 (1) pp. 13–30, 1988.