De l'Apprentissage Statistique pour le Contrôle Optimal et le Traitement du Signal

Olivier Pietquin


UNIVERSITE PAUL SABATIER - TOULOUSE III
Ecole Doctorale : MITT Toulouse
Spécialité : Informatique

Thèse

présentée pour l'obtention de

L'Habilitation à Diriger des Recherches de l'Université Paul Sabatier

par **Olivier PIETQUIN**

De l'Apprentissage Statistique pour le Contrôle Optimal et le Traitement du Signal

Contribution à la Prise en Compte de l'Humain dans les Systèmes Intelligents

Soutenue le 10 mai 2011

Membres du jury

Rapporteurs :	Olivier Cappé	Directeur de Recherche CNRS - LCTI
	Fabrice Lefèvre	Professeur, Université d'Avignon et Pays du Vaucluse - LIA-CERI
	Olivier Sigaud	Professeur, Université Pierre et Marie Curie, Paris - ISIR
Examinateurs :	Régine André-Obrecht	Professeur, Université Paul Sabatier, Toulouse - IRIT
	Florence Sèdes	Professeur, Université Paul Sabatier, Toulouse - IRIT
	François Charpillet	Directeur de Recherche INRIA, Nancy Grand-Est
	Frédérick Garcia	Directeur de Recherche INRA - BIA

J'adore qu'un plan se déroule sans accroc !
- Col. John "Hannibal" Smith

A ceux qui ont encore le bénéfice du doute.

Résumé

Les travaux présentés dans ce manuscrit ont pour but le développement de méthodes de prise de décisions optimales (statiques ou séquentielles) ou de traitement de signaux par des méthodes d'apprentissage automatique. Ils répondent cependant à un certain nombre de contraintes qui ont été imposées par l'objectif de prendre en compte la présence de l'humain dans la boucle de décision ou de traitement ou même comme générateur des signaux que l'on analyse. La présence de l'humain rend la nature des données que l'on traite assez imprévisible et, à une même situation en apparence, des décisions différentes peuvent être prises. Dans les domaines de l'apprentissage statistique, du contrôle optimal ou du traitement du signal ceci se traduit par la nécessité de gérer l'incertain, de traiter le caractère stochastique du système ainsi que la non-stationnarité de celui-ci. Ainsi, les décisions que l'on considère optimales peuvent dépendre de manière aléatoire de la situation mais de plus, cette dépendance peut varier avec le temps. Des méthodes d'optimisation convergeant vers une solution globale ne sont donc pas adaptées mais des méthodes permettant d'apprendre au fil de l'eau et de poursuivre l'évolution de la solution optimale seront préférées.

Par ailleurs, dans le cas où les décisions résultent en une action sur le monde extérieur, il est nécessaire de quantifier le risque pris en accomplissant ces actions, particulièrement si ces actions doivent avoir un impact sur l'humain. Ceci passe par une estimation de l'incertitude sur le résultat des actions possibles et la sélection en conséquence de ces actions. Une autre implication est qu'il ne sera pas toujours envisageable de tester toutes les actions possibles pour en estimer les effets puisque ces actions peuvent ne pas être acceptables pour l'humain. Ainsi, il faudra apprendre à partir d'exemples de situations imposées ce qui se traduit par une phase d'inférence utilisant les informations observables pour en déduire des conséquences sur des situations que l'on ne peut observer.

Les travaux exposés dans ce manuscrit apportent des contributions théoriques permettant de tenir compte de ces contraintes et des applications à des problèmes concrets imposant ces contraintes seront exposées.

Préambule

Dans ce document sont résumés les travaux réalisés depuis l'obtention de notre thèse en 2004. Ces travaux ont été effectués au sein de l'équipe "*Information, Multimodalité & Signal*" (IMS [1]) dont nous avons été responsable depuis sa création en 2006 jusqu'en 2010 et située sur le campus de Metz de SUPELEC, ainsi que de l'équipe INSERM "*Imagerie Adaptative, Diagnostique et Interventionnelle*" (IADI) basée au CHU de Nancy.

Les recherches décrites dans ce document ont pour point commun un traitement statistique de l'information avec des applications dans les domaines du traitement du signal, de l'intelligence artificielle et du contrôle optimal. En particulier, elles concernent la prise de décisions séquentielles, l'estimation de paramètres et la segmentation non-supervisée et sans modèle de signaux (son, images).

Il en résulte des applications, des projets et des publications pluridisciplinaires témoignant de notre intérêt pour la mise en pratique de la théorie mais aussi pour la généricité des solutions développées. Ainsi, ces travaux ont trouvé des applications dans les contextes de l'optimisation de systèmes de dialogue homme-machine, la gestion d'énergie dans l'industrie métallurgique, l'assistance à la conduite de véhicules, le traitement de signaux biomédicaux, (électrocardiogramme, images IRM fonctionnelles), le traitement de signaux multimédias (son, image, vidéo). Dans la grande majorité de ces travaux, les applications visent à suppléer à l'intervention humaine ou à en améliorer l'efficacité mais aussi à coopérer avec l'humain pour l'aider à réaliser une tâche.

Les contributions scientifiques seront exposées en 2 parties :

– Apprentissage statistique pour le contrôle optimal (Partie I)
– Apprentissage statistique pour le traitement du signal (Partie II)

Chacune de ces parties sera illustrée par un certain nombre d'applications. Ces travaux ont été particulièrement menés dans le cadre des thèses suivantes encadrées en partie par nos soins,

1. M. Geist. *Optimisation des chaînes de production dans l'industrie sidérurgique : une approche statistique de l'apprentissage par renforcement*. Thèse en mathématiques, Université Paul Verlaine de Metz, Novembre 2009,
2. J. Oster. *Traitement temps-réel des signaux électrophysiologiques acquis dans un environnement d'Imagerie par Résonance Magnétique*. Thèse en automatique et traitement du signal, Nancy Université, Novembre 2009,
3. B. Chevaillier. *Analyse de données d'IRM fonctionnelle rénale par quantification vectorielle*. Thèse en mathématiques, Université Paul Verlaine de Metz, mars 2010,

mais aussi dans le cadre de différents projets collaboratifs régionaux, nationaux ou européens.

Les publications parues pendant et depuis la thèse sont listées ci-après et rappelées dans la bibliographie à la fin de ce document. Notons qu'une grande majorité d'entre elles font état de travaux réalisés après la thèse (depuis 2004 donc).

Livres

1. O. Pietquin and O. Lemon, editors. *Data Driven Methods for Adaptive Spoken Dialogue Systems*. Springer, 2011. accepted for publication

1. http ://ims.metz.supelec.fr

2. O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. SIMILAR Collection. Presses Universitaires de Louvain, 2004. 246 pages

Articles de revues internationales

1. O. Pietquin, M. Geist, S. Chandramohan, and H. Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 2011. 21 pages - accepted for publication - Impact Factor 2.214
2. O. Pietquin and H. Hastie. A survey on metrics for the evaluation of user simulations. *Knowledge Engineering Review*, 2011. 15 pages - accepted for publication - Impact Factor 1.611
3. B. Chevaillier, D. Mandry, J.-L. Collette, M. Claudon, M.-A. Galloy, and O. Pietquin. Functional segmentation of renal DCE-MRI sequences using unsupervised learning algorithms. *Neural Processing Letters*, page 14 pages, 2011. accepted for publication - Impact Factor 1.47
4. M. Geist and O. Pietquin. Kalman temporal differences. *Journal of Artificial Intelligence Research (JAIR)*, 39 :489–532, October 2010. - Impact Factor (2008) 3.241
5. B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Adaptive Black Blood Fast Spin Echo for End-Systolic Rest Cardiac Imaging. *Magnetic Resonance in Medicine*, 64(6) :1760–1771, December 2010. - Impact Factor 3.131
6. J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Nonlinear Bayesian Filtering for Denoising of Electrocardiograms acquired in a Magnetic Resonance Environment. *IEEE Transactions on Biomedical Engineering*, 57(7) :1628 – 1638, May 2010. Impact Factor 1.322
7. J. Oster, O. Pietquin, R. Abächerli, M. Kraemer, and J. Felblinger. Independent component analysis based artefact reduction : application to electrocardiogram for improved magnetic resonance imaging triggering. *Physiological Measurement*, 30 :1381–1397, November 2009. Impact Factor 1.691
8. M. Geist, O. Pietquin, and G. Fricout. From supervised to reinforcement learning : a kernel-based Bayesian filtering framework. *International Journal On Advances in Software*, 2(1) :101–116, 2009
9. O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2) :589–599, March 2006. Impact Factor 1.848
10. O. Pietquin, L. Couvreur, and P. Couvreur. Applied clustering for automatic speaker-based segmentation of audio materials. *Journal of Operations Research, Statistics and Computer Science (JORBEL now 4OR)*, 41(1-2) :1–12, 2001. - Impact Factor 1.089

Articles de revues nationales

1. M. Geist, O. Pietquin, and G. Fricout. Différences temporelles de Kalman : cas déterministe. *Revue d'Intelligence Artificielle*, 24(2) :423–442, September 2010

Chapitres de livres

1. O. Pietquin. Natural language and dialogue processing. In F. M. Jean-Philippe Thiran, Hervé Bourlard, editor, *Multi-modal signal processing : methods and techniques to build multimodal interactive systems*, chapter 4, pages 61–90. Elsevier Science & Technology Books, January 2010
2. B. Chevaillier, D. Mandry, J.-L. Collette, M. Claudon, and O. Pietquin. Functional semi-automated segmentation of renal DCE-MRI sequences using a Growing Neural Gas algorithm. In A. A. Zaher, editor, *Recent Advances in Signal Processing*, chapter 5, pages 69–80. Nov 2009
3. O. Pietquin. Machine learning methods for spoken dialogue simulation and optimization. In A. Mellouk and A. Chebira, editors, *Machine Learning*, pages 167–184. IN-TECH, January 2009

4. M. Geist, O. Pietquin, and G. Fricout. Bayesian reward filtering. In S. G. et al., editor, *Recent Advances in Reinforcement Learning*, volume 5323 of *Lecture Notes in Computer Science (LNCS)*, pages 96–109. Springer Verlag, June 2008. Revised and selected papers of EWRL 2008
5. O. Pietquin. Optimising spoken dialogue strategies within the reinforcement learning paradigm. In M. E. Cornelius Weber and N. M. Mayer, editors, *Reinforcement Learning, Theory and Applications*, pages 239–256. I-Tech Education and Publishing, Vienna, Austria, January 2008
6. O. Pietquin. Machine learning for spoken dialogue management : an experiment with speech-based database querying. In J. E. . J. Domingue, editor, *Artificial Intelligence : Methodology, Systems & Applications*, volume 4183 of *Lecture Notes in Artificial Intelligence*, pages 172–180. Springer Verlag, 2006

Actes de conférences internationales avec comité de lecture

1. O. Pietquin, M. Geist, and S. Chandramohan. Sample Efficient On-line Learning of Optimal Dialogue Policies with Kalman Temporal Differences. In *International Joint Conference on Artificial Intelligence (IJCAI 2011)*, Barcelona, Spain, July 2011. to appear
2. F. Tango, L. Minin, R. Aras, and O. Pietquin. Automation Effects on Driver's Behaviour when integrating a PADAS and a Distraction Classifier. In *Proceedings of the Intenational Conference on Human-Computer Interfaces (HCI 2011)*, Orlando (FL, USA), July 2011. 10 pages - Invited Paper
3. L. Daubigney and O. Pietquin. Single-pass P300 detection with Kalman filtering and SVMs. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2011)*, Bruges (Belgium), April 2011. 6 pages
4. J. Fix, M. Geist, O. Pietquin, and H. Frezza-Buet. Dynamic Neural Field Optimization using the Unscented Kalman Filter. In *Proceedings of the IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB 2011)*, Paris (France), April 2011. 8 pages
5. M. Geist and O. Pietquin. Parametric Value Function Approximation : a Unified View. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2011)*, Paris (France), April 2011. 8 pages
6. O. Pietquin, F. Tango, and R. Aras. Batch Reinforcement Learning for Optimizing Longitudinal Driving Assistance Strategies. In *Proceedings of the IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS 2011)*, April 2011. 8 pages
7. M. Geist and O. Pietquin. Managing Uncertainty within the KTD Framework. In *Proceedings of the Workshop on Active Learning and Experimental Design (AL&E collocated with AISTAT 2010)*, Journal of Machine Learning Research Conference and Workshop Proceedings, Sardinia (Italy), 2011. 12 pages - to appear
8. M. Geist and O. Pietquin. Eligibility Traces through Colored Noises. In *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia), October 2010. 8 pages (best paper award)
9. M. Geist and O. Pietquin. Statistically Linearized Least-Squares Temporal Differences. In *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia), October 2010. IEEE. 8 pages
10. M. Geist and O. Pietquin. Revisiting natural actor-critics with value function approximation. In *Proceedings of the 22nd Benelux Conference on Artificial Intelligence (BNAIC 2010)*, Luxembourg (Luxembourg), October 2010. to appear
11. S. Chandramohan and O. Pietquin. User and noise adaptive dialogue management using hybrid system actions. In G. G. Lee, J. Mariani, W. Minker, and S. Nakamura, editors, *Spoken Dialogue Systems for Ambient Environments*, volume 6392 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 13–24, Gotemba, Shizuoka (Japan), October 2010. Springer Verlag, Heidelberg - Berlin. Proceedings of the International Workshop on Spoken Dialogue Systems (IWSDS 2010)

12. S. Rossignol, O. Pietquin, and M. Ianotto. Grounding simulation in spoken dialog systems with bayesian networks. In G. G. Lee, J. Mariani, W. Minker, and S. Nakamura, editors, *Spoken Dialogue Systems for Ambient Environments*, volume 6392 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 110–121, Gotemba, Shizuoka (Japan), October 2010. Springer-Verlag, Heidelberg-Berlin. Proceedings of the 2nd International Workshop on Spoken Dialogue Systems (IWSDS 2010)

13. S. Rossignol and O. Pietquin. Single-speaker/multi-speaker co-channel speech classification. In *Proceedings of the International Conference on Speech Communication and Technologies (Interspeech 2010)*, pages 2322–2325, Makuhari (Japan), September 2010. ISCA

14. S. Chandramohan, M. Geist, and O. Pietquin. Optimizing spoken dialogue management with fitted value iteration. In *Proceedings of the International Conference on Speech Communication and Technologies (Interspeech 2010)*, pages 86–89, Makuhari (Japan), September 2010. ISCA

15. S. Chandramohan, M. Geist, and O. Pietquin. Sparse approximate dynamic programming for dialog management. In *Proceedings of the 11th SIGDial Conference on Discourse and Dialogue*, pages 107–115, Tokyo (Japan), September 2010. ACL

16. M. Geist and O. Pietquin. Revisiting natural actor-critics with value function approximation. In V. Torra, Y. Narukawa, and M. Daumas, editors, *Proceedings of 7th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2010)*, volume 6408 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 207–218, Perpinya (France), October 2010. Springer Verlag - Heidelberg Berlin

17. M. Geist and O. Pietquin. Statistically Linearized Recursive Least Squares. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010)*, Kittilä (Finland), August-September 2010. 5 pages, to appear

18. B. Chevaillier, J.-L. Collette, D. Mandry, M. Claudon, and O. Pietquin. Objective assessment of renal DCE-MRI image segmentation. In *Proceedings of the European Signal Processing Conference (EUSIPCO 2010)*, Aalborg (Danmark), August 2010. Eurasip. 1214-1218

19. F. Tango, M. Alonso, M. H. Vega, R. Aras, and O. Pietquin. A Reinforcement Learning approach for designing and optimizing interaction strategies for a Human-Machine Interface of a Partially Autonomous Driver Assistance System. In *Proceedings of the Workshop on Human Modelling in Assisted Transportation (HMAT 2010)*, Belgirate (Italy), June 2010. Springer Verlag, Heidelberg - Berlin. to appear

20. F. Tango, R. Aras, and O. Pietquin. Learning Optimal Control Strategies from Interactions for a Partially Autonomous Driver Assistance System. In *Proceedings of the Workshop on Human Modelling in Assisted Transportation (HMAT 2010)*, Belgirate (Italy), June 2010. Springer Verlag, Heidelberg - Berlin. to appear

21. B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Beat to Beat Management of Heart Cycle Changes for Black Blood Imaging in End-Systolic Rest. In *Proceedings of the ISMRM workshop on Current Concepts of Motion Correction for MRI and MRS*, Kitzbuhel (Austria), February 2010

22. J.-L. Gutzwiller, H. Frezza-Buet, and O. Pietquin. Online Speaker Diarization with a Size-Monitored Growing Neural Gas Algorithm. In M. Verleysen, editor, *Proceedings of the 18 th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 505–510, Bruges (Belgium), April 2010

23. J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Bayesian Framework for Artifact Reduction on ECG in MRI. In *Proceedings of the 35th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pages 489–492, Dallas (TX, USA), March 2010

24. M. Saidi, O. Pietquin, and R. André-Obrecht. Application of the EMD decomposition to discriminate nasalized vs. vowels phones in French. In *Proceedings of the International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2010)*, pages 128–132, Innsbruck (Austria), February 2010. ACTA Press

25. O. Pietquin, S. Rossignol, and M. Ianotto. Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proceedings of the 1rst International Workshop on Spoken Dialogue Systems Technology (IWSDS 2009)*, Irsee (Germany), December 2009. 4 pages

26. M. Geist, O. Pietquin, and G. Fricout. Tracking in Reinforcement Learning. In *Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009)*, volume 5863, Part I, pages 502–511, Bangkok (Thailand), December 2009. Springer LNCS. ENNS best student paper award

27. M. Saidi, O. Pietquin, and R. André-Obrecht. EMD decomposition to discriminate nasal vs. oral vowels in French . In *Proceedings of the 13th International conference on Speech and Computer (SPECOM 2009)*, St Petersburg (Russia), June 2009. 5 pages

28. M. Lohezic, B. Fernandez, J. Oster, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Free breathing black-blood systolic imaging using heart rate prediction and motion compensated reconstruction. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009

29. B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Adaptive Trigger Delay Using a Predictive Model Applied to Black Blood Fast Spin Echo Cardiac Imaging in Systole. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009

30. J. Oster, B. Fernandez, M. Lohezic, D. Mandry, P.-A. Vuissoz, O. Pietquin, and J. Felblinger. Adaptive Heart Rate Prediction for Black-Blood Systolic Iimaging. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009

31. J. Oster, J. Pascal, O. Pietquin, M. Kraemer, J.-P. Blondé, and J. Felblinger. Real-Time Adaptive suppression of MR gradient Artifacts on Electorcardiograms using a new 3D Hall Probe. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009

32. M. Geist, O. Pietquin, and G. Fricout. Kernelizing Vector Quantization Algorithms. In M. Verleysen, editor, *Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN 09)*, pages 541–546, Bruges (Belgium), April 2009

33. J. Oster, O. Pietquin, R. Abächerli, M. Kraemer, and J. Felblinger. A Specific QRS Detector for Electrocardiography during MRI : Using Waveltets and Local Regularity Characterization. In *Proceedings of the 34th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pages 341–344, Taipei (Taiwan), April 2009

34. M. Geist, O. Pietquin, and G. Fricout. Kalman Temporal Differences : the deterministic case . In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009)*, pages 185–192, Nashville (TN, USA), April 2009

35. S. Rossignol and O. Pietquin. Precise Voicing Information Extraction in Speech Signals Using the Analytic Signal. In *Proceedings of the 8th IEEE Symposium on Signal Processing and Information Technology (ISSPIT 2008)*, pages 207–212, Sarajevo (Bosnia & Herzegovina), December 2008

36. M. Geist, O. Pietquin, and G. Fricout. Online Bayesian Kernel Regression from Nonlinear Mapping of Observations. In *Proceedings of the 18th IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2008)*, number a53, pages 309–314, Cancun (Mexico), October 2008

37. M. Claudon, D. Mandry, C. Pasquier, B. Chevaillier, J.-L. Collette, and O. Pietquin. Functional semi-automated segmentation of renal DCE-MRI sequences : preliminary results. In *Proceedings of the 15th Symposium of the European Society of Urogenital Radiology (ESUR 2008)*, Munich (Germany), September 2008

38. J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Independent Component Analysis based Artifact Reduction Method for ECG in MR. In *Proceedings of the European Society for Magnetic Resonance in Medicine and Biology congress (ESMRMB 08)*, Valencia (Spain), October 2008

39. D. Mandry, B. Chevaillier, J.-L. Collette, M.-A. Galloy, Y. Ponvianne, J. Felblinger, O. Pietquin, and M. Claudon. Functional semi-automated segmentation of renal DCE-MRI sequences using vector quantization. In *Proceedings of the European Society for Magnetic Resonance in Medicine and Biology congress (ESMRMB 08)*, Valencia (Spain), October 2008

40. M. Geist, O. Pietquin, and G. Fricout. A Sparse Nonlinear Bayesian Online Kernel Regression. In *Proceedings of the 2nd IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (AdvComp 2008)*, volume I, pages 199–204, Valencia (Spain), October 2008. (best paper award)

41. B. Chevaillier, D. Mandry, Y. Ponvianne, J.-L. Collette, M. Claudon, and O. Pietquin. Functional semi-automated segmentation of renal DCE-MRI sequences using a Growing Neural Gas algorithm. In *Proceedings of the 16th European Signal Processing Conference (EUSIPCO'08)*, page 5 pages (Proceedings on CDROM), Lausanne (Switzerland), August 2008

42. B. Chevaillier, Y. Ponvianne, J.-L. Collette, D. Mandry, M. Claudon, and O. Pietquin. Functional Semi-Automated Segmentation of Renal DCE-MRI Sequences. In *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 525–528, Las Vegas (NV, USA), April 2008

43. J. Oster, O. Pietquin, G. Bosser, and J. Felblinger. Adaptive RR Prediction for Cardiac MRI. In *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 513–516, Las Vegas (NV, USA), April 2008

44. V. Galtier, O. Pietquin, and S. Vialle. AdaBoost Parallelization on PC Clusters with Virtual Shared Memory for Fast Feature Selection. In *Proceedings of the 1st IEEE International Conference on Signal Processing and Communication*, pages 165–168, Dubai (United Arab Emirates), November 2007

45. O. Lemon and O. Pietquin. Machine Learning for Spoken Dialogue Systems. In *Proceedings of the 10th European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, Anvers (Belgium), August 2007

46. J. Oster, F. Odile, G. Bosser, O. Pietquin, C. Pasquier, P.-A. Vuissoz, and J. Felblinger. Adaptive Prediction of RR interval for online MR parameters changes. In *Proceedings of the Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM 2007)*, Berlin (Germany), May 2007

47. O. Pietquin. Learning to Ground in Spoken Dialogue Systems. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume IV, pages 165–168, Honolulu (Hawaii, USA), April 2007

48. O. Pietquin and T. Dutoit. Dynamic Bayesian Networks for NLU Simulation with Application to Dialog Optimal Strategy Learning. In *Proceedings of the 31st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume I, pages 49–52, Toulouse (France), May 2006

49. O. Pietquin. Consistent Goal-Directed User Model for Realistic Man-Machine Task-Oriented Spoken Dialogue Simulation. In *Proceedings of the 7th IEEE International Conference on Multimedia and Expo*, pages 425–428, Toronto (Canada), July 2006

50. O. Pietquin. A Probabilistic Description of Man-Machine Spoken Communication. In *Proceedings of the 5th IEEE International Conference on Multimedia and Expo (ICME 2005)*, pages 410–413, Amsterdam (The Netherlands), July 2005

51. O. Pietquin and R. Beaufort. Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning. In *Proceedings of the 9th European Conference on Speech Communication and Technologies (Interspeech/Eurospeech)*, pages 861–864, Lisbon (Portugal), September 2005. ISCA

52. M. Bagein, O. Pietquin, C. Ris, and G. Wilfart. Enabling Speech Based Access to Information Management Systems over Wireless Network. In *Proceedings of the 3rd workshop on Applications and Services in Wireless Networks (ASWN 2003)*, Berne (Switzerland), July 2003. 6 pages

53. M. Bagein, O. Pietquin, C. Ris, and G. Wilfart. An Architecture for Voice-Enabled Interfaces over Local Wireless Networks. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, Orlando, (USA, FL), July 2003. 6 pages

54. O. Pietquin and T. Dutoit. Aided Design of Finite-State Dialogue Management Systems. In *Proceedings of the 4th IEEE International Conference on Multimedia and Expo (ICME 2003)*, volume III, pages 545–548, Baltimore (USA, MA), July 2003

55. O. Pietquin and S. Renals. ASR System Modeling For Automatic Evaluation And Optimization of Dialogue Systems. In *Proceedings of the 27th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, volume I, pages 45–48, Orlando, (USA, FL), May 2002

Actes de conférences nationales avec comité de lecture

1. S. Rossignol, O. Pietquin, and M. Ianotto. Simulation du processus de croyance mutuelle de la compréhension dans le dialogue (grounding process) à l'aide des réseaux bayésiens. In *Actes des Journées d'Etude de la Parole (JEP 2010)*, Mons (Belgium), May 2010. 5 pages

2. M. Geist and O. Pietquin. Gestion de l'incertitude dans le cadre de l'approximation de la fonction de valeur pour l'apprentissage par renforcement. In *actes de la conférence francophone sur l'apprentissage automatique (CAP 2010)*, pages 101–112, Clermont-Ferrand (France), May 2010. PUG

3. M. Geist, O. Pietquin, and G. Fricout. Astuce du Noyau & Quantification Vectorielle. In *Actes du 17ème colloque sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'10)*, Caen (France), January 2010. 8 pages

4. B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Synchronisation adaptative utilisant un modèle prédictif : Applications à l'imagerie cardiaque par résonance magnétique en sang noir et en systole. In *Actes des Journées de Recherche en Imagerie et Technologies de la Santé (RITS 2009)*, Lille (France), mars 2009

5. D. Mandry, B. Chevaillier, Y. Ponvianne, J.-L. Collette, M.-A. Galloy, O. Pietquin, and M. Claudon. Segmentation fonctionnelle rénale semi-automatique par quantification vectorielle de séries dynamiques en IRM. In *Journées Françaises de Radiologie (JFR 2008)*, Paris (France), October 2008

6. M. Geist, O. Pietquin, and G. Fricout. Filtrage bayésien de la récompense. In *actes des Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2008)*, pages 113–122, Metz (France), June 2008

7. J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Méthode de réduction des artéfacts présents sur l'ECG basée sur l'analyse en composantes indépendantes. In *Actes du 12ème congrès du Groupe de Recherche sur les Applications du Magnétisme en Médecine (GRAMM'08)*, Lyon (France), mars 2008

8. O. Pietquin. Un cadre probabiliste pour l'optimisation des systèmes de dialogue. In *Proceedings of the 4rth International Conference : Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2007)*, Hammamet (Tunisia), March 2007. 8 pages

9. O. Pietquin. Réseau bayésien pour un modèle d'utilisateur et un module de compréhension pour l'optimisation des systèmes de dialogues. In *Actes de la Conférence Francophone sur le Traitement du Langage Naturel (TALN 2005)*, volume I, pages 481–486, Dourdan (France), June 2005

10. O. Pietquin. Une description probabiliste de la communication parlée entre homme et machine. In *Actes de la 16ème Conférence Francophone sur l'Interaction Homme-Machine (IHM 2004)*, pages 247–250, Namur (Belgique), August-September 2004

11. O. Pietquin. Environnement virtuel pour la simulation et l'apprentissage de stratégies de dialogue. In *Actes de la 15ème Conférence Francophone sur l'Interaction Homme-Machine (IHM 2003)*, Caen (France), November 2003. 4 pages

12. O. Pietquin and T. Dutoit. Modélisation d'un système de reconnaissance dans le cadre de l'evaluation et l'optimisation automatique des systèmes de dialogue. In *Actes des Journées d'Etude de la Parole, JEP 2002*, pages 281–284, Nancy (France), June 2002

Communications sans acte

1. L. Daubigney, M. Geist, and O. Pietquin. Apprentissage par renforcement pour la personnalisation d'un logiciel d'enseignement des langues. In *Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2011)*, Mons (Belgium), May 2011. 4 pages
2. B. Chevaillier, J.-L. Collette, D. Mandry, M. Claudon, and O. Pietquin. Segmentation fonctionnelle de séquences d'IRM rénales à rehaussement de contraste par quantification vectorielle. In *Colloque Recherche en Imagerie et Technologies pour la Santé (RITS 2011)*, Rennes (France), April 2011. 3 pages
3. S. Chague, J. D'Hose, J.-F. Goudou, B. Dorizzi, L. Giulieri, Q.-C. Pham, F. Sedes, M. Brut, D. Nicholson, and O. Pietquin. METHODEO : Méthodologie d'évaluation des algorithmes d'exploitation des enregistrements de la vidéoprotection. In *Workshop Interdisciplinaire sur la Sécurité Globale (WISG 2011)*, Troyes (France), January 2011. 6 pages
4. O. Pietquin. Batch reinforcement learning for spoken dialogue systems with sparse value function approximation. In *NIPS Workshop on Learning and Planning from Batch Time Series Data*, Vancouver (Canada), 2011
5. O. Pietquin and F. Tango. Batch reinforcement learning for optimizing driving assistance strategies. In *NIPS Workshop on Learning and Planning from Batch Time Series Data*, Vancouver (Canada), 2011
6. R. Aras and O. Pietquin. Optimal Average Reward Controllers For POMDPs. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, Besançon (France), June 2010. 6 pages
7. M. Geist and O. Pietquin. Statistically Linearized Least-Squares Temporal Differences. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, Besançon (France), June 2010. 8 pages
8. M. Geist and O. Pietquin. Revisiting natural actor-critics with value function approximation. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, Besançon (France), June 2010. 6 pages
9. M. Geist and O. Pietquin. Managing Uncertainty within Value Function Approximation in Reinforcement Learning. In *Active Learning and Experimental Design workshop (collocated with AISTATS 2010)*, Sardinia, Italy, 2010. 8 pages, oral presentation
10. B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Amélioration de lŠimagerie du ventricule droit en IRM cardiaque en ń sang noir ż par une méthode adaptative. In *Journée Claude Huriet*, Nancy (France), December 2009
11. J.-L. Collette and O. Pietquin. Localisation de Source Sonore par Goniometrie Acoustique pour la Détection de Chute. In *2ème colloque PARAChute*, Troyes (France), November 2009
12. M. Geist, O. Pietquin, and G. Fricout. Différences Temporelles de Kalman : le cas stochastique. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, Paris (France), June 2009. 13 pages
13. M. Geist, O. Pietquin, and G. Fricout. Différences Temporelles de Kalman. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, Paris (France), June 2009. 20 pages
14. M. Geist, O. Pietquin, and G. Fricout. Bayesian Reward Filtering. In *8th European Workshop on Reinforcement Learning (EWRL 2008)*, Lille (France), June 2008. 14 pages
15. M. Geist, O. Pietquin, and G. Fricout. Filtrage Bayésien de la Récompense. In *Journée NeuroInfo - Loria*, Nancy (France), July 2008

16. B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Synchronisation Adaptative Utilisant un Modèle Prédictif : Applications à lŠImagerie Cardiaque par Résonance Magnétique en Sang Noir et en Systole. Journée Claude Huriet, Nancy (France), December 2008
17. M. Geist, O. Pietquin, and G. Fricout. Kalman Temporal Differences : Uncertainty and Value Function Approximation. In *NIPS Workshop on Model Uncertainty and Risk in Reinforcement Learning*, Vancouver (Canada), December 2008
18. F. Gaspard, O. Pietquin, F. Sèdes, J.-F. Seignole, and J.-F. Sulzer. Architecture Générique de Stockage Multimedia Réparti avec Recherche et Indexation distribuées (Projet ITEA2 LINDO). In *Journée d'étude sur l'Analyse Vidéo pour le Renseignement et la Sécurité (AViRS 2008)*, Paris (France), April 2008. SEE

Brevets

1. J.-P. Morard, O. Pietquin, and S. Vialle. Method for the segmentation encoding of an image, July 2010. Patent n° WO/2010/072983
2. J.-P. Morard, S. Vialle, O. Pietquin, and V. Galtier. Procédé de diffusion de séquences de données vidéo par un serveur vers un terminal client, March 2009. Brevet n° FR2920933 (A1)
3. J.-P. Morard, S. Vialle, O. Pietquin, and V. Galtier. Method for broadcasting video data sequences by a server to a client terminal, March 2009. Patent n° WO/2009/034275
4. O. Pietquin and V. Philomin. Method of determining motion-related features and method of performing motion classification, November 2008. Patent n° WO2008139399
5. O. Pietquin. A method of recognizing a motion pattern of an object, April 2008. Patent n° EP1904951

Rapports techniques

1. S. Rossignol, S. Janarthanam, X. Liu, O. Pietquin, and M. Ianotto. User simulations of different types of Appointment Scheduling and Self-Help user. CLASSiC Project Deliverable 3.6, January 2011
2. S. Keizer, O. Pietquin, S. Rossignol, and S. Young. Agenda-based user simulations and EM training tools for Appointment Scheduling and TownInfo domains. CLASSiC Project Deliverable 3.4, October 2010
3. M. Geist and O. Pietquin. A Brief Survey of Parametric Value Function Approximation. Technical report, September 2010
4. O. Pietquin and H. Hastie. Metrics for the evaluation of user simulation. CLASSiC Project Deliverable 3.5, March 2010
5. O. Pietquin, S. Rossignol, M. Ianotto, and P. Crook. Probabilistic user simulations for training in the (PO)MDP framework including the simulation of grounding in TownInfo dialogues. CLASSiC Project Deliverable 3.2, October 2009
6. O. Lemon, O. Pietquin, H. Frezza-Buet, V. Rieser, X. Liu, P. Bretier, S. Young, and J. Henderson. Shared Context Model (XML Schema). CLASSiC Project Deliverable 3.1, February 2009
7. O. Pietquin, H. Frezza-Buet, and P. Crook. New frameworks for generalization, with implementation in DIPPER ISU DM system. CLASSiC Project Deliverable 1.2, October 2009
8. S. Laborie, F. Sedes, J.-F. Sulzer, O. Pietquin, N. Allezard, R. Pinchuk, J.-P. Guignard, P. Moreira, C. Fernandez, S. Moens, J.-F. Seignole, and D. Milan. Proposed Content Indexation Agent Software. LINDO Deliverable D3.1, 2008
9. J.-F. Sulzer, J.-P. Guignard, J.-F. Seignole, F. Sedes, O. Pietquin, R. Pinchuk, S. Moens, D. Milan, N. Allezard, P. Moreira, C. Fernandez, and A.-M. Manzat. Preliminary Requirement Specification (generic and applicative). LINDO deliverable D2.1, 2008

Thèses

1. O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. PhD thesis, Faculté Polytechnique de Mons, TCTS Lab (Belgique), April 2004
2. O. Pietquin. Algorithme de recouvrement à haute vitesse de données mémorisées sur support optique. Master's thesis, Faculté Polytechnique de Mons (FPMs), Belgium, 1999

Table des matières

Table des figures

Liste des tableaux

Liste des acronymes

AR Apprentissage par Renforcement
ARI Apprentissage par Renforcement Inverse
AR Auto-Régressif
ARMA *Auto-Regressive Moving Average*
BAGARRE *BAyesian Gradient ARtifact REduction*
CHU Centre Hospitalier Universitaire
DIR *Double Inversion Recovery*
DIR-FSE *Double Inversion Recovery Fast Spin Echo*
DRC Dynamique à Rehaussement de Contraste
ECG électrocardiogramme
EKF Filtre de Kalman Etendu ou *Extended Kalman Filter*
ESV Extra-Systole Ventriculaire
GNG *Growing Neural Gas*
GPTD *Gaussian Process Temporal Differences*
IDCT *Inverse Discrete Cosine Transform*
IRM Imagerie par Résonance Magnétique
KTD différences temporelles de Kalman ou *Kalman Temporal Differences*
LSPI *Least Square Policy Iteration*
LSTD *Least Square Temporal Differences*
LMS *Least Mean Square*
MA *Moving Average*
MFCC *MEL Cepstrum Frequency Coefficients*
MHD MagnetoHydroDynamique
NEE *Noise Energy Evolution*
NMSE *Normalised Mean Squared Error*
PADAS *Partially Autonomous Driving Assistance System*
PDM Processus Décisionnel de Markov
PDMPO Processus Décisionnel de Markov Partiellement Observable
PMIR *Physiological Multi Input Regression*
PSNRE *Pseudo Signal to Noise Ration Evolution*
RBF *Radial Basis Functions*
RSA *Respiratory Sinus Arrythmia*
SDS *Spoken Dialogue Systems*
SEE *Signal Energy Evolution*

SPKF *Sigma Point Kalman Filter*

SVM Machines à Vecteurs Supports ou *Support Vector Machines*

TD Différence Temporelle ou *Temporal Difference*

UKF Filtre de Kalman non-parfumé ou *Unscented Kalman Filter*

VCG VectoCardioGramme

XKTD *eXtended Kalman Temporal Differences*

Chapitre 1

Introduction

Dans ce chapitre, après avoir présenté le fil conducteur des travaux qui ont été réalisés et la structure du manuscrit, sont tout d'abord décrits les travaux de thèse qui ont été à la base des contributions développées dans le reste du document. Ensuite, ces contributions sont listées et mises en relation afin de dégager la cohérence des travaux réalisés.

1.1 Motivation

Les travaux présentés dans ce manuscrit ont pour but le développement de méthodes de prise de décisions optimales (statiques ou séquentielles) ou de traitement de signaux par des méthodes d'apprentissage automatique. Ils répondent cependant à un certain nombre de contraintes qui ont été imposées par l'objectif de prendre en compte la présence de l'humain dans la boucle de décision ou de traitement ou même comme générateur des signaux que l'on analyse. La présence de l'humain rend la nature des données que l'on traite assez imprévisible et, à une même situation en apparence, des décisions différentes peuvent être prises. Dans les domaines de l'apprentissage statistique, du contrôle optimal ou du traitement du signal ceci se traduit par la nécessité de gérer l'incertain, de traiter le caractère stochastique du système ainsi que la non-stationnarité de celui-ci. Ainsi, les décisions que l'on considère optimales peuvent dépendre de manière aléatoire de la situation mais de plus, cette dépendance peut varier avec le temps. Des méthodes d'optimisation convergeant vers une solution globale ne sont donc pas adaptées mais des méthodes permettant d'apprendre au fil de l'eau et de poursuivre l'évolution de la solution optimale seront préférées.

Par ailleurs, dans le cas où les décisions résultent en une action sur le monde extérieur, il est nécessaire de quantifier le risque pris en accomplissant ces actions, particulièrement si ces actions doivent avoir un impact sur l'humain. Ceci passe par une estimation de l'incertitude sur le résultat des actions possibles et la sélection en conséquence de ces actions. Une autre implication est qu'il ne sera pas toujours envisageable de tester toutes les actions possibles pour en estimer les effets puisque ces actions peuvent ne pas être acceptables pour l'humain. Ainsi, il faudra apprendre à partir d'exemples de situations imposées ce qui se traduit par une phase d'inférence utilisant les informations observables pour en déduire des conséquences sur des situations que l'on ne peut observer.

Les travaux exposés dans ce manuscrit apportent des contributions théoriques permettant de tenir compte de ces contraintes et des applications à des problèmes concrets imposant ces contraintes seront exposées.

1.2 Résumé des travaux réalisés avant l'obtention de la thèse

Le travail de thèse [173] était déjà pluridisciplinaire et se situait au confluent du traitement automatique de la parole et du contrôle optimal. Il a été l'occasion de s'intéresser conjointement aux théories du traitement du signal, du langage et de l'apprentissage automatique. Son thème était essentiellement l'optimisation de stratégies pour les systèmes de dialogue homme-machine par le biais d'algorithmes

d'*apprentissage par renforcement* [242]. Les systèmes développés ont pour but d'aider l'utilisateur humain à accomplir une tâche comme obtenir une information sur un lieu, interroger une base de données, faire une réservation d'hôtel *etc.* Les algorithmes standards d'apprentissage par renforcement étant très gourmands en données, une grande partie de la thèse a été dédiée à modéliser le dialogue homme-machine, et par conséquent le comportement d'un utilisateur, de manière probabiliste [191, 175], pour pouvoir simuler l'interaction [191, 177] et les erreurs introduites pas les différents modules constituant un systèmes de dialogue [200, 186, 190] ; ceci afin de proposer des outils de conception semi-automatisés de systèmes de dialogue vocaux [189, 10, 11].

Notons qu'avant de travailler sur ce thème, des travaux avaient été menés dans le domaine de la segmentation en locuteurs d'un flux audio [187] dans le cadre du projet européen THISL en collaboration avec l'Université de Sheffield (UK) et la BBC notamment.

1.3 Contributions scientifiques

Comme indiqué dans la section précédente, le problème majeur que pose l'utilisation d'algorithmes standard d'apprentissage par renforcement est la nécessité d'un volume conséquent de données ou d'interactions réelles pour obtenir la convergence. Ces algorithmes ont aussi d'autres défauts et particulièrement le fait qu'ils passent mal à l'échelle. Ainsi, pour des problèmes de taille réelle, ces algorithmes ne peuvent pas donner de réponse satisfaisante. Par ailleurs, il était nécessaire de tenir compte de la présence de l'humain dans l'environnement que l'on cherche à contrôler de manière optimale. Ainsi, le système apprenant ne peut pas se permettre d'accomplir des actions totalement aléatoires (au prix de voir l'utilisateur se désintéresser du système) voire dangereuses (c'est le cas dans certaines applications visées dans ce document comme la conduite assistée d'un véhicule). Aussi, l'être humain rend l'environnement dans lequel il évolue hautement stochastique et non-stationnaire. La thèse avait permis de mettre en évidence ces limitations dans un domaine précis qu'était celui des systèmes de dialogue homme-machine, néanmoins cela reste un problème général auquel il a été jugé bon de s'attaquer. Ce problème général est exposé au Chapitre 2. Ainsi, une première contribution importante exposée au Chapitre 3 a été la mise au point d'algorithmes d'apprentissage par renforcement *efficaces* permettant le traitement de problèmes de *très grandes dimensions*. Ce fut particulièrement le thème de la thèse de Matthieu GEIST [71], réalisée en collaboration avec ArcelorMittal et l'INRIA Nancy Grand-Est et qui a donné lieu à plusieurs publications dans le domaine de l'apprentissage par renforcement [86, 95, 97, 94, 99, 74, 78, 75]. Ces travaux ont trouvé des applications dans le domaine du dialogue homme-machine bien sur [24, 23](dans le cadre des projets européens CLASSiC [1] (FP7 ICT) et ALLEGRO [2] (INTERREG) notamment) mais aussi de l'aide à la gestion de l'énergie dans l'industrie métallurgique [71] ou de l'assistance à la conduite automobile dans le cadre du projet européen ISi-PADAS [3] (FP7 SST) [245, 244]. Ces applications seront décrites au Chapitre 4.

Cette thèse a aussi apporté des contributions dans le domaine de l'apprentissage automatique en général et en particulier de l'estimation de paramètres en ligne pour des systèmes non-stationnaires [85, 91, 96, 82]. C'est la thématique exposée dans le Chapitre5. Dans ce chapitre, des méthodes d'estimation de paramètres par des approches bayésiennes dans le cas particulier du traitement de signaux électrocardiographiques en environnement contraint (IRM) sont développées. Il s'agit plus particulièrement des travaux de thèse de Julien OSTER [154] réalisés en collaboration avec l'équipe IADI (INSERM) qui ont débouché aussi sur plusieurs publications sur ce thème [156, 160, 155, 55, 131, 163, 59, 164, 58]. Cette thèse a, de plus, permis d'étudier l'extraction de paramètres pertinents du signal considéré [158] ainsi qu'un traitement statistique de celui-ci [161, 157, 159] en vue particulièrement de réaliser une classification de segments issus du signal.

Dans ce même ordre d'idée, la thèse de Béatrice CHEVAILLIER [28] s'est attaquée à la modélisation statistique d'un signal pour sa segmentation automatique. C'est en particulier la quantification vectorielle, comme développé dans le Chapitre 6, qui a été étudiée et appliquée à un problème de segmentation d'images d'IRM fonctionnelles rénales. Cette thèse a donné lieu à plusieurs publications [34, 33, 133,

1. http ://www.classic-project.org
2. http ://www.allegro-project.org
3. http ://www.isi-padas.eu

37, 32, 29]. Des méthodes similaires ont été utilisées pour la segmentation d'autres types de signaux, et plus spécifiquement la segmentation en locuteurs de signaux acoustiques au fil de l'eau [187, 103] dans le cadre notamment du projet LINDO[4] (ITEA 2). Là encore, le caractère aléatoire et non-stationnaire introduit par la variabilité inhérente à la présence de l'humain dans le système analysé devait être pris en compte.

Durant les travaux présentés ci-dessus, c'est la généricité des solutions qui a guidé nos recherches. C'est la raison pour laquelle les applications ont été nombreuses mais c'est aussi pourquoi les perspectives sont nombreuses tant dans le domaine de nouvelles applications que dans celui de la théorie, afin d'améliorer encore cette généricité par exemple. Ces perspectives seront développées dans la Partie III.

1.4 Organisation du document

Ce manuscrit est structuré en quatre parties. Les deux premières parties ont pour but de présenter les contributions scientifiques développées depuis la thèse dans les domaines du contrôle optimal (Partie I) et du traitement du signal (Partie II) grâce à des méthodes d'apprentissage automatique. La Partie **??** s'attachera à décrire l'environnement pédagogique et scientifique dans lequel les recherches décrites ont été réalisées. Ainsi, les activités de dissémination scientifique comme la participation à des projets, la contribution à l'animation scientifique ou les responsabilités administratives seront aussi exposées. Les activités pédagogiques développées autour de ces thématiques seront décrites. Enfin, les suites prévues à ces activités seront développées dans le projet de recherche exposé dans la Partie III.

4. http ://www.lindo-itea.eu/

Première partie

Contrôle Optimal

Chapitre 2

Apprentissage par renforcement

Dans un large domaine d'applications, le problème de la commande optimale de systèmes dynamiques et stochastiques se pose. On peut citer la commande de procédés, la gestion optimale de stocks ou même encore les interfaces homme-machine [172]. Il s'agit alors d'établir une correspondance entre les états d'un système et les actions à lui appliquer pour qu'il fonctionne de manière optimale.

L'automatique apporte une famille de solutions à ce problème en passant souvent par une modélisation formelle du système à contrôler. Néanmoins, cette modélisation n'est pas toujours simple, voire possible. De plus, des modélisations exactes mais trop complexes ne permettent pas toujours de dériver une commande optimale. Dans ce type de cas, il peut être intéressant d'apprendre cette commande optimale par interaction avec le système à contrôler et en cherchant à minimiser un critère de coût (ou de manière équivalente à maximiser un critère de gain). C'est le principe de l'*Apprentissage par Renforcement* (AR) [242, 205].

Dans cette première partie nous nous intéressons particulièrement à ce type d'apprentissage. Dans le paradigme de l'AR, un *agent* artificiel interagit avec un *environnement* (ou système) dans le but de découvrir *en ligne* sa commande optimale (que l'on appellera *politique optimale*) du point de vue d'un critère d'optimisation numérique (voir figure 2.1). L'environnement est considéré comme se trouvant dans un *état* précis à chaque instant, cet état peut être observé directement ou partiellement par l'agent, et la commande optimale consiste à choisir l'*action* idéale (ou une distribution de probabilité sur les actions possibles) à associer à chacun des états dans lesquels peut se trouver l'environnement. A chaque instant, l'agent accomplit donc une action qui influence l'état de l'environnement qui transite de manière déterministe ou stochastique vers un nouvel état. Cette influence est perçue par l'agent qui observe le changement d'état du système et obtient un *signal de renforcement* (ou de *récompense*) immédiat après chaque action. C'est, en moyenne, le cumul (éventuellement pondéré) de ces récompenses immédiates tout au long de l'interaction qui tiendra lieu de critère à maximiser.

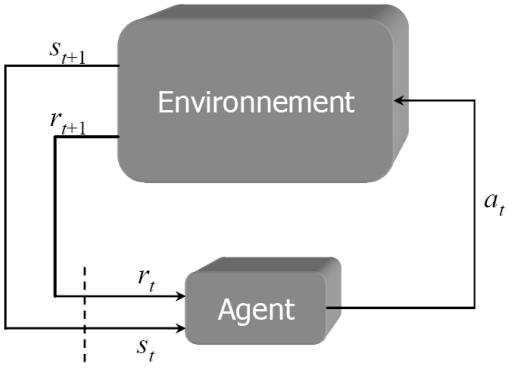


FIGURE 2.1 – Principe de l'AR.

Il ne s'agit donc pas d'optimiser chacune des actions individuellement mais bien la séquence d'actions dans son ensemble ou plus précisément la stratégie qui associe une action (ou une distribution sur les actions) à chaque état dans le but de visiter une séquence d'états désirés. Dans les cas les plus intéressants, les actions influencent un avenir plus ou moins proche en ne rendant possibles que certains chemins dans l'espace d'états du système et gênèrent donc des récompenses retardées dont il faut pouvoir tenir compte. L'exemple le plus courant est celui de la partie d'échecs où un joueur (l'agent) peut concéder la perte de sa reine si ceci lui assure le gain de la partie dans les coups à venir. L'apprentissage en ligne et la prise en compte de récompenses retardées sont parmi les caractéristiques les plus prometteuses et les plus difficiles à appréhender de la discipline. De plus, le système pouvant être stochastique, il s'agit d'optimiser la prise de décision dans l'incertain puisque les résultats des actions ne sont potentiellement pas déterministes.

Une difficulté majeure supplémentaire se pose en formulant le problème de la découverte de commande optimale de la sorte : celui de l'équilibre entre la gestion de la connaissance acquise par le passé et l'incertitude qu'il reste sur ces connaissances. En effet, après un certain nombre d'interactions avec l'environnement, l'agent peut choisir d'accomplir une action optimale du point de vue de la connaissance du système qu'il a déjà acquise. Néanmoins, si sa connaissance n'est pas parfaite, cette action peut être sous-optimale du point de vue de la commande du système. L'agent peut donc aussi choisir une action qui améliore sa connaissance du système mais qui parait sous-optimale. Ce problème est connu sous le terme de "*dilemme entre exploitation et exploration*". C'est évidemment un problème important lorsque l'humain fait partie de l'environnement à contrôler puisqu'il est important de ne pas accomplir des actions exploratoires trop éloignées de l'optimale ou même dangereuses.

A cela peut s'ajouter l'*observabilité partielle* de l'état du système par l'agent. En effet, dans des conditions réelles, il n'est pas toujours simple d'avoir un accès direct à l'état du système, mais on dispose plutôt d'une série d'informations fournies par des capteurs plus ou moins précis. De plus, ces capteurs peuvent mesurer de manière indirecte les grandeurs d'intérêt (comme une mesure de température déduite d'un spectromètre par exemple). Il faut alors prendre des décisions à partir de séquences d'observations et plus à partir de la connaissance de l'état.

La notion d'AR définit donc une classe de problèmes ayant un certain nombre de caractéristiques et laisse la porte ouverte à une large panoplie de solutions algorithmiques faisant toujours l'objet de recherches intensives dans le monde l'apprentissage automatique et de l'intelligence artificielle. L'AR est particulièrement intéressant dans le cas où l'humain fait partie de l'environnement pour plusieurs raisons. Tout d'abord, l'humain rend particulièrement imprévisible les réactions de l'environnement et toute tentative de dénombrer ou même de modéliser l'intégralité des séquences d'interactions possibles serait veine. Par ailleurs, si l'humain peut souvent donner un avis (favorable ou défavorable) sur les performances d'un système, il lui est souvent difficile d'exprimer ce qui aurait pu rendre l'interaction meilleure (donc l'apprentissage supervisé est impossible mais un signal de renforcement peut être obtenu). C'est aussi un paradigme générique d'apprentissage qui requiert un minimum de travail (essentiellement d'ordre technique et de manipulation de données, mais pas fondamental) pour transposer son application d'un domaine à un autre. Enfin, dans le paradigme général de l'AR, il n'est pas nécessairement supposé que le système est stationnaire (c'est-à-dire que le comportement reste identique avec le temps). Ainsi, les variabilités inévitables entre les comportements de différentes personnes, ou même d'une personne identique à différents instants, pourront être prises en compte.

2.1 Formalisme et algorithmes usuels

Dans la grande majorité des cas, le problème de l'AR est placé dans le cadre formel des Processus Décisionnels de Markov (PDM) [14, 205, 234]. Un PDM est un tuple $\{S, A, P, R, \gamma\}$ où S est l'espace d'état, A est l'espace d'action, $P : s, a \in S \times A \rightarrow p(.|s, a) \in \mathcal{P}(S)$ est une famille de probabilités de transitions, $R : S \times A \times S \rightarrow \mathbb{R}$ est la fonction (bornée) de récompense et γ le facteur d'actualisation. Les probabilités sont donc supposées Markov d'ordre 1, c'est à dire que la transition d'un état s à un autre état s' étant donné une action a n'est conditionnée que par l'action a et l'état s dans lequel le système se trouve et pas de l'historique des couples état-action passés. Une politique π associe à chaque état une distribution sur les actions : $\pi : s \in S \rightarrow \pi(.|s) \in \mathcal{P}(A)$. On qualifie alors la qualité d'une politique par

sa fonction de valeur définie par :

$$V^{\pi}(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi],$$

où r_i est la récompense immédiate observée au pas de temps (ou date) i, la moyenne étant faite sur toutes les trajectoires possibles, sachant que l'agent démarre dans l'état s et suit la politique π par la suite. La Q-fonction (ou fonction de valeur sur les couples état-action ou encore fonction de qualité) ajoute un degré de liberté supplémentaire sur le choix de la première action et est définie par :

$$Q^{\pi}(s,a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi].$$

L'AR a pour objectif de déterminer (à partir d'interactions) une politique optimale π^*, c'est-à-dire une politique qui maximise la fonction de valeur pour chaque état :

$$\pi^* = \underset{\pi}{\operatorname{argmax}}(V^{\pi}).$$

Il peut exister plusieurs politiques optimales. Néanmoins, elles partagent toutes la même fonction de valeur (que l'on notera $V^*(s)$) ainsi que la même Q-fonction (que l'on notera $Q^*(s,a)$). Il est clair que la connaissance des probabilités de transitions et de la fonction de récompense sont nécessaires pour déduire une politique de la fonction de valeur V^* alors que cette connaissance n'est pas nécessaire si on cherche à extraire la politique optimal de la fonction Q^*. En effet, on a

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, E_{s'|s,a}[V^*(s')],$$

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, Q^*(s,a).$$

Ainsi, le problème de l'AR peut se résumer à l'estimation de ces fonctions pour en déduire une politique optimale. Grâce à la propriété de Markov, il est possible de réécrire les définitions des fonctions V et Q. Ainsi, nous avons :

$$V^{\pi}(s) = E_{s',a|\pi,s}[R(s,a,s') + \gamma V^{\pi}(s')], \tag{2.1}$$

$$Q^{\pi}(s,a) = E_{s',a'|\pi,s,a}[R(s,a,s') + \gamma Q^{\pi}(s',a')]. \tag{2.2}$$

Les équations (2.1) et (2.2) sont appelées *équations d'évaluation* de Bellman. Si tous les éléments du PDM sont connus, il existe alors deux schémas parmi d'autres qui mènent à l'obtention d'une politique optimale.

Premièrement, l'algorithme d'itération sur les politiques consiste à apprendre la fonction de valeur d'une politique donnée, puis à améliorer cette politique, la nouvelle étant *gloutonne* par rapport à la fonction de valeur apprise. Une politique gloutonne est celle qui associe à un état s l'action a qui maximise $Q^{\pi}(s,a)$ (ou $E_{s'|s,a}[V^{\pi}(s')]$ où s' est l'état pris par le système au pas de temps suivant). Cela suppose de résoudre une des équations d'évaluation de Bellman (2.1) ou (2.2). La politique gloutonne $\pi_g(s)$ est donc :

$$\pi_g(s) = \underset{a}{\operatorname{argmax}}\, E_{s'|s,a}[V(s')], \tag{2.3}$$

$$\pi_g(s) = \underset{a}{\operatorname{argmax}}\, Q(s,a). \tag{2.4}$$

En itérant les phases d'évaluation et d'amélioration de la politique, cet algorithme converge vers la politique optimale. En effet, une fois que la politique devient stable, une politique optimale est trouvée puisqu'une politique gloutonne par rapport à sa propre fonction de valeur est optimale.

Le second schéma, appelé iteration sur les valeurs, estime directement la fonction de valeur optimale $V^*(s)$ ou $Q^*(s,a)$ et en déduit une politique optimale. Il nécessite de résoudre l'*équation d'optimalité* de Bellman qui s'obtient aussi grâce à la propriété de Markov :

$$Q^*(s,a) = E_{s'|s,a}[R(s,a,s') + \gamma \max_{b \in A} Q^*(s',b)]. \tag{2.5}$$

Ces deux méthodes nécessitent la résolution de systèmes d'équations dans lesquels interviennent les probabilités de transition et la fonction de récompense. Elle sont classées parmi les méthodes de programmation dynamique. Si les paramètres du PDM ne sont pas connus, on se retrouve face au problème plus général de l'AR. On suppose alors souvent que l'on cherche une représentation θ d'une fonction de valeur (V ou Q) et l'on vise à apprendre en ligne la représentation idéale pour en déduire une politique. Si les espaces d'état S et d'action A sont finis de cardinalité suffisamment faible, une représentation exacte de la fonction de valeur est possible, et θ est alors un vecteur avec autant de composantes qu'il y a d'états (ou de couples état-action pour la Q-fonction). C'est une représentation dite "tabulaire". Si ces espaces sont de trop grande taille, une approximation $\hat{V}_\theta(s)$ est nécessaire. Un choix classique en AR est la paramétrisation linéaire, pour laquelle la fonction de valeur est approchée comme suit :

$$\hat{V}_\theta(s) = \sum_{j=1}^{p} w_j \phi_j(s) = \theta^T \phi(s), \tag{2.6}$$

où $(\phi_j)_{1 \leq j \leq p}$ est un ensemble de fonctions de base que l'on regroupe dans un vecteur $\phi(s)$, qui doivent être définies à l'avance, et les paramètres sont les poids w_j :

$$\theta = [w_1, \ldots, w_p]^T. \tag{2.7}$$

Beaucoup d'algorithmes d'approximation de la fonction de valeur nécessitent une telle représentation pour assurer la convergence [231], ou même pour être applicable [20]. D'autres représentations sont possibles, par exemple un réseau de neurones pour lequel θ est composé des poids des connexions synaptiques associées.

Il existe plusieurs méthodes pour résoudre le problème et parmi elles, les méthodes dites de Différences Temporelles ou *Temporal Differences* (TD) qui sont des méthodes de résolution en ligne du problème. Elles forment une classe d'algorithmes qui consistent à corriger la représentation de la fonction de valeur (ou de qualité) selon l'*erreur de différence temporelle* (l'erreur TD) faite sur cette dernière. Cette erreur peut être définie comme la différence entre le membre de droite et le membre de gauche d'une des équations de Belman (2.1), (2.2) ou (2.5) comme expliqué plus tard.

La plupart de ces approches peuvent s'écrire de la façon générique suivante :

$$\theta_i = \theta_{i-1} + K_i \delta_i. \tag{2.8}$$

Dans cette expression, θ_{i-1} est l'ancienne représentation de la fonction de valeur, θ_i est cette même représentation mise à jour selon la dernière transition observée, δ_i est l'erreur de différence temporelle et K_i est un gain indiquant dans quelle direction la représentation de la fonction de valeur doit être corrigée.

Dans l'équation (2.8), le terme δ_i est l'*erreur de différence temporelle*. Supposons qu'au pas de temps i la transition $(s_i, a_i, r_i, s_{i+1}, a_{i+1})$ soit observée. Pour les algorithmes d'AR de type TD, c'est-à-dire les algorithmes qui visent l'évaluation de la fonction de valeur pour une politique donnée π, l'erreur TD est :

$$\delta_i = r_i + \gamma \hat{V}_{\theta_{i-1}}(s_{i+1}) - \hat{V}_{\theta_{i-1}}(s_i). \tag{2.9}$$

Pour les algorithmes de type SARSA [242], c'est-à-dire les algorithmes qui ont pour but d'évaluer la fonction de qualité d'une politique donnée π, et étant donné l'approximation $\hat{Q}_{\theta_{i-1}}$ de la Q-fonction, l'erreur TD est :

$$\delta_i = r_i + \gamma \hat{Q}_{\theta_{i-1}}(s_{i+1}, a_{i+1}) - \hat{Q}_{\theta_{i-1}}(s_i, a_i). \tag{2.10}$$

Enfin, pour les algorithmes de type Q-learning [253], c'est-à-dire les algorithmes dont l'objectif est de calculer la Q-fonction optimale, l'erreur TD est de la forme suivante :

$$\delta_i = r_i + \gamma \max_{b \in A} \hat{Q}_{\theta_{i-1}}(s_{i+1}, b) - \hat{Q}_{\theta_{i-1}}(s_i, a_i). \tag{2.11}$$

Le type de différence temporelle utilisé est directement lié au type d'équation de Bellman à résoudre (équation d'évaluation pour (2.9) et (2.10), équation d'optimalité pour (2.11)), et donc si l'algorithme intervient dans une méthode de de type itération de la politique ou de la valeur. Le terme K_i est un gain spécifique à chaque méthode. Nous ne décrirons pas ici ces méthodes mais un état de l'art plus complet conservant le point de vue développé jusqu'ici peut être trouvé dans [75, 72] notamment.

2.2 Point de vue alternatif

La section précédente présentait la vision standard de l'AR dans le cadre des PDM. Dans cette vision de l'AR, l'environnement est modélisé par un système dynamique stochastique constitué d'états et contrôlé par un agent cherchant à maximiser une fonction cumulante des récompenses r_i sur le long terme.

2.2.1 Idée générale

Une approche nouvelle basée sur un point de vue alternatif est ici proposée. Un système dynamique stochastique est vu comme possédant des fonctions de valeur sous-jacentes $V \in \mathbb{R}^S$ et des Q-fonctions $Q \in \mathbb{R}^{S \times A}$ que l'agent peut observer localement en interagissant avec le système. Lorsqu'un agent applique une action sur le système, il provoque un changement d'état et la génération d'une récompense immédiate. Cette récompense est en fait une observation locale de l'ensemble des fonctions sous-jacentes qui régissent le comportement du système. A partir d'une séquence de telles observations, l'agent devrait pouvoir inférer des informations sur une quelconque de ces fonctions. Une bonne estimation de la fonction de valeur $\hat{V}(s)$ (resp. Q-fonction $\hat{Q}(s,a)$) est fournie par l'espérance conditionnelle sur toutes les trajectoires de $V(s)$ (resp. $Q(s,a)$) étant donné la séquence de récompenses observées :

$$\hat{V}_i(s) = E[V(s)|r_1, \ldots, r_i]. \tag{2.12}$$

$$\hat{Q}_i(s,a) = E[Q(s,a)|r_1, \ldots, r_i]. \tag{2.13}$$

Interagir avec le système devient donc un moyen de générer des observations qui aident à l'estimation des fonctions de valeur qui sont des propriétés cachées du système. De ces estimations, des mises à jour de la politique courante peuvent être déduites afin de s'approcher de la politique optimale. Dans cette optique, il est aussi légitime de chercher à adopter le comportement qui permet de collecter le plus d'observations pertinentes ce qui permet un point de vue intéressant sur le problème bien connu du dilemme entre exploration et exploitation mentionné plus haut.

Deux cas particuliers de fonctions de valeur sont celles associées à la politique suivie par l'agent π et celle associée à la politique optimale π^*. Nous nous concentrerons sur les estimations de ces deux fonction particulières ou des Q-fonctions correspondantes.

Il est impossible de résoudre les équations (2.12) et (2.13) dans le cas général mais inférer un état caché à partir de séquences d'observations est typiquement traité dans le cadre du *filtrage de Kalman* [109] dans la communauté du traitement du signal. Les fonctions de valeur seront alors considérées comme étant générées par un jeu de paramètres et le but sera la recherche du jeu optimal de paramètres cachés θ^* qui fournit la *meilleure estimation* de la fonction de valeur. Dans le paradigme considéré, pour nous conformer au maximum au paradigme général de l'AR, ces paramètres seront mis à jour après chaque interaction et nous ferons l'hypothèse que la *meilleure mise à jour linéaire* est celle recherchée. Dans la suite, le filtrage de Kalman est d'abord présenté, puis une méthode permettant de placer l'approximation de fonction de valeur dans ce cadre (en utilisant les équations de Bellman introduites précédemment) sera décrite.

2.2.2 Filtrage de Kalman

A l'origine, le paradigme du filtrage de Kalman [109] a pour but de traquer l'état caché X (modélisé comme un vecteur aléatoire) d'un système dynamique stochastique non-stationnaire par le biais d'observations indirectes $\{Y_1, \ldots, Y_i\}$ de celui-ci. Pour ce faire, à l'instant $i-1$ l'algorithme calcule une prédiction de l'état ($\hat{X}_{i|i-1}$) et de l'observation ($\hat{Y}_{i|i-1}$) à la date i, ceci à partir d'une connaissance analytique de l'évolution des états et de la manière dont ils gênèrent des observations. Après l'obtention effective de l'observation suivante Y_i (à la date i), la prédiction de l'état est corrigée pour obtenir l'estimation de l'état $\hat{X}_{i|i}$ en utilisant l'erreur de prédiction de l'observation ($e_i = Y_i - \hat{Y}_{i|i-1}$) suivant une équation de type Windrow-Hoff (équation de mise à jour *linéaire*) :

$$\hat{X}_{i|i} = \hat{X}_{i|i-1} + K_i(Y_i - \hat{Y}_{i|i-1}) = \hat{X}_{i|i-1} + K_i e_i, \tag{2.14}$$

où K_i est le *gain de Kalman* qui sera décrit plus en détail par la suite. Dans le travail original de Kalman, la forme linéaire de l'équation (2.14) est une contrainte : en adoptant un point de vue statistique, le but du filtre de Kalman est de calculer récursivement le meilleur [1] estimateur linéaire $\hat{X}_i$ de l'état à la date i étant donné la séquence d'observations $\{Y_1, \ldots, Y_i\}$. Kalman considère que la meilleure estimation est celle qui minimise une fonction de coût quadratique :

$$J_i(\hat{X}) = E[\|X_i - \hat{X}\|^2 | Y_1, \ldots, Y_i]. \tag{2.15}$$

Pour calculer le gain optimal K_i sous les contraintes (2.14) et (2.15), plusieurs hypothèses sont faites.

Premièrement l'évolution du système est supposée régie par une *équation d'évolution* ou *équation de process* (basée sur une fonction f_i éventuellement non-stationnaire) connue :

$$X_{i+1} = f_i(X_i, v_i). \tag{2.16}$$

L'équation (2.16) lie l'état suivant X_{i+1} à l'état courant X_i et v_i est un bruit aléatoire habituellement appelé *bruit d'évolution* ou *bruit de process* qui modélise l'incertitude dans l'évolution du système.

Deuxièmement les observations sont supposées liées aux états par une autre fonction connue g_i utilisée dans l'équation usuellement appelée *équation d'observation* ou *équation de capteur* :

$$Y_i = g_i(X_i, w_i). \tag{2.17}$$

L'équation (2.17) relie l'observation courante Y_i à l'état courant X_i et w_i est un bruit aléatoire habituellement appelé *bruit d'observation* modélisant l'incertitude introduite par les observations imprécises. Ce bruit combiné au bruit d'évolution sont à l'origine du problème d'estimation de l'état caché (l'estimation de l'état courant étant donné la séquence d'observations bruitées).

Les équations (2.16) et (2.17) fournissent ce qui est appelé, dans le domaine de la commande optimale des systèmes, la description en *espace d'état* du système.

L'hypothèse centrale de Kalman est que v_i et w_i sont des bruits additifs, blancs, de moyenne nulle et indépendants, de variance P_v et P_w respectivement, ce qui signifie que :

$$E[v_i] = E[w_i] = 0, \tag{2.18}$$

$$E[v_i \cdot w_j] = 0 \quad \forall i, j, \tag{2.19}$$

$$E[v_j \cdot v_i] = E[w_j \cdot w_i] = 0 \quad \forall i \neq j. \tag{2.20}$$

Ces hypothèses étant posées ainsi que les contraintes (2.14) et (2.15) et adoptant un point de vue statistique, l'algorithme du filtre de Kalman fournit les quantités optimales $\hat{X}_{i|i-1}$, $\hat{Y}_{i|i-1}$ et K_i (les équations suivantes supposent les bruits additifs) :

$$\begin{aligned} \hat{X}_{i|i-1} &= E[X_i | Y_1, \ldots, Y_{i-1}] = E[f_{i-1}(X_{i-1}, v_{i-1}) | Y_1, \ldots, Y_{i-1}] \\ &= E[f_{i-1}(X_{i-1}) | Y_1, \ldots, Y_{i-1}] = E[f_{i-1}(\hat{X}_{i-1|i-1})], \end{aligned} \tag{2.21}$$

$$\begin{aligned} \hat{Y}_{i|i-1} &= E[Y_i | Y_1, \ldots, Y_{i-1}] = E[g_i(X_i, w_i) | Y_1, \ldots, Y_{i-1}] \\ &= E[g_i(X_i) | Y_1, \ldots, Y_{i-1}] = E[g_i(\hat{X}_{i-1|i-1})], \end{aligned} \tag{2.22}$$

$$K_i = P_{X e_i} P_{e_i}^{-1}. \tag{2.23}$$

où $P_{X e_i} = E[(X_i - \hat{X}_{i|i-1}) e_i | Y_1, \ldots, Y_{i-1}]$ et $P_{e_i} = \mathrm{cov}(e_i | Y_1, \ldots, Y_{i-1})$.

Le propos n'est pas ici de fournir les développements complets qui mènent à ces résultats généraux qui peuvent être trouvés dans l'article original [109]. Néanmoins, le Chapitre 3 fournit un développement plus complet dans le cadre de l'AR.

Plusieurs commentaires importants peuvent être faits à ce stade. Tout d'abord, aucune hypothèse spécifique sur les distributions des bruits v et w n'a été faite excepté qu'ils soient à moyenne nulle et de variance connue (P_v et P_w). A partir de là, le filtre de Kalman fournit la meilleure estimation linéaire de l'état du système. Cette estimation peut ne pas être optimale mais si les deux bruits possèdent des

1. En ce sens que l'équation de mise à jour est linéaire.

distributions gaussiennes alors ils sont totalement décrits par leur moyenne et leur variance. Dans ce cas spécifique, l'estimation linéaire est alors optimale et le filtre de Kalman fournit la solution optimale. Dans la suite de cette partie, l'hypothèse gaussienne de la distribution des bruits n'est jamais faite. Nous nous contenterons de chercher la meilleure estimation linéaire.

Ensuite, aucune hypothèse de linéarité n'est faite au sujet des fonctions f_i et g_i. Bien que [109] fournisse des solutions exactes au problème d'estimation dans le cas d'un espace d'état linéaire, seules les quantités impliquées dans (5.4), (5.5) et (5.6) sont requises. Il existe des schémas d'approximation afin d'estimer ces quantités y compris dans le cas où les équations (2.16) et (2.17) sont non-linéaires et même non-dérivables. Le Filtre de Kalman Etendu ou *Extended Kalman Filter* (EKF) [235] ou le Filtre de Kalman non-parfumé ou *Unscented Kalman Filter* (UKF) [107] sont de tels schémas.

Enfin, le filtrage de Kalman ne peut pas être totalement confondu avec le filtrage bayésien. Le filtrage bayésien consisterait à calculer entièrement la distribution *a posteriori* de l'état étant donné les observations (la totalité des moments). Le filtrage de Kalman se concentre seulement sur les moments d'ordres 1 et 2 de cette distribution (moyenne et variance). Encore une fois, ce n'est que dans le cas de distributions gaussiennes et de linéarité du modèle espace d'état que le filtrage bayésien se résume au filtrage de Kalman puisque dans ce cas les deux moments en question suffisent à décrire toute la distribution. Dans cette partie, seulement le filtrage de Kalman est considéré.

2.2.3 Formulation en espace d'état de l'estimation de fonctions de valeur

Avant de fournir le cadre général, les idées fondamentales sont introduites par le biais du problème de l'estimation de la fonction de valeur $V^\pi(s)$. Un point de vue statistique est donc adopté et le vecteur de paramètres θ est modélisé comme un vecteur aléatoire. La modélisation en espace d'état [2] suppose de définir l'évolution des paramètres. Néanmoins, la dynamique des paramètres ou de la fonction de valeur est difficilement modélisable puisque cela dépend du fait que le système soit stationnaire ou non ou encore que l'évaluation de la fonction de valeur prenne place dans un schéma d'apprentissage de type itération optimiste de la politique [3]. Ici une heuristique est choisie suivant le principe du rasoir d'Occam et l'évolution des paramètres est modélisée comme une marche aléatoire :

$$\theta_i = \theta_{i-1} + v_i. \tag{2.24}$$

Dans cette équation, θ_i est le (véritable) vecteur de paramètres (inconnu) à la date i et v_i est le bruit d'évolution. Le vecteur de paramètres θ_i est donc un processus aléatoire. Comme il est stationnaire (puisque $E[\theta_i] = E[\theta_{i-1}]$), il ne devrait pas poser de problème dans le cas où la fonction de valeur est stationnaire. En revanche, ce modèle devrait aussi permettre la *traque* d'une fonction de valeur non-stationnaire (même si ce modèle n'est pas totalement correct, ce qui ne peut être obtenu dans le cas général de toute manière).

Un autre problème est de lier ce qui est observé (les récompenses) à ce qui doit être inféré (le vecteur de paramètres représentant la fonction de valeur ou la Q-fonction). L'équation d'évaluation de Bellman (2.1) est une bonne candidate pour produire une telle équation d'observation :

$$r_i = V^\pi(s_i) - \gamma V^\pi(s_{i+1}). \tag{2.25}$$

Notons qu'il s'agit ici d'une version *échantillonnée* de l'équation (2.1) puisqu'il n'y a pas d'intervention des probabilités (ou d'espérance). De plus, la solution de l'équation de Bellman ne réside pas nécessairement dans l'espace d'hypothèses engendré par les paramètres [4]. C'est pourquoi il existe un biais inductif. Il faut donc ajouter un terme de bruit n_i, qui sera modélisé ici comme un bruit blanc [5] centré :

$$r_i = \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) + n_i. \tag{2.26}$$

2. La dénomination espace d'état (*state-space* en anglais) vient du filtrage de Kalman exposé plus tôt et n'a donc pas de lien avec la notion d'état d'un PDM.

3. A chaque amélioration de la politique, la fonction de valeur associée change elle aussi. De ce fait, la fonction de valeur à évaluer est non-stationnaire, même si l'environnement est stationnaire.

4. C'est-à-dire dans l'ensemble des fonctions qui peuvent être représentées par le jeu de paramètres étant donné une famille de fonctions paramétrées.

5. Cette hypothèse n'est pas toujours correcte et ce cas sera traité dans la Section 3.3.

Notons encore qu'aucune hypothèse de "gaussianité" n'est faite.

Les équations d'évolution et d'observation peuvent être résumées dans la formulation d'espace d'état suivante :

$$\begin{cases} \theta_i &= \theta_{i-1} + v_i, \\ r_i &= \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) + n_i. \end{cases} \tag{2.27}$$

Ceci est un modèle pour l'approximation de la fonction de valeur. Il suppose qu'il existe un processus aléatoire θ_i qui génère des récompenses par le biais de l'équation d'évaluation de Bellman, ces observations étant bruitées du fait du biais inductif et du fait qu'une version *échantillonnée* de l'équation de Bellman est utilisée [6]. Les états et les actions peuvent être considérés comme des variables exogènes qui font partie de la définition du modèle d'observation à la date i. Estimer la fonction de valeur revient donc à l'estimation de ce processus aléatoire caché. Ce problème peut être traité par le filtrage bayésien qui vise à estimer l'entièreté de la distribution de θ_i conditionné aux récompenses observées dans le passé. Dans cette partie, le point de vue plus restrictif du filtrage de Kalman est adopté et uniquement les moments d'ordre 1 et 2 seront estimés.

2.2.4 Transformation non-parfumée

La formulation en espace d'état décrite dans la section précédente est basée sur l'équation d'évaluation de la fonction de valeur de Bellman (équation (2.1)). Rien n'est supposé *a priori* sur la forme de la paramétrisation utilisée pour représenter la fonction de valeur. Ainsi, des paramétrisations non-linéaires devraient pouvoir être considérées. De plus, une formulation en espace d'état liant les paramètres à la Q-fonction optimale pourrait de façon similaire être écrite en utilisant l'équation d'optimalité de Bellman (équation (2.5)), ce qui introduirait un opérateur $\max$ et donc une forte non-linéarité. Ainsi, il est indispensable de pouvoir estimer les quantités impliquées dans (5.4), (5.5) et (5.6) même dans le cas où les fonctions f_i and g_i sont non-linéaires pour pouvoir travailler dans le cadre général que nous nous sommes fixé. C'est pourquoi, nous présentons ici la transformation non-parfumée qui permet l'approximation de ces quantités, dans le cas de fonctions non-linéaires et même non-dérivables (comme c'est le cas pour l'opérateur $\max$).

Laissons pour l'instant de côté l'AR et le filtrage de Kalman. Soit X un vecteur aléatoire, et Y une fonction de X. Le problème posé est de calculer la moyenne et la variance de Y en connaissant celles de X ainsi que la fonctionnelle les liant. Si cette dernière est linéaire, la relation entre X et Y peut s'écrire $Y = AX$ où A est une matrice de dimension *ad hoc*. Dans ce cas, moyenne et covariance peuvent être calculées analytiquement : $E[Y] = AE[X]$ et $E[YY^T] = AE[XX^T]A^T$.

Si la transformation est non-linéaire, elle peut se mettre sous la forme générique $Y = f(X)$. Une première solution est d'approximer la fonctionnelle même, c'est-à-dire de la linéariser autour de la moyenne du vecteur aléatoire X. Cela mène aux approximations suivantes pour la moyenne et la covariance de Y : $E[Y] \approx f\left(E[X]\right)$ et $E[YY^T] \approx \left(\nabla f\left(E[X]\right)\right) E[XX^T] \left(\nabla f\left(E[X]\right)\right)^T$. Ceci est la base du filtrage de Kalman étendu (voir [235] par exemple), qui a été intensivement étudié et utilisé dans les décennies passées. Cependant, cette approche présente quelques sévères limitations. Premièrement, elle ne permet pas de prendre en compte des non-linéarités non dérivables, et ne peut donc pas prendre en compte l'équation d'optimalité de Bellman (2.5) à cause de l'opérateur $\max$. Cette approche nécessite également d'évaluer le gradient de f, ce qui peut être compliqué voire impossible. Enfin, cela suppose que f est localement linéarisable, ce qui n'est malheureusement pas toujours le cas et peut mener à de mauvaises approximations, comme illustré par [107].

L'idée de base de la transformation non-parfumée est qu'il est plus judicieux d'approximer un vecteur aléatoire arbitraire qu'une fonction non-linéaire arbitraire. Son principe est d'échantillonner de façon *déterministe* un ensemble de "sigma-points" à partir de la moyenne et de la covariance de X. Les images de ces sigma-points par l'application f sont ensuite calculées, et elles sont utilisées pour calculer les statistiques d'intérêt. Ce schéma d'approximation ressemble aux méthodes de Monte-Carlo, cependant ici l'échantillonnage est déterministe et nécessite la génération de moins d'échantillons, en garantissant cependant une précision donnée [107].

6. Pas d'utilisation des probabilités de transitions, pas d'espérance.

Nous décrivons à présent la transformation non-parfumée originale. D'autres variantes ont été introduites depuis, mais le principe de base est le même. Soit n la dimension de X. Un ensemble de $2n+1$ sigma-points et poids associés est calculé comme suit :

$$\begin{cases} x^{(0)} = \bar{X} & w_0 = \frac{\kappa}{n+\kappa}, \quad j = 0 \\ x^{(j)} = \bar{X} + \left(\sqrt{(n+\kappa)P_X}\right)_j & w_j = \frac{1}{2(n+\kappa)}, \quad 1 \leq j \leq n \\ x^{(j)} = \bar{X} - \left(\sqrt{(n+\kappa)P_X}\right)_{n-j} & w_j = \frac{1}{2(n+\kappa)}, \quad n+1 \leq j \leq 2n \end{cases} \tag{2.28}$$

où $\bar{X}$ est la moyenne de X, P_X est sa matrice de variance, κ est un coefficient d'échelle permettant de contrôler la précision de la transformation non-parfumée [107], et $(\sqrt{(n+\kappa)P_X})_j$ est la $j^{\text{ème}}$ colonne de la décomposition de Cholesky de la matrice $(n+\kappa)P_X$. L'image de chacun de ces points par f est ensuite calculée : $y^{(j)} = f(x^{(j)}), 0 \leq j \leq 2n$. L'ensemble des sigma-points et de leurs images peut alors être utilisé pour calculer les moments d'ordres 1 et 2 de Y, et même P_{XY}, la covariance entre X et Y :

$$\begin{cases} \bar{Y} \approx \bar{y} = \sum_{j=0}^{2n} w_j y^{(j)} \\ P_Y \approx \sum_{j=0}^{2n} w_j \left(y^{(j)} - \bar{y}\right)\left(y^{(j)} - \bar{y}\right)^T \\ P_{XY} \approx \sum_{j=0}^{2n} w_j \left(x^{(j)} - \bar{X}\right)\left(y^{(j)} - \bar{y}\right)^T \end{cases} \tag{2.29}$$

Chapitre 3

Différences temporelles de Kalman

Nous présentons ici la contribution principale apportée à l'AR, le cadre théorique des *différences temporelles de Kalman ou Kalman Temporal Differences* (KTD), qui a fait l'objet de la thèse de Matthieu GEIST [71] financée par ArcelorMittal dans le cadre d'une convention CIFRE et en collaboration avec l'équipe CORIDA de l'INRIA Nancy - Grand-Est.

3.1 Différences temporelles de Kalman : cas général

Dans cette section, un point de vue très général est adopté, les algorithmes spécifiques étant dérivés par la suite. L'AR étant essentiellement séquentiel, il est nécessaire de travailler sur des transitions. Suivant le type d'algorithme utilisé, des transitions de natures différentes peuvent être considérées. Pour l'instant, une transition est notée de façon générique par t_i qui pourra prendre les formes suivantes :

$$t_i = \begin{cases} (s_i, s_{i+1}) \\ (s_i, a_i, s_{i+1}, a_{i+1}) \\ (s_i, a_i, s_{i+1}) \end{cases} \tag{3.1}$$

selon que l'objectif soit l'évaluation de la fonction de valeur, de qualité, ou l'optimisation directe de Q. De façon similaire, pour les mêmes cas, les notations suivantes sont adoptées :

$$g_{t_i}(\theta_i) = \begin{cases} \hat{V}_{\theta_i}(s_i) - \gamma \hat{V}_{\theta_i}(s_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) \\ \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\theta_i}(s_{i+1}, b) \end{cases} \tag{3.2}$$

Ainsi, tous les schémas de différences temporelles décrits dans la Section 2.1 peuvent s'écrire de façon générique :

$$\delta_i = r_i - g_{t_i}(\theta_i) \tag{3.3}$$

Avec ces notations, $g_{t_i}(\theta_i)$ peut être vu comme la prédiction de la récompense au pas de temps i en accord avec la représentation θ_i, et δ_i peut être vu comme un terme d'innovation qui quantifie l'information gagnée en observant la nouvelle récompense r_i lors de la transition t_i. Ces notions sont raffinées plus tard.

Comme dans la section 2.2.3, le vecteur de paramètres θ est modélisé comme étant un vecteur aléatoire suivant une marche aléatoire. Le problème peut ainsi s'exprimer sous une forme espace d'état :

$$\begin{cases} \theta_i = \theta_{i-1} + v_i & \text{(équation d'évolution)} \\ r_i = g_{t_i}(\theta_i) + n_i & \text{(équation d'observation)} \end{cases} \tag{3.4}$$

3.1.1 Coût minimisé

En se référant au cadre du filtrage de Kalman décrit dans la Section 2.2.2, l'objectif est d'estimer le vecteur de paramètres qui minimise l'espérance de l'erreur quadratique conditionnée aux récompenses observées depuis l'origine des temps. Le coût associé s'écrit :

$$J_i(\theta) = E\left[\|\theta_i - \theta\|^2 | r_{1:i}\right] \text{ avec } r_{1:i} = r_1, \dots, r_i \tag{3.5}$$

Ce problème est assez connu et l'estimateur minimisant l'erreur quadratique moyenne est l'espérance conditionnelle :

$$\underset{\theta}{\operatorname{argmin}}\, J_i(\theta) = \hat{\theta}_{i|i} = E\left[\theta_i | r_{1:i}\right] \tag{3.6}$$

Cependant, à part pour des cas spécifiques (notamment le cas où les équations d'évolution et d'observation sont linéaires et les bruits gaussiens), cet estimateur ne peut pas être calculé analytiquement. Il faut donc s'imposer des contraintes supplémentaires et l'objectif est ici de trouver le meilleur estimateur *linéaire*, comme dans le cas du filtre de Kalman. Il peut être écrit sous une forme similaire à l'équation (2.8) :

$$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i \tilde{r}_i \tag{3.7}$$

Dans l'équation (3.7), $\hat{\theta}_{i|i}$ est l'estimation du vecteur de paramètres au temps i et $\hat{\theta}_{i|i-1} = E[\theta_i | r_{1:i-1}]$ est la prédiction de cette estimation en accord avec les récompenses observées dans le passé $r_{1:i-1}$. Pour le modèle de marche aléatoire adopté, nous avons $\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$. En effet, en utilisant la définition de la prédiction et l'équation d'évolution nous avons $\hat{\theta}_{i|i-1} = E[\theta_{i-1} + v_i | r_{1:i-1}]$. Or le bruit d'évolution est blanc et centré donc $E[v_i | r_{1:i-1}] = 0$. Ceci mène au résultat : $\hat{\theta}_{i|i-1} = E[\theta_{i-1} | r_{1:i-1}] = \hat{\theta}_{i-1|i-1}$. L'innovation

$$\tilde{r}_i = r_i - \hat{r}_{i|i-1} \tag{3.8}$$

est la différence entre la récompense observée r_i et sa prédiction basée sur la précédente estimation du vecteur de paramètres, donnée par (rappelons que le bruit d'observation est également blanc et centré) :

$$\hat{r}_{i|i-1} = E\left[r_i | r_{1:i-1}\right] = E\left[g_{t_i}(\theta_i) + n_i | r_{1:i-1}\right] = E\left[g_{t_i}(\theta_i) | r_{1:i-1}\right] \tag{3.9}$$

Notons que cette innovation n'est pas exactement l'erreur de différence temporelle définie dans l'équation (3.3), qui est une variable aléatoire en conséquence de sa dépendance au vecteur aléatoire θ_i : c'est son espérance conditionnée aux données précédemment observées. Etant donné la mise à jour postulée (3.7), il s'agit de déterminer le gain K_i qui permette la minimisation du coût (3.5).

3.1.2 Gain optimal

En utilisant des égalités classiques, la fonction de coût peut se réécrire de la façon suivante (l'opérateur trace associant à une matrice carrée la somme de ses éléments diagonaux) :

$$\begin{aligned} J_i(\theta) &= E\left[\|\theta_i - \theta\|^2 | r_{1:i}\right] = E\left[(\theta_i - \theta)^T (\theta_i - \theta) | r_{1:i}\right] \\ &= \mathrm{trace}\left(E\left[(\theta_i - \theta)(\theta_i - \theta)^T | r_{1:i}\right]\right) = \mathrm{trace}\left(\mathrm{cov}\left(\theta_i - \theta | r_{1:i}\right)\right) \end{aligned} \tag{3.10}$$

Une première étape pour calculer le gain optimal est d'exprimer la covariance de l'erreur sur les paramètres conditionnée aux récompenses comme une fonction du gain K_i. Mais d'abord quelques notations supplémentaires sont introduites (rappelons également la définition de l'innovation (3.8)) :

$$\begin{cases} \tilde{\theta}_{i|i} = \theta_i - \hat{\theta}_{i|i} & \text{et} \quad \tilde{\theta}_{i|i-1} = \theta_i - \hat{\theta}_{i|i-1} \\ P_{i|i} = \mathrm{cov}\left(\tilde{\theta}_{i|i} | r_{1:i}\right) & \text{et} \quad P_{i|i-1} = \mathrm{cov}\left(\tilde{\theta}_{i|i-1} | r_{1:i-1}\right) \\ P_{r_i} = \mathrm{cov}\left(\tilde{r}_i | r_{i|i-1}\right) & \text{et} \quad P_{\theta r_i} = E\left[\tilde{\theta}_{i|i-1} \tilde{r}_i | r_{1|i-1}\right] \end{cases} \tag{3.11}$$

En utilisant la mise à jour postulée (3.7) et les différents estimateurs étant non-biaisés, la covariance peut être réécrite :

$$\begin{aligned} P_{i|i} &= \mathrm{cov}\left(\theta_i - \hat{\theta}_{i|i} | r_{1:i}\right) = \mathrm{cov}\left(\theta_i - \left(\hat{\theta}_{i|i-1} + K_i \tilde{r}_i\right) | r_{1:i-1}\right) \\ &= \mathrm{cov}\left(\tilde{\theta}_{i|i-1} - K_i \tilde{r}_i | r_{1:i-1}\right) = P_{i|i-1} - P_{\theta r_i} K_i^T - K_i P_{\theta r_i}^T + K_i P_{r_i} K_i^T \end{aligned} \tag{3.12}$$

Le gain optimalest ainsi obtenu en annulant le gradient de la trace de cette matrice.

Notons tout d'abord que le gradient étant linéaire, pour trois matrices de dimensions *ad hoc* A, B et C, B étant symétrique, nous avons les identités algébriques suivantes :

$$\nabla_A \left(\text{trace}\left(ABA^T\right)\right) = 2AB \text{ et } \nabla_A \left(\text{trace}\left(AC^T\right)\right) = \nabla_A \left(\text{trace}\left(CA^T\right)\right) = C \tag{3.13}$$

et donc en utilisant l'équation (3.12) et les identités précédentes nous avons :

$$\nabla_{K_i} \left(\text{trace}\left(P_{i|i}\right)\right) = 0 \Leftrightarrow \quad K_i = P_{\theta r_i} P_{r_i}^{-1} \tag{3.14}$$

En injectant le gain optimal (3.14) dans l'expression de la matrice de covariance de l'erreur conditionnée aux récompenses observées (3.12), nous en obtenons une expressions simplifiée :

$$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T \tag{3.15}$$

Il est à noter, encore une fois, qu'aucune hypothèse gaussienne n'a été faite pour obtenir ces résultats comme c'est pourtant généralement le cas dans la littérature faisant usage du filtre de Kalman (mais nous sommes plus proche de la description originelle de Kalman).

3.1.3 Algorithme général

L'algorithme le plus général de différences temporelles de Kalman, qui se subdivise en trois parties, peut maintenant être obtenu. La première étape consiste à calculer les prédictions $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$. Rappelons que pour un modèle de marche aléatoire la prédiction du vecteur de paramètres est égale à son estimation précédente : $\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$. La covariance prédite peut également être calculée analytiquement :

$$P_{i|i-1} = \text{cov}\left(\tilde{\theta}_{i|i-1}|r_{1:i-1}\right) = \text{cov}\left(\tilde{\theta}_{i-1|i-1} + v_{i-1}|r_{1:i-1}\right) = P_{i-1|i-1} + P_{v_{i-1}} \tag{3.16}$$

La seconde étape consiste à calculer quelques statistiques d'intérêt. C'est principalement cette partie qui sera spécialisée dans la section suivante. La première statistique à calculer est la prédiction de la récompense $\hat{r}_{i|i-1}$ (3.9). La seconde est la covariance entre l'erreur sur les paramètres et l'innovation :

$$P_{\theta r_i} = E\left[(\theta_i - \hat{\theta}_{i|i-1})(r_i - \hat{r}_{i|i-1})|r_{1:i-1}\right] \tag{3.17}$$

Etant donné la forme de l'équation d'observation ($r_i = g_{t_i}(\theta_i) + n_i$) et l'indépendance du bruit d'observation qui est centré, cette statistique peut se réécrire :

$$P_{\theta r_i} = E\left[(\theta_i - \hat{\theta}_{i|i-1})(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})|r_{1:i-1}\right] \tag{3.18}$$

Enfin, la dernière statistique à calculer est la variance de l'innovation, qui peut être écrite (en utilisant à nouveau les caractéristiques du bruit d'observation) :

$$\begin{aligned} P_{r_i} &= E\left[(r_i - \hat{r}_{i|i-1})^2|r_{1:i-1}\right] = E\left[(g_{t_i}(\theta_i) - \hat{r}_{i|i-1} + n_i)^2|r_{1:i-1}\right] \\ &= E\left[(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})^2|r_{1:i-1}\right] + P_{n_i} \end{aligned} \tag{3.19}$$

La dernière étape de l'algorithme est la phase de correction qui consiste à calculer le gain (3.14) et à mettre à jour le vecteur de paramètres (3.7) ainsi que la matrice de covariance associée (3.15). Notons que la méthode proposée étant en ligne, elle se doit d'être initialisée avec un *a priori* sur l'espérance $\hat{\theta}_{0|0}$ et la covariance $P_{0|0}$ des paramètres. L'approche générale proposée est résumée dans l'algorithme 1. Le calcul de la variance $P_{i|i}$ peut sembler à première vue inutile, mais elle est utilisée dans tous les cas (linéaire, linéarisé, non-linéaire) pour calculer les statistiques d'intérêt, comme explicité dans la section 3.2.

3.2 Transitions déterministes

La principale difficulté de KTD est le calcul des statistiques d'intérêt $\hat{r}_{i|i-1}$, $P_{\theta r_i}$ et P_{r_i} (pour lesquelles les statistiques $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$ sont nécessaires). Nous étudions dans cette section le cas déterministe car, comme pour tous les algorithmes basés sur la minimisation du résidu de Bellman, la méthode KTD induit un biais dans l'estimation des paramètres dans le cas de transitions stochastiques. Le cas stochastique sera étudié dans la Section 3.3

Algorithme 1 : Algorithme KTD général.

Initialisation : *a priori* $\hat{\theta}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition t_i ainsi que la récompense associée r_i ;

Phase de prédiction;

$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;

$P_{i|i-1} = P_{i-1|i-1} + P_{v_{i-1}}$;

Calcul des statistiques d'intérêt;

$\hat{r}_{i|i-1} = E[g_{t_i}(\theta_i)|r_{1:i-1}]$;

$P_{\theta r_i} = E\left[(\theta_i - \hat{\theta}_{i|i-1})(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})|r_{1:i-1}\right]$;

$P_{r_i} = E\left[(g_{t_i}(\theta_i) - \hat{r}_{i|i-1})^2|r_{1:i-1}\right] + P_{n_i}$;

Phase de correction;

$K_i = P_{\theta r_i} P_{r_i}^{-1}$;

$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;

$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

3.2.1 KTD-V

Le premier problème auquel nous nous attaquons est l'évaluation de la fonction de valeur, c'est-à-dire trouver une solution approchée de l'équation d'évaluation de Bellman (2.1). Dans le cas d'une représentation linéaire comme celle proposée dans l'équation (2.6), il est possible de trouver une solution analytique au problème de l'estimation de la fonction $\hat{V}_\theta(s)$. Le gain peut ainsi être défini de façon algébrique et récursive :

$$K_i = \frac{P_{i|i-1}H_i}{H_i^T P_{i|i-1} H_i + P_{n_i}} \tag{3.20}$$

où $H_i = \phi(s_i) - \gamma\phi(s_{i+1})$ (voir [95, 99, 75, 71] pour une dérivation complète).

Nous préférons développer ici les dérivations dans le cas d'une paramétrisation générique de la fonction de valeur $\hat{V}_\theta$: cela peut être un réseau de neurones [16, 73], une représentation par noyaux semi-paramétrique [86, 94], ou toute autre représentation, du moment qu'elle puisse être décrite par un ensemble fini de p paramètres. La formulation générale espace d'état de l'équation (3.4) s'écrit dans ce cas :

$$\begin{cases} \theta_i = \theta_{i-1} + v_i \\ r_i = \hat{V}_{\theta_i}(s_i) - \gamma\hat{V}_{\theta_i}(s_{i+1}) + n_i \end{cases} \tag{3.21}$$

Le problème est toujours de calculer les statistiques d'intérêt, ce qui est fait ici en utilisant la transformation non-parfumée décrite à la Section 2.2.4. La première chose à faire est de calculer l'ensemble des sigma-points à partir des statistiques connues $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$, ainsi que les poids associés :

$$\Theta_{i|i-1} = \left\{\hat{\theta}^{(j)}_{i|i-1},\ 0 \leq j \leq 2p\right\} \tag{3.22}$$

$$\mathcal{W} = \{w_j,\ 0 \leq j \leq 2p\} \tag{3.23}$$

Ensuite les images de ces sigma-points sont calculées en utilisant l'équation d'observation du modèle espace d'état (3.21), qui est lié à l'équation d'évaluation de Bellman (2.1) :

$$\mathcal{R}_{i|i-1} = \left\{\hat{r}^{(j)}_{i|i-1} = \hat{V}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i) - \gamma\hat{V}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}),\ 0 \leq j \leq 2p\right\} \tag{3.24}$$

Enfin, les sigma-points et leurs images étant calculés, les statistiques d'intérêt peuvent être approchées

par :

$$\hat{r}_{i|i-1} \approx \sum_{j=0}^{2p} w_j \hat{r}_{i|i-1}^{(j)} \tag{3.25}$$

$$P_{r_i} \approx \sum_{j=0}^{2p} w_j \left(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1}\right)^2 + P_{n_i} \tag{3.26}$$

$$P_{\theta r_i} \approx \sum_{j=0}^{2p} w_j \left(\hat{\theta}_{i|i-1}^{(j)} - \hat{\theta}_{i|i-1}\right)\left(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1}\right) \tag{3.27}$$

Etant donné que la transformation non-parfumée n'est plus une approximation dans le cas d'une transformation linéaire, cette formulation est toujours valide pour l'évaluation de la fonction de valeur avec une paramétrisation linéaire. L'algorithme 2 résume l'approche et est donc un algorithme générique.

Algorithme 2 : KTD-V, KTD-SARSA et KTD-Q.

Initialisation : *a priori* $\hat{\theta}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition $t_i = \begin{cases} (s_i, s_{i+1}) \text{ (KTD-}V\text{)} \\ (s_i, a_i, s_{i+1}, a_{i+1}) \text{ (KTD-SARSA)} \\ (s_i, a_i, s_{i+1}) \text{ (KTD-}Q\text{)} \end{cases}$ ainsi que la récompense r_i ;

Phase de prédiction;

$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;

$P_{i|i-1} = P_{i-1|i-1} + P_{v_{i-1}}$;

Calcul des sigma-points ;

$\Theta_{i|i-1} = \left\{\hat{\theta}_{i|i-1}^{(j)}, \quad 0 \leq j \leq 2p\right\}$ (en utilisant $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$);

$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p \quad\}$;

$\mathcal{R}_{i|i-1} = \begin{cases} \left\{\hat{r}_{i|i-1}^{(j)} = \hat{V}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i) - \gamma \hat{V}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}),\ 0 \leq j \leq 2p\right\} \text{ (KTD-}V\text{)} \\ \left\{\hat{r}_{i|i-1}^{(j)} = \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i, a_i) - \gamma \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}, a_{i+1}),\ 0 \leq j \leq 2p\right\} \text{ (KTD-SARSA)} \\ \left\{\hat{r}_{i|i-1}^{(j)} = \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}, b),\ 0 \leq j \leq 2p\right\} \text{ (KTD-}Q\text{)} \end{cases}$;

Calcul des statistiques d'intérêt;

$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}_{i|i-1}^{(j)}$;

$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}_{i|i-1}^{(j)} - \hat{\theta}_{i|i-1})(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1})$;

$P_{r_i} = \sum_{j=0}^{2p} w_j \left(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1}\right)^2 + P_{n_i}$;

Phase de correction;

$K_i = P_{\theta r_i} P_{r_i}^{-1}$;

$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i \left(r_i - \hat{r}_{i|i-1}\right)$;

$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

3.2.2 KTD-SARSA

Traitons ensuite du problème de l'évaluation de la fonction de qualité pour une politique donnée. L'algorithme associé est appelé KTD-SARSA, ce qui peut induire en erreur. En effet, de façon générale, le terme SARSA est associé à l'évaluation de la Q-fonction dans le cadre d'une itération optimiste de

la politique (politique ϵ-gloutonne par exemple). Ici nous nous focalisons sur l'aspect évaluation de la fonction de qualité, en laissant le contrôle de côté (ce problème étant tout de même considéré dans la section 3.2.5). Pour une paramétrisation générale $\hat{Q}_\theta$, en considérant l'équation d'évaluation de Bellman (2.2) échantillonnée, le modèle espace d'état (3.4) s'écrit :

$$\begin{cases} \theta_i = \theta_{i-1} + v_i \\ r_i = \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \hat{Q}_{\theta_i}(s_{i+1}, a_{i+1}) + n_i \end{cases} \tag{3.28}$$

Pour une politique fixée, l'évaluation de la Q-fonction sur la chaîne de Markov valuée induite par l'espace état-action est similaire au problème de l'évaluation de la fonction de valeur sur la chaîne de Markov valuée induite par l'espace d'état. Il est alors assez évident d'étendre KTD-V à l'évaluation de la Q-fonction comme également résumé dans l'algorithme 2.

3.2.3 KTD-Q

Intéressons-nous enfin à l'optimisation directe de la Q-fonction, c'est-à-dire à trouver une solution approchée de l'équation d'optimalité de Bellman (2.5). Une paramétrisation générale $\hat{Q}_\theta$ est adoptée. Le modèle espace d'état (3.4) est alors spécialisé comme suit :

$$\begin{cases} \theta_i = \theta_{i-1} + v_i \\ r_i = \hat{Q}_{\theta_i}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\theta_i}(s_{i+1}, b) + n_i \end{cases} \tag{3.29}$$

Ici la distinction entre paramétrisation linéaire ou non n'est pas utile à faire. En effet l'opérateur $\max$, inhérent à l'équation d'optimalité de Bellman, rend l'équation d'observation de (3.29) non-linéaire quelle que soit la paramétrisation. Cet opérateur est difficile à prendre en compte, particulièrement à cause de sa non-dérivabilité.

Heureusement, comme elle approxime le vecteur aléatoire plutôt que la fonction, la transformation non-parfumée ne requiert pas de dérivation. Etant donné l'algorithme KTD général présenté section 3.1 et la transformation non-parfumée décrite dans la section 2.2.4, il est possible d'obtenir KTD-Q [1], l'algorithme KTD pour l'optimisation directe de la Q-fonction. Une première étape est de calculer les sigma-points associés au vecteur aléatoire de paramètres, comme dans les équations (3.22-3.23). Ensuite, l'image de ces sigma-points à travers l'équation d'observation du modèle espace d'état (3.29), qui contient l'opérateur $\max$, est calculée :

$$\mathcal{R}_{i|i-1} = \left\{ \hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i) - \gamma \max_{b \in A} \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, b),\ 0 \leq j \leq 2p \right\} \tag{3.30}$$

Enfin, les sigma-points et leurs images sont utilisés pour calculer les statistiques d'intérêt, comme dans les équations (3.25-3.27). L'approche KTD-Q proposée est aussi résumée dans l'algorithme 2.

3.2.4 Coût computationnel

Soit p le nombre de paramètres. La transformation non-parfumée implique de calculer une décomposition de Cholesky qui a une complexité computationnelle en $O(p^3)$. Cependant, pour les algorithmes considérés, la structure de mise à jour particulière de la matrice de covariance $P_{i|i-1}$, qui est de rang un, permet de considérer un algorithme spécifique de mise à jour de la décomposition de Cholesky dont la complexité est en $O(p^2)$. Les détails de cette approche "racine carrée" sont donnés par [250]. Les différents algorithmes impliquent d'évaluer $2p + 1$ fois la fonction g_{t_i} à chaque pas de temps. Pour KTD-V et KTD-SARSA et une paramétrisation générale, chaque évaluation est en $O(p)$. Pour KTD-Q, le maximum selon les actions doit être calculé. Notons $\mathcal{A}$ le nombre d'actions si l'espace correspondant est fini, et la complexité computationnelle de l'algorithme utilisé pour trouver ce maximum sinon (par exemple le nombre d'échantillons tirés multiplié par la complexité de leur évaluation pour Monte-Carlo). Ainsi chaque évaluation est bornée par $O(p\mathcal{A})$. Le reste des opérations est de l'algèbre linéaire basique,

1. Pendant de l'algorithme Q-learning de [253].

de complexité au plus $O(p^2)$. Ainsi, la complexité computationnelle (par itération) de KTD-V et KTD-SARSA est $O(p^2)$, celle de KTD-Q $O(\mathcal{A}p^2)$. L'ensemble des algorithmes requiert de stocker le vecteur de paramètres ainsi que la matrice de covariance associée, leur complexité en mémoire est donc $O(p^2)$.

3.2.5 Expérimentations

Des applications pratiques à des problèmes réels seront présentées dans le Chapitres 4. Ici, nous proposons un ensemble de tests de référence classiques en AR de façon à comparer KTD à différents algorithmes de l'état de l'art et à illustrer de manière plus évidente les différents aspects de cette approche, à savoir le biais causé par les transitions stochastiques (chaîne de Boyan), la robustesse à la non-stationnarité (chaîne de Boyan, *mountain car*), l'incertitude de la valeur utilisée pour une forme d'apprentissage actif (pendule inversé), ainsi que l'efficacité en terme d'échantillons (c'est-à-dire vitesse de convergence, toutes les expériences). Les autres algorithmes considérés sont TD, SARSA, Q-learning et LSTD [2] [20] qui est un algorithme hors-ligne efficace standard de la littérature. Nous ne considérons pas leurs extensions aux traces d'éligibilité (voir Section 3.3.3), dans la mesure où LSTD produit de meilleurs résultats que TD(λ) et où varier la valeur du facteur d'éligibilité a peu d'incidence sur les performance de LSTD(λ), comme noté par [19]. Plus de détails sur ces expériences et notamment sur les paramètres exactes utilisés pour initialiser les différents algorithmes peuvent être trouvés dans [75].

Chaîne de Boyan

La première expérimentation est la chaîne de Boyan [19]. Le but est d'une part d'illustrer le problème posé par les transitions stochastiques et d'autre part de montrer l'efficacité de KTD en terme d'échantillons et sa capacité à prendre en compte un environnement non-stationnaire sur une version déterministe du problème. Notons que la faculté de KTD à traquer une solution plutôt qu'à converger vers celle-ci a été développée dans [97].

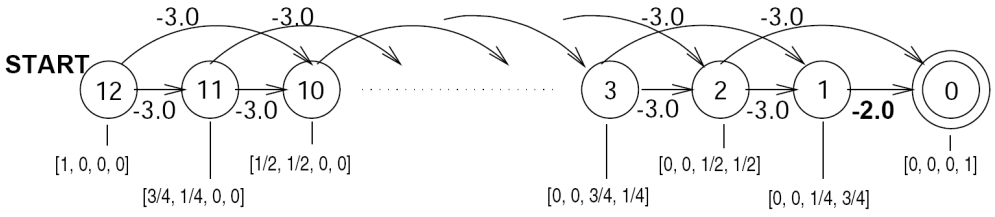
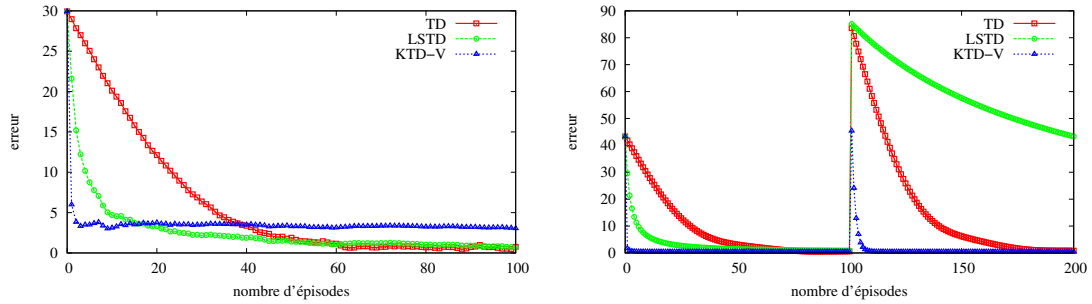


FIGURE 3.1 – Chaîne de Boyan.

Nous considérons donc dans un premier temps un problème faisant intervenir des **transitions stochastiques**. La chaîne de Boyan (voir Fig. 3.1) est une chaîne de Markov valuée à 13 états dont l'état s^0 est absorbant, s^1 transite vers s^0 avec une probabilité de 1 et une récompense de -2, et s^i transite vers s^{i-1} ou s^{i-2}, $2 \leq i \leq 12$, pour chacun avec une probabilité de 0.5 et une récompense de -3. Pour cette expérience, KTD-V est comparé à TD [242] ainsi qu'à LSTD [20]. La paramétrisation est linéaire, et les vecteurs de base $\phi(s)$ pour les états s^{12}, s^8, s^4 et s^0 sont respectivement $[1,0,0,0]^T$, $[0,1,0,0]^T$, $[0,0,1,0]^T$ et $[0,0,0,1]^T$. Pour les autres états, les vecteurs de base sont obtenus par interpolation linéaire. L'approximation de la fonction de valeur est donc $\hat{V}_\theta(s) = \theta^T\phi(s)$. La fonction de valeur optimale est linéaire en ces bases, et le vecteur de paramètres optimal correspondant est [3] $\theta^* = [-24,-16,-8,0]^T$. La performance est mesurée par la distance euclidienne $\|\theta - \theta^*\|$ entre le vecteur de paramètres courant et le vecteur optimal. Les résultats sont présentés figure 3.2.a.

LSTD converge plus rapidement que TD, comme attendu, et KTD-V converge encore plus vite. Cependant ce dernier algorithme ne converge pas vers le vecteur de paramètres optimal, ce qui s'explique par le fait que la fonction de coût minimisée est biaisée. Cette expérience a été menée pour montrer le problème causé par les transitions stochastiques.

2. *Least Square Temporal Differences*.
3. Ceci peut être calculé analytiquement.



a. Cas stochastique. b. Cas déterministe et non-stationnaire.

FIGURE 3.2 – Résultats : chaîne de Boyan.

Nous nous concentrons à présent sur des PDM **déterministes**. Pour l'expérience suivante, la chaîne de Boyan est rendue déterministe en posant la probabilité de transiter de s^i à s^{i-1} à 1. KTD-V est à nouveau comparé à LSTD et TD. De plus, pour simuler un changement dans le PDM, et donc la non-stationnarité, le signe de la récompense est inversé à partir du $100^{\text{ème}}$ épisode. La fonction de valeur optimale est toujours linéaire en les fonctions de base, et le vecteur de paramètres optimal est $\theta^*_{(-)} = [-35, -23, -11, 0]^T$ avant le changement de récompense, et $\theta^*_{(+)} = -\theta^*_{(-)}$ après. Les résultats sont présentés sur la figure 3.2.b.

A nouveau KTD-V converge plus rapidement que LSTD et TD, cependant maintenant vers le vecteur de paramètres optimal, l'environnement étant déterministe. Après le changement de récompense, LSTD est très lent à converger, à cause de la non-stationnarité induite. TD s'adapte plus rapidement, le taux d'apprentissage étant constant. KTD-V s'adapte quant à lui très rapidement. Ainsi, si KTD-V est biaisé dans le cas stochastique, il converge plus vite et s'adapte plus rapidement que TD et LSTD dans le cas déterministe et non-stationnaire. Cette capacité à prendre en compte les non-stationnarités est importante pour suivre la dynamique de la fonction de valeur. Même si l'environnement est stationnaire, cela peut être utile dans un contexte de contrôle, comme illustré dans la section 3.2.5.

Pendule inversé

La seconde expérience est le pendule inversé tel que décrit par [119] et représenté sur la Fig. 3.3. Le but est ici de comparer deux algorithmes de type "itération de la valeur", à savoir KTD-Q et Q-learning avec approximation de la Q-fonction, qui ont tous deux pour objectif d'estimer directement la fonction de qualité optimale. Autant que nous le sachions, KTD-Q est le seul algorithme d'ordre 2 qui suive un schéma d'itération de la valeur, la difficulté majeure étant la prise en compte de l'opérateur $\max$ [4].

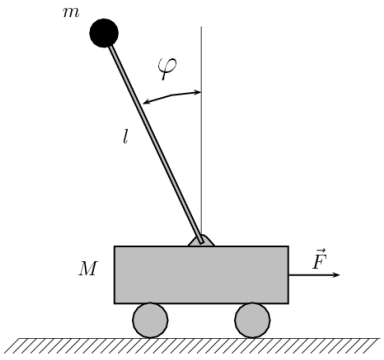


FIGURE 3.3 – Pendule Inversé.

Ce problème consiste à balancer un pendule de longueur et masse inconnues de façon à ce qu'il reste vertical en appliquant des forces au chariot sur lequel il est fixé. Trois actions sont possibles : pousser

4. A noter que les auteurs de [256] proposent également un tel algorithme, cependant pour une classe plus restreinte de PDM.

à gauche (-1), pousser à droite ($+1$), ou ne rien faire (0). L'état du système est donné par la position angulaire ω et la vitesse angulaire $\dot{\omega}$. Les transitions déterministes sont calculées grâce à la dynamique du système physique :

$$\ddot{\omega} = \frac{g\sin(\omega) - \beta ml\dot{\omega}^2\sin(2\omega)/2 - 50\beta\cos(\omega)a}{4l/3 - \beta ml\cos^2(\omega)} \tag{3.31}$$

où g est la constante de gravitation, m et l sont la masse et la longueur du pendule, M la masse du chariot et $\beta = \frac{1}{m+M}$. Une récompense nulle est donnée tant que la position angulaire appartient à l'intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Sinon, l'épisode se termine et une récompense de -1 est donnée. La paramétrisation est donnée par un terme constant et un ensemble de 9 noyaux gaussiens équi-répartis (centrés en $\{-\frac{\pi}{4}, 0, \frac{\pi}{4}\} \times \{-1, 0, 1\}$ et d'écart-type 1), cela pour chaque action. Il y a donc 30 fonctions de base. Le facteur d'actualisation γ est fixé à 0.95.

Dans un premier temps nous comparons la capacité des deux algorithmes à **apprendre la politique optimale**. Pour Q-learning, le taux d'apprentissage est fixé à $\alpha_i = \alpha_0 \frac{n_0+1}{n_0+i}$ où $\alpha_0 = 0.5$ et $n_0 = 200$, en accord avec [119]. Pour KTD-Q, les paramètres sont $P_{0|0} = 10I$, $P_{n_i} = 1$ et $P_{v_i} = 0I$. Les vecteurs de paramètres sont initialisés à 0. La politique suivie par l'agent est aléatoire (équi-probabilité des actions), et les deux algorithmes apprennent à partir des mêmes trajectoires. L'agent est initialisé dans un état aléatoire proche de l'équilibre $(0, 0)$. La longueur moyenne d'un tel épisode aléatoire est d'environ 10 pas. Les résultats sont présentés figure 3.4.

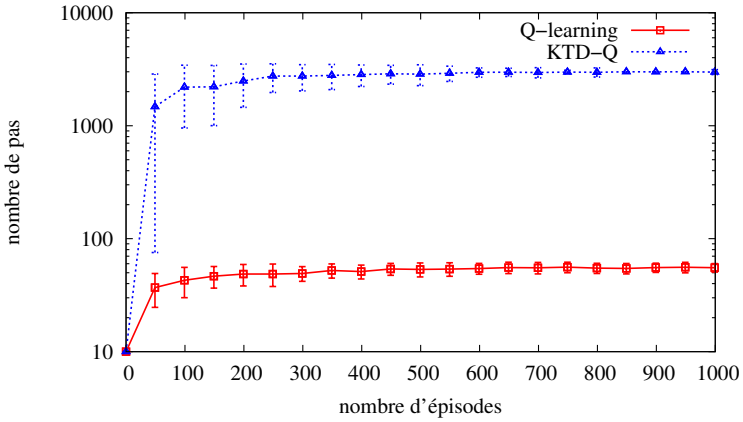


FIGURE 3.4 – Résultats : Pendule inversé.

Pour chaque essai, l'apprentissage est fait sur 1000 épisodes. Tous les 50 épisodes, l'apprentissage est gelé et la politique courante est testée. Pour cela, l'agent est initialisé dans un état aléatoire proche de l'équilibre et la politique gloutonne est suivie. Chaque test est répété 100 fois et moyenné [5]. La mesure de performance est le nombre de pas d'un épisode. Le nombre maximum de pas autorisé est de 3 000, ce qui correspond à maintenir le pendule à la verticale pendant 5 minutes. Les résultats de la Fig. 3.4.1 sont moyennés sur 100 essais (chaque essai correspondant à 1000 épisodes, la politique étant testée tous les 50 épisodes) et présentés en échelle semi-logarithmique.

Asymptotiquement, KTD-Q apprend la politique optimale (c'est-à-dire maintenir le pendule pendant le nombre maximum de pas autorisés) et de bonnes politiques sont apprises après seulement quelques dizaines d'épisodes. Avec le même nombre d'épisodes et la même paramétrisation, Q-learning échoue à apprendre une politique qui permette de maintenir le pendule pendant plus de quelques secondes, ce qui est en accord avec les résultats présentés dans [119].

Mountain car

La dernière expérience est le *mountain car* telle que décrite par [242]. L'objectif ici est d'illustrer le comportement des algorithmes dans le cadre d'un schéma d'itération de la politique optimiste : apprendre

5. L'évaluation de la performance de la politique gloutonne courante étant initialisée dans un état aléatoire proche de l'équilibre.

en contrôlant induit des dynamiques non-stationnaires des fonctions de valeur. Cette tâche consiste à conduire un véhicule sous-puissant en haut d'une route de montagne abrupte, la gravité étant plus forte que le moteur de la voiture. Une deuxième colline se trouve derrière le véhicule qui peut donc s'aider de celle-ci pour prendre de la vitesse (voir Fig. 3.5).

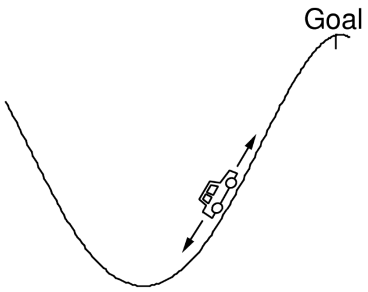


FIGURE 3.5 – *Mountain Car*.

L'état est donné par la position et la vitesse $(x, \dot{x}) \in [-1.2, 0.5] \times [-0.07, 0.07]$. Les trois actions possibles sont : (*i*) aller à gauche (-1), (*ii*) à droite $(+1)$ ou (*iii*) ne rien faire (0). La dynamique du système est donnée par :

$$\begin{cases} \dot{x}_{i+1} = \text{bound}\left[\dot{x}_i + 10^{-3}(a_i - 2.5\cos(3x_i))\right] \\ x_{i+1} = \text{bound}\left[x_i + \dot{x}_{i+1}\right] \end{cases} \tag{3.32}$$

où l'opérateur bound force les bornes de la position et de la vitesse. Quand la position atteint la borne inférieure, la vitesse est mise à zéro. Quand elle atteint la borne supérieure, l'épisode se termine avec une récompense nulle. La récompense est de -1 le reste du temps. Le facteur d'actualisation est fixé à 0.1. L'état est normalisé, et la paramétrisation est composée d'un terme constant et d'un ensemble de 9 noyaux gaussiens équi-répartis (centrés en $\{0, 0.5, 1\} \times \{0, 0.5, 1\}$ et d'écart-type 0.1), cela pour chaque action. Il y a donc 30 fonctions de base.

Cette expérience compare SARSA [242] avec approximation de la fonction de qualité, LSTD [20] et KTD-SARSA, dans un contexte d'itération optimiste de la politique. La politique suivie est ϵ-gloutonne [6], avec $\epsilon = 0.1$. Chaque épisode commence dans un état aléatoire (tirage uniforme sur le domaine). Un maximum de 1500 pas est autorisé.

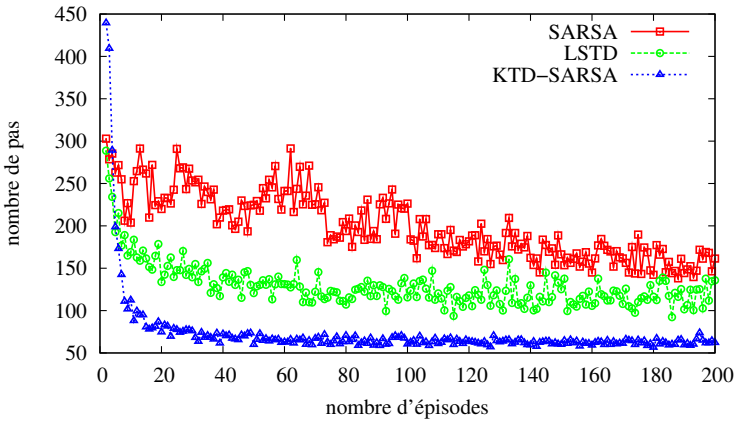


FIGURE 3.6 – *Mountain car*.

Pour chaque essai, l'apprentissage se fait sur 200 épisodes, et la figure 3.6 montre la longueur de chaque épisode moyennée sur 300 essais. KTD-SARSA converge plus rapidement et vers une meilleure po-

6. Gloutonne avec une probabilité 1-ϵ et aléatoire (distribution uniforme sur les actions) avec une probabilité ϵ.

litique que LSTD, qui se comporte mieux que SARSA avec approximation de la Q-fonction. De meilleurs résultats ont peut-être été rapportés dans la littérature pour une paramétrisation de type *tile-coding* [242], cependant la paramétrisation choisie ici est plus brute et implique bien moins de paramètres. Le fait de suivre une politique ϵ-gloutonne implique la non-stationnarité de la Q-fonction apprise, ce qui peut expliquer que LSTD échoue à trouver une politique quasi-optimale. LSTD a été étendu par [119] à LSPI (*Least-Squares Policy Iteration*), qui permet de chercher une politique optimale plus efficacement. Cependant cet algorithme est hors-ligne, et n'implique pas d'apprendre tout en contrôlant, c'est pourquoi il n'est pas comparé ici. KTD-SARSA obtient des politiques optimales très rapidement, après seulement quelques dizaines d'épisodes. L'apprentissage est également plus stable avec l'algorithme proposé.

3.3 Transitions stochastiques

Supposons à présent que les transitions soient stochastiques (les politiques sont toujours supposées déterministes). Nous nous concentrons dans un premier temps sur l'évaluation de la fonction de valeur. L'extension à l'évaluation de la Q-fonction est simple, et l'optimisation directe de la fonction de qualité pose des problèmes particuliers à cause de son aspect *off-policy* (la politique apprise n'est pas la politique suivie). L'équation de Bellman qu'il s'agit maintenant de résoudre est l'espérance de l'équation (2.1) :

$$V^\pi(s) = E_{s'|s,\pi(s)}\left[R(s,\pi(s),s') + \gamma V^\pi(s')\right], \forall s \tag{3.33}$$

Il a été montré dans la section 3.2.5 qu'utiliser directement KTD dans un problème stochastique induit un biais de la fonction de coût minimisée (3.5), ce biais étant très similaire à celui apparaissant lors de la minimisation d'un résidu quadratique de Bellman, comme explicité par [8]. Le calcul exact de ce biais peut être trouvé dans [71], nous en rappelons l'expression ici :

$$\text{trace}\left(K_i E\left[\underset{s'|s_i}{\text{cov}}\left(r_i + \gamma V_\theta(s')\right)|r_{1:i-1}\right]K_i^T\right) = \|K_i\|^2 E\left[\underset{s'|s_i}{\text{cov}}\left(r_i + \gamma V_\theta(s')\right)|r_{1:i-1}\right] \tag{3.34}$$

où K_i est le gain de Kalman, $\|.\|$ est la norme euclidienne usuelle, la covariance dépend des probabilités de transition et l'espérance est sur les paramètres conditionnés aux observations passées. De plus, sous certaines hypothèses (bruits et *a priori* gaussien, canal sans mémoire), l'estimateur de KTD (en utilisant la transformation non-parfumée) est l'estimateur du maximum *a posteriori* (voir [249, chapitre 4.5] pour une preuve dans le cas général d'un filtre de Kalman à sigma-points dont le modèle d'évolution est une marche aléatoire). Il est possible de montrer que cet estimateur du maximum *a posteriori*, sous ces mêmes hypothèses, minimise le coût empirique suivant :

$$C_i(\theta) = \sum_{j=1}^{i} \frac{1}{P_{n_j}}\left(r_j - g_{t_j}(\theta)\right)^2 \tag{3.35}$$

Sous cette forme, le lien avec les problèmes liés aux transitions stochastiques des approches minimisant un résidu quadratique de Bellman est encore plus clair. Dans cette section nous proposons d'étendre KTD avec un modèle de bruit coloré ayant été introduit par [50] pour une approche bayésienne basée sur une modélisation de la fonction de valeur par un processus gaussien.

3.3.1 Un modèle de bruit coloré

La politique étant fixée dans un contexte d'évaluation, le PDM se réduit à une chaîne de Markov valuée dont la probabilité de transition est donnée par $p^\pi(.|s) = p(.|s,\pi(s))$ et dont la récompense est $R^\pi(s,s') = R(s,\pi(s),s')$. La fonction de valeur peut être définie comme l'espérance (sur l'ensemble des trajectoires possibles) du processus aléatoire du cumul pondéré de récompenses suivant :

$$D^\pi(s) = R^\pi(s,s') + \gamma D^\pi(s'),\ s' \sim p^\pi(.|s) \tag{3.36}$$

Ce processus aléatoire peut se décomposer en deux parties, la fonction de valeur et un résidu aléatoire centré :

$$D^\pi(s) = V^\pi(s) + \Delta V^\pi(s) \tag{3.37}$$

où, par définition, $V^\pi(s) = E[D^\pi(s)]$ et $\Delta V^\pi(s) = D^\pi(s) - V^\pi(s)$ est le résidu. En injectant l'équation (3.37) dans l'équation (3.36), la récompense peut être exprimée comme une fonction de la valeur plus un bruit :

$$R^\pi(s, s') = V^\pi(s) - \gamma V^\pi(s') + N(s, s') \tag{3.38}$$

le bruit étant défini par :

$$N(s, s') = \Delta V^\pi(s) - \gamma \Delta V^\pi(s') \tag{3.39}$$

Comme [50], nous supposons les résidus indépendants, ce qui mène à un modèle de bruit coloré.

3.3.2 Extension de KTD

Rappelons l'équation d'observation de la formulation espace-d'état (3.4) : $r_i = g_{t_i}(\theta_i) + n_i$. Dans le cadre de travail de KTD, le bruit d'observation n_i est supposé blanc, ce qui est nécessaire à l'obtention de l'algorithme final. Dans la version étendue de KTD (XKTD pour *eXtended Kalman Temporal Differences*), le modèle de bruit coloré (3.39) est utilisé en lieu et place de ce bruit blanc.

Les résidus étant centrés et supposés indépendants, ce bruit est en fait un bruit à moyenne mobile [7] (bruit MA pour *Moving Average*) qui est la somme de deux bruits blancs :

$$n_i = -\gamma u_i + u_{i-1}, \quad u_i \sim (0, \sigma_i^2) \tag{3.40}$$

Notons que le bruit blanc u_i est centré de variance σ_i^2, cependant aucune supposition n'est faite à propos de sa distribution (et particulièrement toujours pas d'hypothèse gaussienne). S'il est assez aisé d'utiliser, dans le cadre du filtrage de Kalman, un modèle de bruit d'observation *Auto-Régressif* (AR) [8] en étendant l'équation d'évolution (voir [235] par exemple), le cas d'un bruit d'observation MA n'est pas traité dans la littérature usuelle, autant que nous le sachions.

Redériver KTD dans le cas d'un bruit MA serait bien trop difficile. Nous proposons plutôt d'exprimer le bruit MA scalaire n_i comme un bruit AR vectoriel. Cela permet d'étendre le modèle espace-d'état (3.4) à un nouveau modèle pour lequel l'algorithme 1 s'applique assez directement. Soit w_i une variable aléatoire auxiliaire. Le bruit MA scalaire (3.40) est équivalent au bruit AR vectoriel suivant :

$$\begin{pmatrix} w_i \\ n_i \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} w_{i-1} \\ n_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ -\gamma \end{pmatrix} u_i \tag{3.41}$$

Le bruit $u'_i = \begin{pmatrix} u_i & -\gamma u_i \end{pmatrix}^T$ est également centré et sa matrice variance est donnée par :

$$P_{u'_i} = \sigma_i^2 \begin{pmatrix} 1 & -\gamma \\ -\gamma & \gamma^2 \end{pmatrix} \tag{3.42}$$

Cette nouvelle formulation du bruit d'observation ayant été définie, il est possible d'étendre la formulation espace-d'état (3.4) :

$$\begin{cases} \mathbf{x}_i = F\mathbf{x}_{i-1} + v'_i \\ r_i = g_{t_i}(\mathbf{x}_i) \end{cases} \tag{3.43}$$

Le vecteur de paramètres est maintenant étendu avec le bruit AR vectoriel $\begin{pmatrix} w_i & n_i \end{pmatrix}^T$:

$$\mathbf{x}_i^T = \begin{pmatrix} \theta_i^T & w_i & n_i \end{pmatrix} \tag{3.44}$$

La matrice d'évolution F prend en compte la structure du bruit d'observation MA (sous sa forme AR). Soit p le nombre de paramètres et I_p la matrice identité de taille p, la matrice d'évolution s'écrit par bloc ($\mathbf{0}$ étant un vecteur colonne de taille $p \times 1$) :

$$F = \begin{pmatrix} I_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & 0 & 0 \\ \mathbf{0}^T & 1 & 0 \end{pmatrix} \tag{3.45}$$

7. De façon générale, un bruit à moyenne mobile y_k est défini par $y_k = \sum_{j=0}^{q} b_j u_{k-j}$ où $(u_k)_k$ est un bruit blanc et $b_0, \ldots, b_q$ est un ensemble de coefficients.

8. De façon générale, un bruit AR y_k est défini par $y_k = \sum_{j=1}^{q} a_i y_{k-i} + u_k$ où $(u_k)_k$ est un bruit blanc et $a_1, \ldots, a_q$ sont des coefficients.

Le bruit d'évolution v_i est également étendu pour prendre en compte le bruit d'observation coloré. Il est toujours centré, cependant la matrice de variance est maintenant définie par :

$$P_{v'_i} = \begin{pmatrix} P_{v_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & \sigma_i^2 & -\gamma\sigma_i^2 \\ \mathbf{0}^T & -\gamma\sigma_i^2 & \gamma^2\sigma_i^2 \end{pmatrix} \tag{3.46}$$

L'équation d'observation reste la même :

$$r_i = g_{t_i}(\mathbf{x}_i) = g_{t_i}(\theta_i) + n_i \tag{3.47}$$

Néanmoins le bruit d'observation fait maintenant partie intégrante de l'équation d'évolution, au même titre que les paramètres.

En utilisant cette nouvelle formulation espace-d'état, l'algorithme général XKTD peut être dérivé. Il est résumé dans l'algorithme 3, qui est très similaire à l'algorithme 1. L'unique différence (excepté le fait que le modèle espace-d'état ne soit pas le même) est l'étape de prédiction : la prédiction de la moyenne et de la variance du vecteur aléatoire étendu $\mathbf{x}_i$ est faite en utilisant la matrice d'évolution F (cette matrice étant l'identité pour KTD). Notons que le coût computationnel est le même pour KTD et son extension XKTD, étant donné que le vecteur de paramètres est étendu avec seulement deux variables. Comme pour KTD, XKTD peut être spécialisé en XKTD-V (évaluation de la fonction de valeur) et XKTD-SARSA (évaluation de la fonction de qualité). Le raisonnement est exactement le même que celui développé dans la section 3.2 et n'est pas répété ici. La spécialisation à XKTD-Q n'est en revanche pas évidente en raison de son aspect *off-policy*.

Algorithme 3 : Algorithme XKTD général.

Initialisation : *a priori* $\hat{\mathbf{x}}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition t_i et la récompense r_i ;

Phase de prédiction;
$\hat{\mathbf{x}}_{i|i-1} = F\hat{\mathbf{x}}_{i-1|i-1}$;
$P_{i|i-1} = FP_{i-1|i-1}F^T + P_{v'_i}$;

Calcul des statistiques d'intérêt (en utilisant l'UT et les statistiques $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$);
$\hat{r}_{i|i-1} = E[g_{t_i}(\theta_i) + n_i | r_{1:i-1}]$;
$P_{\mathbf{x}r_i} = E\left[(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})(g_{t_i}(\theta_i) + n_i - \hat{r}_{i|i-1}) | r_{1:i-1}\right]$;
$P_{r_i} = E\left[(g_{t_i}(\theta_i) + n_i - \hat{r}_{i|i-1})^2 | r_{1:i-1}\right]$;

Phase de correction;
$K_i = P_{\mathbf{x}r_i} P_{r_i}^{-1}$;
$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

3.3.3 Extension aux traces d'éligibilité

Les traces d'éligibilités sont un modèle largement répandu dans la littérature sur l'Apprentissage par Renforcement [242] qui a pour but de propager sur plusieurs état l'expérience d'une transition. Les effets de ces traces sont, par exemple, une réduction de la variance dans les résultats obtenu par apprentissage. Dans la section 3.3.1, nous avons introduit un modèle génératif de la récompense basé sur la fonction de valeur en deux états consécutifs plus un bruit coloré composé de deux résidus toujours supposés blancs. Nous en rappelons l'expression :

$$R^\pi(s, s') = V^\pi(s) - \gamma V^\pi(s') + (\Delta V^\pi(s) - \gamma \Delta V^\pi(s')) \tag{3.48}$$

Nous notons ici $N^{(1)}$ le bruit composé des deux résidus :

$$N^{(1)}(s,s') = \Delta V^\pi(s) - \gamma \Delta V^\pi(s') \tag{3.49}$$

Rappelons que le principe des traces d'éligibilité (en "vue avant") consiste à considérer une moyenne géométrique des erreurs TD à k pas (voir [242]). Nous avons fait de même avec ce modèle comportant un terme de bruit [73]. Ainsi, pour une prédiction à deux pas nous avons :

$$R^\pi(s,s') + \gamma R^\pi(s',s'') = V^\pi(s) - \gamma^2 V^\pi(s'') + \left(\Delta V^\pi(s) - \gamma^2 \Delta V^\pi(s'')\right) \tag{3.50}$$

De manière générale, nous notons $N^{(k)}$ le bruit correspondant à une prédiction à k pas :

$$N^{(k)}(s,s^{(k)}) = \Delta V^\pi(s) - \gamma^k \Delta V^\pi(s^{(k)}) \tag{3.51}$$

Ainsi, on peut généralement considérer une prédiction à k pas :

$$\sum_{i=0}^{k-1} \gamma^i R^\pi(s^{(i)}, s^{(i+1)}) = V^\pi(s) - \gamma^k V^\pi(s^{(k)}) + N^{(k)}(s,s^{(k)}) \tag{3.52}$$

Une nouvelle classe de bruits

Pour l'approche classique des traces d'éligibilité, c'est la moyenne géométrique des TD d'ordre k qui est considérée. Ici, nous allons plutôt nous intéresser à la moyenne géométrique des bruits d'ordre k. C'est une vision différente d'une même chose, dans la mesure où le bruit d'observation peut être vu comme un équivalent probabiliste de l'erreur de TD. Par exemple, pour la prédiction d'ordre k nous pouvons écrire :

$$\begin{aligned} &\sum_{i=0}^{k-1} \gamma^i R^\pi(s^{(i)}, s^{(i+1)}) = V^\pi(s) - \gamma^k V^\pi(s^{(k)} + N^{(k)}(s,s^{(k)}) \\ \Leftrightarrow\ & N^{(k)}(s,s^{(k)}) = \sum_{i=0}^{k-1} \gamma^i R^\pi(s^{(i)}, s^{(i+1)}) + \gamma^k V^\pi(s^{(k)}) - V^\pi(s) \end{aligned} \tag{3.53}$$

Nous définissons N^λ cette moyenne géométrique des bruits :

$$\begin{aligned} N^\lambda &= (1-\lambda) \sum_{k=1}^{\infty} \lambda^{k-1} N^{(k)} \\ &= \Delta V^\pi(s) - \gamma(1-\lambda)\left(\Delta V^\pi(s') + \cdots + (\gamma\lambda)^{k-1} \Delta V^\pi(s^{(k)}) + \dots\right) \end{aligned} \tag{3.54}$$

Comme pour la dérivation de XKTD nous supposons que les résidus sont blancs. Cette hypothèse forte a déjà été discutée. Nous notons u_i le bruit blanc centré de variance théorique $\Delta V^\pi(s^{(i)})$. Un modèle de bruit d'observation possible (et équivalent au bruit (3.54) sous les hypothèses données) est donc le suivant :

$$n_i = u_i - \gamma(1-\lambda)\left(u_{i+1} + \gamma\lambda u_{i+2} + \cdots + (\gamma\lambda)^{k-1} u_{i+k} + \dots\right) \tag{3.55}$$

Un premier problème est que ce bruit est non-causal et est donc difficile à intégrer dans un filtre de Kalman, par essence causal. Cependant, ce sont les corrélations induites par ce bruit qui nous intéressent. Si l'on suppose la variance du bruit blanc constante, cette corrélation n'est pas affectée par la causalité [9]. Nous supposons donc la variance des résidus constante, ce qui permet de réécrire le bruit de manière causale [10] :

$$n_i = u_i - \gamma(1-\lambda)\left(u_{i-1} + \gamma\lambda u_{i-2} + \cdots + (\gamma\lambda)^{k-1} u_{i-k} + \dots\right) \tag{3.56}$$

Cette hypothèse de variance constante des résidus peut paraître forte. Cependant, les analyses de convergences (disponibles dans [71] et [75]) montrent que pour les fonctions de coût minimisées par KTD et XKTD, ce terme de variance n'apparaît qu'en tant que pondération des coûts quadratiques instantanés. Cette hypothèse porte donc moins à conséquence que l'on pourrait le supposer *a priori*. De plus, pratiquement, nous considérons toujours ce bruit stationnaire.

9. Le bruit non-causal et son équivalent causal ont les mêmes corrélations.
10. C'est le pendant des vues avant et arrière des traces d'éligibilité.

Intégration à Kalman

Le bruit n_i ayant été introduit, il reste encore à l'intégrer dans KTD. La technique est similaire à celle utilisée pour l'obtention de XKTD. Il faut d'abord remarquer que n_i peut se décomposer en la somme d'un bruit blanc u_i et d'un bruit autorégressif b_i :

$$n_i = u_i - \gamma(1-\lambda)b_{i-1} \tag{3.57}$$

$$\begin{aligned} \text{avec } b_i &= u_i + \gamma\lambda u_{i-1} + \cdots + (\gamma\lambda)^k u_{i-k} + \dots \\ &= u_i + \gamma\lambda b_{i-1} \end{aligned} \tag{3.58}$$

Nous utilisons ensuite les deux bruits n_i et b_i pour introduire un bruit auto-régressif vectoriel $\begin{pmatrix} b_i & n_i \end{pmatrix}^T$:

$$\begin{pmatrix} b_i \\ n_i \end{pmatrix} = \begin{pmatrix} \gamma\lambda & 0 \\ -\gamma(1-\lambda) & 0 \end{pmatrix} \begin{pmatrix} b_{i-1} \\ n_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_i \tag{3.59}$$

Pour intégrer ce bruit coloré à KTD, il suffit d'étendre le modèle espace d'état comme cela a déjà été fait auparavant, ce qui donne le modèle suivant :

$$\begin{cases} \mathbf{x}_i = F(\lambda)\mathbf{x}_{i-1} + v'_i \\ r_i = g_{t_i}(\mathbf{x}_i) \end{cases} \tag{3.60}$$

Nous détaillons chacun de ces termes. Le vecteur de paramètres est maintenant étendu avec le bruit AR vectoriel $\begin{pmatrix} b_i & n_i \end{pmatrix}^T$:

$$\mathbf{x}_i = \begin{pmatrix} \theta_i \\ b_i \\ n_i \end{pmatrix} \tag{3.61}$$

La matrice d'évolution $F(\lambda)$ rend compte de la structure du nouveau bruit d'observation. Nous rappelons que I_p désigne la matrice identité de taille $p \times p$:

$$F(\lambda) = \begin{pmatrix} I_p & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & \gamma\lambda & 0 \\ \mathbf{0}^T & -\gamma(1-\lambda) & 0 \end{pmatrix} \tag{3.62}$$

Le bruit d'évolution v_i est également étendu de façon à prendre en compte ce bruit coloré, $v'_i = \begin{pmatrix} v_i & u_i & u_i \end{pmatrix}^T$; ce nouveau bruit d'évolution est centré et sa matrice de variance $P_{v'_i}$ peut s'écrire par blocs en utilisant le fait que les bruits v_i et u_i sont indépendants (et donc décorrélés) :

$$v'_i = \begin{pmatrix} v_i \\ u_i \\ u_i \end{pmatrix} \sim \left(\begin{pmatrix} \mathbf{0} \\ 0 \\ 0 \end{pmatrix}, P_{v'_i} = \begin{pmatrix} P_{v_i} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}^T & \sigma_i^2 & \sigma_i^2 \\ \mathbf{0}^T & \sigma_i^2 & \sigma_i^2 \end{pmatrix} \right) \tag{3.63}$$

Enfin, l'équation d'observation reste la même, malgré la notation quelque peu abusive maintenant connue :

$$r_i = g_{t_i}(\mathbf{x}_i) = g_{t_i}(\theta_i) + n_i \tag{3.64}$$

Comme précédemment, le bruit d'observation est une part de l'équation d'évolution, il doit donc être estimé.

L'approche résultante est résumée dans l'algorithme 4, très similaire à l'algorithme qui correspond à XKTD (la seule différence résidant dans la matrice d'évolution utilisée).

Une fois encore la transformation non-parfumée présentée dans la Section 2.2.4 peut être utilisée pour dériver des algorithmes pratiques. L'approche est la même que pour XKTD, présentée Section 3.3, nous donnons simplement l'algorithme 5 correspondant. Notons que pour la même raison que nous n'avons pas défini d'extension XKTD-Q, c'est-à-dire le caractère *off-policy* des algorithmes basés sur l'itération de la valeur, nous ne proposons pas d'extension KTD-Q(λ). Les algorithmes permettant respectivement l'évaluation de la valeur et de la fonction de qualité (soit KTD-$V(\lambda)$ et KTD-SARSA(λ)) sont donnés.

Algorithme 4 : KTD(λ) général

Initialisation : *a priori* $\hat{\mathbf{x}}_{0|0}$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition t_i ainsi que la récompense r_i ;

Phase de prédiction;
$\hat{\mathbf{x}}_{i|i-1} = F(\lambda)\hat{\mathbf{x}}_{i-1|i-1}$;
$P_{i|i-1} = F(\lambda)P_{i-1|i-1}F(\lambda)^T + P_{v'_i}$;

Calcul des statistiques d'intérêt;
$\hat{r}_{i|i-1} = E[g_{t_i}(\theta_i) + n_i | r_{1:i-1}]$;
$P_{\mathbf{x}r_i} = E\left[(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})(g_{t_i}(\theta_i) + n_i - \hat{r}_{i|i-1})|r_{1:i-1}\right]$;
$P_{r_i} = E\left[(g_{t_i}(\theta_i) + n_i - \hat{r}_{i|i-1})^2|r_{1:i-1}\right]$;

Phase de correction;
$K_i = P_{\mathbf{x}r_i}P_{r_i}^{-1}$;
$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

Algorithme 5 : KTD-$V(\lambda)$ et KTD-SARSA(λ)

Initialisation : *a priori* $\hat{\mathbf{x}}_{0|0} = \begin{pmatrix}\hat{\theta}_{0|0}^T & 0 & 0\end{pmatrix}^T$ et $P_{0|0}$;

pour $i \leftarrow 1, 2, \ldots$ **faire**

Observer la transition $t_i = \begin{cases}(s_i, s_{i+1}) \text{ (KTD-V}(\lambda)\text{)} \\ (s_i, a_i, s_{i+1}, a_{i+1}) \text{ (KTD-SARSA}(\lambda)\text{)}\end{cases}$ ainsi que la récompense associée r_i ;

Phase de prédiction;
$\hat{\mathbf{x}}_{i|i-1} = F(\lambda)\hat{\mathbf{x}}_{i-1|i-1}$;
$P_{i|i-1} = F(\lambda)P_{i-1|i-1}F(\lambda)^T + P_{v'_{i-1}}$;

Calcul des sigma-points ;
$\mathbf{X}_{i|i-1} = \left\{\hat{\mathbf{x}}_{i|i-1}^{(j)}, \quad 0 \leq j \leq 2p+4\right\}$ (en utilisant $\hat{\mathbf{x}}_{i|i-1}$ et $P_{i|i-1}$);
$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p+4 \quad\}$;
`/* notons que` $(\hat{\mathbf{x}}_{i|i-1}^{(j)})^T = \left((\hat{\theta}_{i|i-1}^{(j)})^T \quad \hat{b}_{i|i-1}^{(j)} \quad \hat{n}_{i|i-1}^{(j)}\right)$ `*/`

$$\mathcal{R}_{i|i-1} = \begin{cases}\text{(KTD-V}(\lambda)\text{)} \\ \left\{\hat{r}_{i|i-1}^{(j)} = \hat{V}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i) - \gamma\hat{V}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}) + \hat{n}_{i|i-1}^{(j)},\ 0 \leq j \leq 2p+4\right\} \\ \text{(KTD-SARSA}(\lambda)\text{)} \\ \left\{\hat{r}_{i|i-1}^{(j)} = \hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_i, a_i) - \gamma\hat{Q}_{\hat{\theta}_{i|i-1}^{(j)}}(s_{i+1}, a_{i+1}) + \hat{n}_{i|i-1}^{(j)},\ 0 \leq j \leq 2p+4\right\}\end{cases} ;$$

Calcul des statistiques d'intérêt;
$\hat{r}_{i|i-1} = \sum_{j=0}^{2p+4} w_j \hat{r}_{i|i-1}^{(j)}$;
$P_{\mathbf{x}r_i} = \sum_{j=0}^{2p+4} w_j (\hat{\mathbf{x}}_{i|i-1}^{(j)} - \hat{\mathbf{x}}_{i|i-1})(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1})$;
$P_{r_i} = \sum_{j=0}^{2p+4} w_j \left(\hat{r}_{i|i-1}^{(j)} - \hat{r}_{i|i-1}\right)^2$;

Phase de correction;
$K_i = P_{\mathbf{x}r_i}P_{r_i}^{-1}$;
$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + K_i\left(r_i - \hat{r}_{i|i-1}\right)$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

Lien avec KTD et XKTD

Nous étudions à présent deux cas limites de KTD(λ), lorsque $\lambda = 0$ et $\lambda = 1$. Rappelons la forme du bruit introduit :

$$\begin{pmatrix} b_i \\ n_i \end{pmatrix} = \begin{pmatrix} \gamma\lambda & 0 \\ -\gamma(1-\lambda) & 0 \end{pmatrix} \begin{pmatrix} b_{i-1} \\ n_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_i \tag{3.65}$$

Considérons tout d'abord le cas où $\lambda = 1$. Nous obtenons :

$$\begin{pmatrix} b_i \\ n_i \end{pmatrix} = \begin{pmatrix} \gamma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} b_{i-1} \\ n_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_i \tag{3.66}$$

Dans ce cas, le bruit auto-régressif b_i est toujours estimé mais ce n'est plus une composante du bruit d'observation n_i, qui se réduit au bruit blanc u_i. Nous obtenons exactement l'algorithme KTD. Considérons l'autre cas extrême, pour lequel $\lambda = 0$. Nous avons alors :

$$\begin{pmatrix} b_i \\ n_i \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ -\gamma & 0 \end{pmatrix} \begin{pmatrix} b_{i-1} \\ n_{i-1} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u_i \tag{3.67}$$

Dans ce cas, b_i joue le rôle de variable aléatoire auxiliaire (dont le rôle est la mémorisation du bruit blanc à l'instant précédent) et l'on retrouve le modèle de bruit à moyenne mobile utilisé pour XKTD. Remarquons que si les modèles de bruit ne sont pas *stricto sensu* les mêmes, les corrélations induites sont identiques (sous hypothèse de variance constante des résidus).

Ainsi, nous avons montré que les approches proposées dans les Sections 3.2 et 3.3 sont des cas particuliers de KTD(λ). Notons tout de même que contrairement à l'usage, ici c'est $\lambda = 0$ qui correspond à la méthode se rapprochant le plus de Monte Carlo (soit XKTD) et non $\lambda = 1$, qui correspond ici à la mise à jour la plus locale. Pour se ramener à des notations plus usuelles il suffirait de remplacer λ par $1 - \lambda$ dans la matrice d'évolution $F(\lambda)$.

3.3.4 Résultats expérimentaux

Dans cette section, un certain nombre d'expérimentations est proposé. La première illustre la réduction de biais lors de l'évaluation de la fonction de valeur d'une simple (mais non-stationnaire) chaîne de Markov valuée. Les effets des traces d'éligibilité sont aussi analysés. Les différentes variantes de KTD (ainsi que TD et LSTD) sont ensuite comparées sur une version stochastique du problème bien connu du *mountain-car*.

Chaîne de Boyan

La chaîne de Boyan a déjà été décrite dans la Section 3.2.5. Le facteur d'actualisation γ est fixé à 1 pour cette tâche épisodique. KTD a été pensé pour pouvoir prendre en compte les environnements non-stationnaires, et XKTD devrait également présenter cette caractéristique. Pour simuler un changement dans le PDM, le signe de la récompense est inversé à partir du $100^{\text{ème}}$ épisode. La fonction de valeur optimale est toujours linéaire en les bases, et le vecteur optimal de paramètres correspondant est $\theta^*_{(+)} = -\theta^*_{(-)}$ après le changement du PDM. L'apprentissage est fait sur 200 épisodes, et les résultats sont moyennées sur 100 essais (chaque essai correspond à 200 épisodes). Les résultats sont présentés sur la figure 3.7.

Avant le changement de PDM, KTD converge plus rapidement que LSTD, qui converge plus rapidement que TD. Cependant, comme prévu, KTD est biaisé. L'algorithme XKTD converge aussi rapidement que KTD, cependant sans être biaisé. Après le changement dans le PDM, les résultats sont similaires à ceux présentés dans la section 3.2.5. LSTD échoue à s'adapter au nouveau PDM, TD fonctionne mieux grâce à sont taux d'apprentissage constant (un taux d'apprentissage ne décroissant pas trop vite aurait suffit) et KTD s'adapte encore plus rapidement. XKTD fait la même chose, le biais en moins. Notons que le choix du bruit a son importance. S'il y a trop de bruit d'évolution, l'adaptation sera rapide mais l'apprentissage sera relativement instable, particulièrement avec des transitions stochastiques [11] et si le bruit est trop

11. L'aspect aléatoire des transitions peut être alors interprétée comme un changement immédiat dans le PDM.

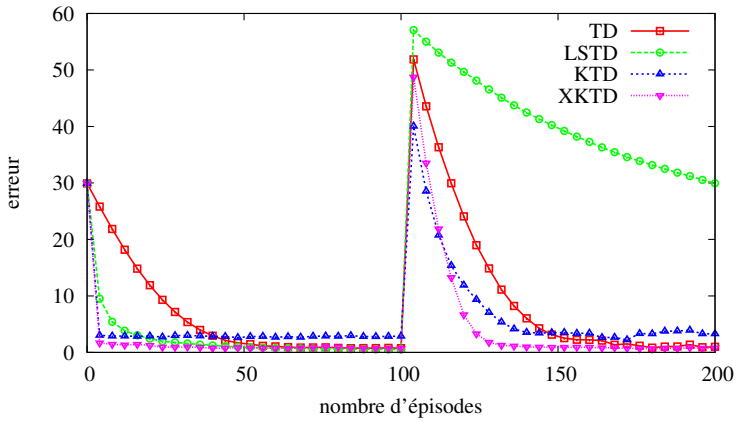


FIGURE 3.7 – Chaîne de Boyan : Résultats de XKTD.

faible, l'adaptation sera lente. C'est une forme de dilemme entre plasticité et stabilité et le même type de problème se pose en choisissant un taux d'apprentissage.

Nous analysons ensuite le comportement de KTD(λ). Dans un premier temps c'est au biais ainsi qu'à la vitesse de convergence de KTD(λ) que nous nous intéressons pour différentes valeurs de λ dans un version stationnaire de la chaîne. Nous considérons également les algorithmes TD et LSTD comme étalon. La même paramétrisation et le même facteur d'actualisation sont utilisés. Il convient de se référer à [71] pour plus de détails sur les paramètres d'initialisation de ces algorithmes le propos étant ici d'illustrer les propriétés de KTD. L'apprentissage se fait sur 100 épisodes et les résultats de la figure 3.8 sont moyennés sur 300 essais.

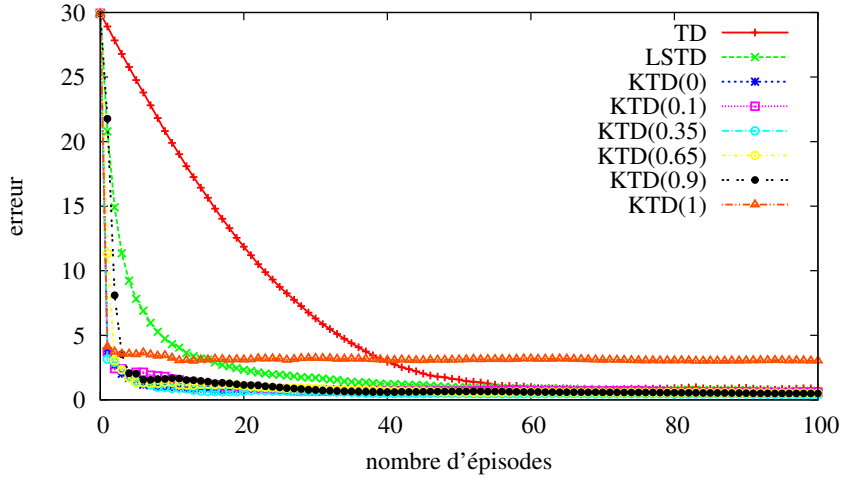


FIGURE 3.8 – Chaîne de Boyan, cas stochastique et stationnaire.

D'après cette expérience, la vitesse de convergence de KTD(λ) est sensiblement la même pour les différentes valeurs de λ et le seul algorithme qui semble être biaisé est KTD(1), soit l'algorithme KTD introduit à la Section 3.1.3. Ce résultat est intéressant : il semble que KTD(λ) réduise le biais de KTD, pour tout λ strictement inférieur à 1. C'est l'un des comportements que nous espérions et il se vérifie.

L'une des motivations principales à l'introduction de KTD est la capacité de l'algorithme à prendre en compte les non-stationnarités et ce surtout dans le cadre d'une itération de la politique généralisée. Nous testons cela sur la chaîne de Boyan que l'on rend une fois encore non-stationnaire en inversant le signe de la récompense comme précédemment. L'apprentissage se fait sur 140 épisodes (changement du signe de la récompense au $70^{\text{ème}}$ épisode) et les résultats de la figure 3.9 sont moyennés sur 300 essais.

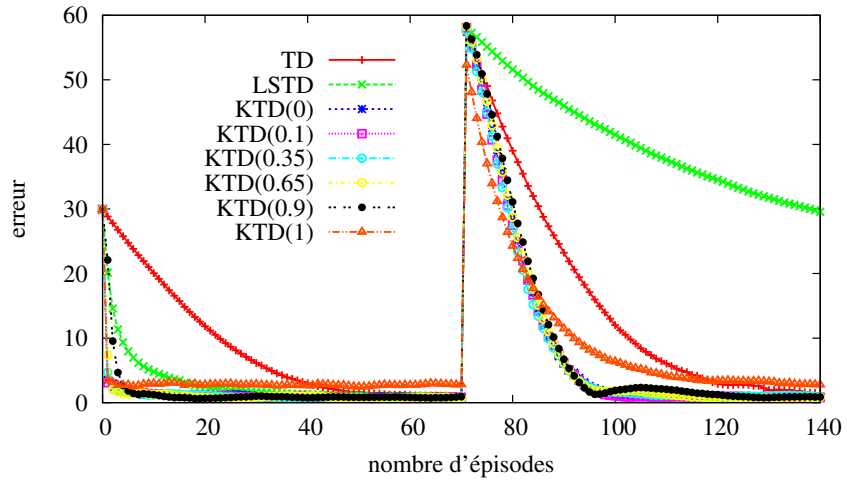


FIGURE 3.9 – Chaîne de Boyan, cas stochastique et non-stationnaire.

Cette expérience montre que KTD(λ) permet de prendre en compte la non-stationnarité de l'environnement et ce pour toute valeur de λ. La seule version de KTD à rester biaisée après le changement de récompense (et la période d'adaptation) est KTD(1), comme précédemment. Sur cet exemple, l'adaptation est sensiblement la même pour les différentes valeurs de λ (exception faite peut-être de $\lambda = 0.9$, qui présente un petit sursaut après le centième épisode que nous ne savons pas expliquer).

Nous avons jusqu'à présent étudié l'erreur moyenne en fonction du nombre d'épisodes. Nous allons à présent nous intéresser à cette erreur moyenne, ainsi qu'à l'écart-type associé, en les voyant comme des fonctions de λ, pour un nombre d'épisodes donnés. L'idée est ici d'étudier l'influence de λ d'une part sur l'erreur moyenne et d'autre part sur la variance associée. Les paramètres sont les mêmes que précédemment, sauf le bruit d'évolution sur les paramètres qui est choisi nul. Les statistiques que nous donnons sont faites sur 1000 essais, chaque essai correspondant à 500 épisodes. L'algorithme considéré est KTD(λ) pour λ prenant 21 valeurs équi-réparties entre 0 et 1. Sur la figure 3.10 nous traçons l'erreur moyenne et l'écart-type associé au bout de 50 épisodes et au bout de 500 épisodes (donc moyenne et écart-type à partir de 1000 éléments), tout cela en fonction du facteur d'éligibilité λ. Notons l'échelle semi-logarithmique.

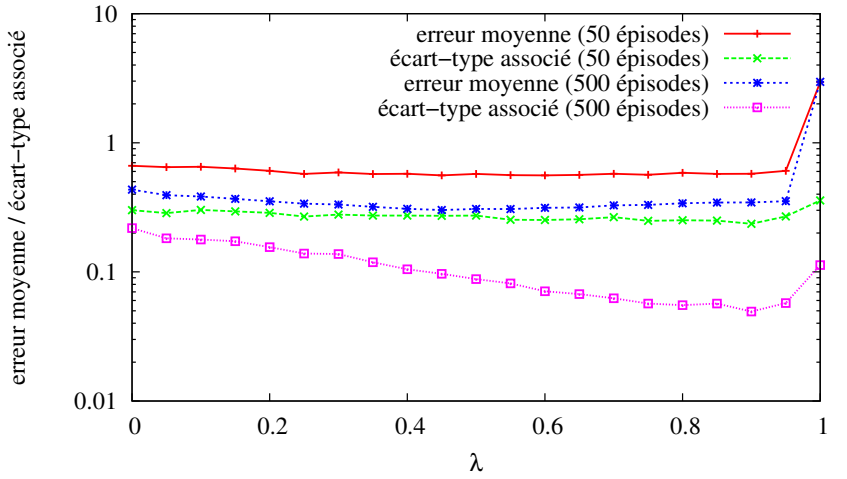


FIGURE 3.10 – Chaîne de Boyan, erreur moyenne et écart-type associé.

Au bout de 50 épisodes, il ne semble pas y avoir d'influence significative du facteur d'éligibilité sur

l'erreur moyenne ou l'écart-type associé, sauf pour $\lambda = 1$ qui correspond à KTD et donc à un estimateur biaisé. Au bout de 500 épisodes, λ ne semble toujours pas influer sur l'erreur moyenne. Notons que pour $\lambda = 1$ la valeur de l'erreur moyenne est la même pour 50 ou 500 épisodes. Le biais est donc atteint tôt. Cependant, au bout de 500 épisodes, le facteur d'éligibilité a une influence sur l'écart-type de l'erreur, avec un minimum autour de $\lambda = 0.9$. Ainsi, l'extension de KTD aux traces d'éligibilité proposée pourrait permettre de réduire la variance de l'estimateur correspondant, au moins asymptotiquement, ce qui peut s'avérer intéressant. De plus cette variance semble la plus faible pour les valeurs proches de $\lambda = 1$, ce qui correspond à une solution non-biaisée d'une part et de relativement locale d'autre part, c'est donc le cas qui nous intéresse *a priori* le plus dans un contexte d'itération de la politique généralisée.

3.4 Gestion de l'incertitude

3.4.1 Principe

Les paramètres étant modélisés comme des variables aléatoires, pour n'importe quel état donné la fonction de valeur paramétrée est également une variable aléatoire. Ce modèle statistique permet de calculer les moyenne et variance associées. Soit $\hat{V}_\theta$ la fonction de valeur approchée, paramétrée par le vecteur aléatoire θ de moyenne $\bar{\theta}$ et de matrice de variance P_θ. Soient $\hat{V}_\theta(s)$ et $\hat{\sigma}^2_{V_\theta}(s)$ les moyenne et variance associées, pour un état s donné. Une première étape pour propager l'information d'incertitude des paramètres vers la valeur de l'état considéré est de calculer les sigma-points associés au vecteur de paramètre, c'est-à-dire $\Theta = \{\theta^{(j)},\ 0 \leq j \leq 2p\}$, ainsi que les poids correspondants, à partir de $\bar{\theta}$ et P_θ, tel qu'expliqué dans la section précédente. Ensuite, les images de ces sigma-points sont calculés en utilisant la fonction de valeur paramétrée : $\mathcal{V}_\theta(s) = \{\hat{V}_\theta^{(j)}(s) = \hat{V}_{\theta^{(j)}}(s), \quad 0 \leq j \leq 2p\}$. Connaissant ces images et les poids associés, les statistiques d'intérêt peuvent être approchées :

$$\begin{cases} \bar{V}_\theta(s) = \sum_{j=0}^{2p} w_j \hat{V}_\theta^{(j)}(s) \\ \hat{\sigma}^2_{V_\theta}(s) = \sum_{j=0}^{2p} w_j (\hat{V}_\theta^{(j)}(s) - \bar{V}_\theta(s))^2 \end{cases} \tag{3.68}$$

Cela est illustré sur la figure 3.11 et l'extension à la Q-fonction est évidente. Ainsi, à chaque pas de temps, une information d'incertitude peut être estimée dans le cadre de travail des KTD.

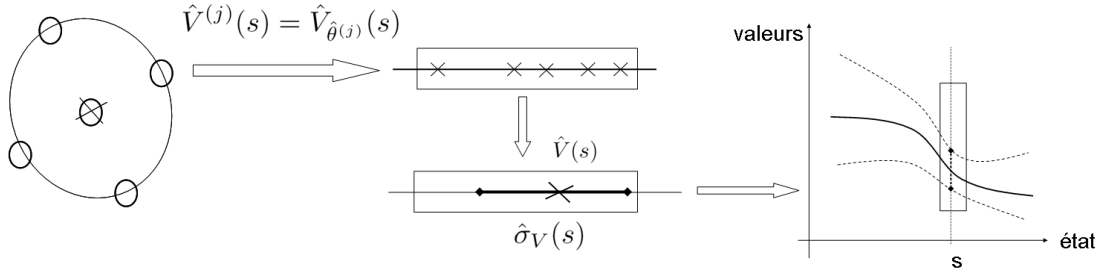


FIGURE 3.11 – Calcul de l'incertitude.

3.4.2 Illustration

La première expérience vise à illustrer l'information d'incertitude disponible sur une simple problème de labyrinthe. L'espace d'état bi-dimensionnel et continu est le carré unité $(x, y) \in [0, 1]^2$. Les actions consistent à se déplacer dans les quatre directions (haut, bas, gauche, droite), l'amplitude du déplacement étant de $0,05$ dans chaque cas. La récompense est de $+1$ si l'agent quitte le labyrinthe dans la zone $\{x \in [\frac{3}{8}, \frac{6}{8}], y = 1\}$, -1 si l'agent le quitte dans la zone $\{x \in [0, \frac{3}{8}[\cup]\frac{6}{8}, 1], y = 1\}$, 0 sinon. L'algorithme considéré est KTD-V (c'est-à-dire que l'équation d'évaluation de Bellman pour la fonction de valeur (2.2) est considérée). La fonction de valeur est un réseau RBF [12], plus précisément neuf noyaux gaussiens équi-répartis (centrés en $\{0, 0.5, 1\} \times \{0, 0.5, 1\}$ et d'écart-type $0,5$). Les paramètres sont donc les poids de

12. *Radial Basis Functions*

chaque gaussienne. Le facteur d'actualisation est fixé à $\gamma = 0,9$. L'agent commence chaque épisode dans une position aléatoire (x_0, y_0) où x_0 est échantillonné selon une distribution gaussienne, $x_0 \sim \mathcal{N}(\frac{1}{2}, \frac{1}{8})$, et y_0 est échantillonné selon une distribution uniforme, $y_0 \sim \mathcal{U}_{[0;0,05]}$. La politique suivie par l'agent (dont la fonction de valeur est apprise par KTD-V) consiste à aller vers le haut avec une probabilité de $0,9$ et à aller dans une des trois autres directions avec probabilité $\frac{0,1}{3}$. Les *a prioris* sont fixés à $\theta_{0|0} = 0$ et $P_{0|0} = 10I$ et les variances des bruits à $P_{n_i} = 1$ et $P_{v_i} = 0I$.

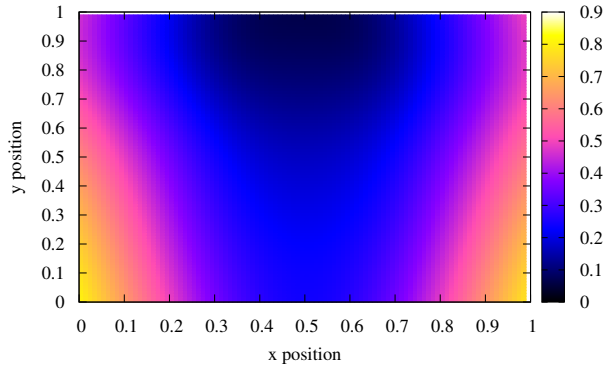


FIGURE 3.12 – Illustration de l'incertitude.

L'apprentissage est fait sur trente épisodes, les résultats étant présentés figure 3.12 qui montre l'écart-type approché de la fonction de valeur comme une fonction des états. Si l'on considère l'abscisse, l'incertitude est plus faible au centre qu'aux bords, ce qui s'explique par le fait que les trajectoires ont lieu plus souvent dans le centre du domaine pendant l'apprentissage (politique considérée et état initial). En considérant l'ordonnée, l'incertitude est plus faible près de la borne supérieure ($y = 1$) que près de la borne inférieure ($y = 0$), ce qui s'explique par le fait que les valeurs rétro-propagées sont d'autant moins certaines qu'elles sont éloignées de la source de la récompense.

3.4.3 Une forme d'apprentissage actif

Principe

Nous expliquons dans cette section comment l'information d'incertitude disponible peut être utilisés dans une forme d'apprentissage actif en particulier dans le cas *off-policy*, c'est à dire dans le cas de l'algorithme KTD-Q. Une question naturelle se pose donc : quelle politique comportementale choisir afin d'accélérer l'apprentissage ? Une réponse triviale, mais impraticable, est de suivre la politique optimale. La solution que nous proposons consiste à échantillonner les actions relativement à l'incertitude de leurs valeurs estimées. Soit i l'index temporel courant. Le système est dans un état s_i et l'agent doit choisir une action a_i. Les prédictions $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$ sont disponibles et peuvent être utilisées pour estimer l'incertitude de de la Q-function paramétrée par $\theta_{i|i-1}$ en l'état s_i, et ce pour toute action a. Soit $\hat{\sigma}^2_{Q_{i|i-1}}(s_i, a)$ la variance associée. Nous proposons d'échantillonner l'action a_i selon la politique (heuristique) suivante :

$$b(.|s_i) = \frac{\hat{\sigma}_{Q_{i|i-1}}(s_i, .)}{\sum_{a \in A} \hat{\sigma}_{Q_{i|i-1}}(s_i, a)} \tag{3.69}$$

Cette politique totalement exploratrice favorise les actions dont le résultat est le moins certain. L'approche correspondante, appelée "KTD-Q actif", est résumée par l'algorithme 6.

Expérimentation

Nous proposons de tester l'apprentissage actif sur la tâche du pendule inversé comme décrite dans la Section 3.2.5. L'algorithme utilise l'information d'incertitude calculée par KTD pour accélérer la conver-

Algorithme 6 : KTD-Q actif

Initialisation : *a prioris* $\hat{\theta}_{0|0}$ et $P_{0|0}$, state s_1;

for $i \leftarrow 1, 2, \ldots$ **do**

Phase de prédiction;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;

Calcul des sigma-points et échantillonnage de l'action;
$\Theta_{i|i-1} = \{\hat{\theta}^{(j)}_{i|i-1}, \quad 0 \leq j \leq 2p\}$;
`/* en utilisant` $\hat{\theta}_{i|i-1}$ `et` $P_{i|i-1}$ `*/`
$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p \quad \}$;
pour $a \in A$ **faire**
$\mathcal{Q}_{i|i-1}(s_i, a) = \{\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a), 0 \leq j \leq 2p\}$;
$\bar{Q}_{i|i-1}(s_i, a) = \sum_{j=0}^{2p} w_j \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a)$;
$\hat{\sigma}^2_{Q_{i|i-1}}(s_i, a) = \sum_{j=0}^{2p} w_j (\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a) - \bar{Q}_{i|i-1}(s_i, a))^2$;
Echantillonner a_i selon $b(.|s_i)$, voir Eq. (3.69);
Observer r_i et s_{i+1} ;
$\mathcal{R}_{i|i-1} = \{\hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i)$
$- \gamma \max_{a \in A} \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a), 0 \leq j \leq 2p\}$;

Calculer les statistiques d'intérêt;
$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;
$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;
$P_{r_i} = \sum_{j=0}^{2p} w_j (\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2 + P_{n_i}$;

Phase de correction;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

gence.

L'agent commence chaque épisode en un état aléatoire correspondant à une légère perturbation de la position d'équilibre. La performance est toujours mesurée comme étant le nombre moyen d'interactions dans un épisode de test (un tel épisode applique la politique gloutonne respectivement à la Q-fonction estimée, l'apprentissage étant figé, et un maximum de 3000 interactions étant autorisé, ce qui correspond à maintenir le pendule à la verticale pendant cinq minutes).

Nous comparons donc l'algorithme KTD-Q, pour lequel les trajectoires sont générées selon une politique totalement aléatoire (c'est-à-dire échantillonnage uniforme des actions, indépendamment de l'état courant) à l'algorithme KTD-Q actif, pour lequel les actions sont échantillonnées en accord avec l'heuristique (3.69) qui utilise l'information d'incertitude disponible, les conditions expérimentales étant les mêmes. Les résultats sont présentés figure 3.13.

Il est à noter que la longueur moyennne d'un épisode lorsque la politique est totalement aléatoire est de dix interactions, tandis que pour la politique comportementale de l'apprentissage actif, elle, est de onze. En conséquence, le nombre d'interactions par épisode n'explique que partiellement l'accélération de l'apprentissage (au plus 10%, bien moins que l'amélioration réelle qui peut atteindre 100% au départ). D'après la figure 3.13, il est clair qu'échantillonner les actions en accord avec la politique (3.69) accélère l'apprentissage. Par exemple, KTD-Q classique nécessite presque 50 épisodes pour atteindre les mêmes performances que KTD-Q actif au bout de 25 épisodes. Ce schéma d'apprentissage actif est donc effi-

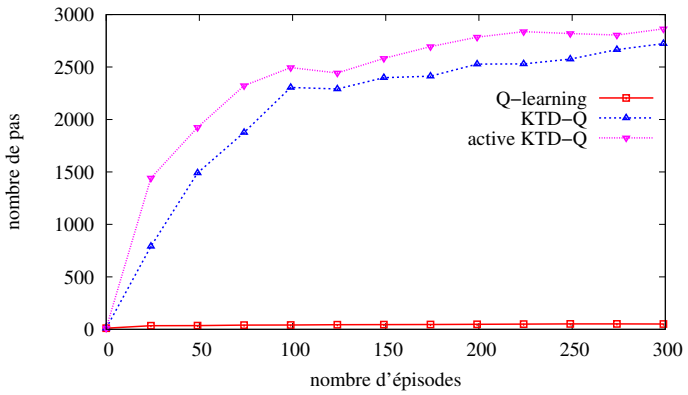


FIGURE 3.13 – Apprentissage actif.

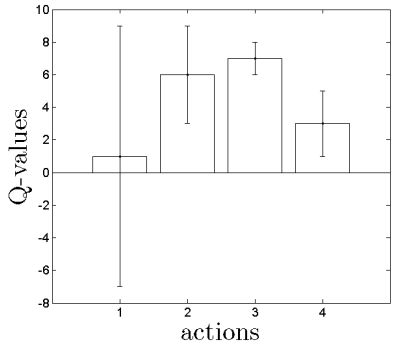


FIGURE 3.14 – Q-valeurs et incertitude associée.

cace, du moins sur cet exemple. Il est à noter que le schéma d'apprentissage actif n'a pas été considéré combiné avec l'algorithme Q-learning. La raison en est simple : ce dernier ne produit pas d'information d'incertitude.

3.4.4 Dilemme entre exploration et exploitation

Dans cette section sont proposées un certain nombre d'approches visant à traiter du dilemme entre exploration et exploitation. La première est le schéma classique ϵ-glouton, qui servira de référence. Les autres sont inspirées de la littérature (cas tabulaire) et utilisent l'information d'incertitude disponible (voir la section 3.4 pour son calcul). La combinaison de KTD-SARSA avec un schéma de contrôle est présentée algorithme 7.

Politique ϵ-gloutonne

Avec une politique ϵ-gloutonne, l'agent choisit une action optimale (respectivement à la Q-fonction estimée courante) avec une probabilité $1-\epsilon$, une action aléatoire (tirage uniforme) avec une probabilité ϵ (δ étant le symbole de Kronecker) :

$$\begin{aligned}\pi(a_{i+1}|s_{i+1}) = (1-\epsilon)\delta(a_{i+1} = \underset{b\in A}{\operatorname{argmax}}\, \bar{Q}_{i|i-1}(s_{i+1},b)) \\ + \epsilon\delta(a_{i+1} \neq \underset{b\in A}{\operatorname{argmax}}\, \bar{Q}_{i|i-1}(s_{i+1},b))\end{aligned} \tag{3.70}$$

Cette politique est peut-être la plus basique, bien qu'elle soit très utilisée, et elle n'utilise aucune information d'incertitude. Une Q-fonction arbitraire pour un état donné et quatre actions différentes est

Algorithme 7 : KTD-SARSA et contrôle

Initialisation : *a prioris* $\hat{\theta}_{0|0}$ et $P_{0|0}$, état s_1, action a_1;

for $i \leftarrow 1, 2, \ldots$ **do**

Appliquer a_i en l'état s_i ;
Observer r_i et s_{i+1} ;

Phase de prédiction;
$\hat{\theta}_{i|i-1} = \hat{\theta}_{i-1|i-1}$;
$P_{i|i-1} = P_{i-1|i-1} + P_{v_i}$;

Calcul des sigma-points et échantillonnage;
$\Theta_{i|i-1} = \{\hat{\theta}^{(j)}_{i|i-1}, \quad 0 \leq j \leq 2p\}$;
/* en utilisant $\hat{\theta}_{i|i-1}$ et $P_{i|i-1}$ */
$\mathcal{W} = \{w_j, \quad 0 \leq j \leq 2p \quad\}$;
for $a \in A$ **do**
$\mathcal{Q}_{i|i-1}(s_{i+1}, a) = \{\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a), 0 \leq j \leq 2p\}$;
$\bar{Q}_{i|i-1}(s_{i+1}, a) = \sum_{j=0}^{2p} w_j \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a)$;
$\hat{\sigma}^2_{Q_{i|i-1}}(s_{i+1}, a) = \sum_{j=0}^{2p} w_j (\hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a) - \bar{Q}_{i|i-1}(s_{i+1}, a))^2$;
Echantillonner a_{i+1} selon $\pi(.|s_{i+1})$, voir Eq. (3.70-3.73);
$\mathcal{R}_{i|i-1} = \{\hat{r}^{(j)}_{i|i-1} = \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_i, a_i)$
$- \gamma \hat{Q}_{\hat{\theta}^{(j)}_{i|i-1}}(s_{i+1}, a_{i+1}), 0 \leq j \leq 2p\}$;
Calcul des statistiques d'intérêt;
$\hat{r}_{i|i-1} = \sum_{j=0}^{2p} w_j \hat{r}^{(j)}_{i|i-1}$;
$P_{\theta r_i} = \sum_{j=0}^{2p} w_j (\hat{\theta}^{(j)}_{i|i-1} - \hat{\theta}_{i|i-1})(\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})$;
$P_{r_i} = \sum_{j=0}^{2p} w_j (\hat{r}^{(j)}_{i|i-1} - \hat{r}_{i|i-1})^2 + P_{n_i}$;
Phase de correction;
$K_i = P_{\theta r_i} P_{r_i}^{-1}$;
$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(r_i - \hat{r}_{i|i-1})$;
$P_{i|i} = P_{i|i-1} - K_i P_{r_i} K_i^T$;

illustrée figure 3.14. Pour chaque action, la figure donne la Q-valeur estimée ainsi que l'incertitude associée (c'est-à-dire ici $\pm$ l'écart-type associé). Par exemple, l'action 3 présente la plus grande valeur et la plus faible incertitude, tandis que l'action 1 a la plus faible valeur mais la plus grande incertitude. La distribution sur les actions associée à la politique ϵ-gloutonne est illustrée sur la figure 3.15.a. La plus grande probabilité est associée à l'action 3, et les autres actions ont la même (faible) probabilité, malgré le fait qu'elles présentent des Q-valeurs estimées et incertitudes associées sensiblement différentes.

Politique confiante-gloutonne

La deuxième approche considérée consiste à agir de façon gloutonne par rapport à la borne supérieure d'un intervalle de confiance estimé. L'approche n'est pas nouvelle [108], mais des garanties PAC (probablement approximativement correct) on récemment été données par [239] pour le cas tabulaire (pour lequel l'écart-type est proportionnel à l'inverse de la racine carrée du nombre de visites de la paire état-action considérée). Dans le cas traité ici, nous postulons que la largeur de l'intervalle de confiance est proportionnel à l'écart-type estimé (ce que est vrai si la distribution des paramètres est gaussienne). Soit α un paramètre libre positif, la politique confiante-gloutonne est définie comme suit :

$$\pi(a_{i+1}|s_{i+1}) = \delta\Big(a_{i+1} = \underset{b \in A}{\operatorname{argmax}} \left(\bar{Q}_{i|i-1}(s_{i+1}, b) + \alpha \hat{\sigma}_{Q_{i|i-1}}(s_{i+1}, b)\right)\Big) \tag{3.71}$$

La même Q-fonction arbitraire est considérée (voir figure 3.14) et la politique confiante-gloutonne est illustrée figure 3.15.b, qui représente la borne supérieure de l'intervalle de confiance par rapport à laquelle l'agent agit de façon gloutonne. L'action 1 est choisie car elle a le plus grand score, malgré le fait qu'elle présente la plus faible valeur estimée. Il est à noter que l'action 3, qui présente la plus grande Q-valeur estimée, n'est qu'en troisième position pour cette politique.

Politique bonus-gloutonne

La troisième approche proposée s'inspire de la méthode proposée par [115]. La politique qu'ils utilisent est gloutonne par rapport à la Q-valeur estimée plus un bonus, ce bonus étant inversement proportionnel à l'inverse du nombre de visites de la paire état-action d'intérêt. Cela peut s'interpréter comme une variance, de la même façon que l'écart-type pour la politique confiante-gloutonne est assimilé à la racine carrée de cette même quantité. La politique bonus-gloutonne que nous proposons utilise donc la variance estimée, et est définie par (β_0 et β étant deux paramètres libres) :

$$\pi(a_{i+1}|s_{i+1}) = \delta\Big(a_{i+1} = \underset{b\in A}{\operatorname{argmax}} \big(\bar{Q}_{i|i-1}(s_{i+1},b) + \beta \frac{\hat{\sigma}^2_{\bar{Q}_{i|i-1}}(s_{i+1},b)}{\beta_0 + \hat{\sigma}^2_{\bar{Q}_{i|i-1}}(s_{i+1},b)}\big)\Big) \tag{3.72}$$

Cette politique bonus-gloutonne est illustrée figure 3.15.c, toujours respectivement à la Q-fonction arbitraire de la figure 3.14. L'action 2 a le plus grand score, elle est donc choisie. A noter que les autres actions ont approximativement le même score, malgré le fait qu'elle présentent des Q-valeurs estimées sensiblement différentes.

Politique de Thompson

Rappelons que KTD maintient les moments d'ordre un et deux du vecteur de paramètres. En supposant que ce vecteur aléatoire suit une distribution gaussienne, nous proposons d'échantillonner un jeu de paramètres en accord avec la distribution estimée, puis d'agir de façon gloutonne respectivement à la Q-fonction résultante. Ce type d'approche à d'abord été proposé par [247] dans le cadre d'un problème de bandit, puis a été plus récemment introduit en apprentissage par renforcement dans le cas tabulaire [45, 240]. La politique de Thompson est définie ici comme suit :

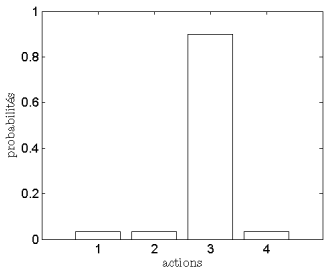
$$\pi(a_{i+1}|s_{i+1}) = \underset{b\in A}{\operatorname{argmax}}\, \hat{Q}_\xi(s_{i+1},b) \text{ avec } \xi \sim \mathcal{N}(\hat{\theta}_{i|i-1}, P_{i|i-1}) \tag{3.73}$$

Cette politique est illustrée figure 3.15.d, qui montre la distribution de l'action gloutonne (les paramètres étant des variables aléatoires, l'action gloutonne l'est également). La plus grand probabilité est associée à l'action 3. Cependant, il est à noter qu'une plus grande probabilité est associée à l'action 1 que à la 4 : la première action a une Q-valeur estimée plus faible, mais moins certaine.

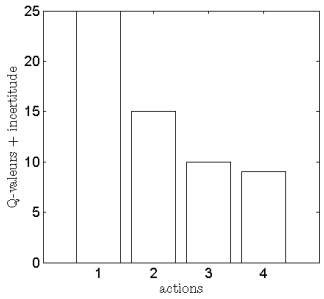
Expérimentation

Le problème du bandit à N bras est un PDM à un état et N actions. Chaque action q implique une récompense de 1 avec probabilité p_a et une récompense de 0 avec probabilité $1 - p_a$. Pour une action a^* (choisie aléatoirement au début de chaque expérimentation), cette probabilité est choisie égale à $p_{a^*} = 0,6$. Pour toutes les autres actions, la probabilité associée est choisie aléatoirement et uniformément entre 0 et $0,5$: $p_a \sim \mathcal{U}_{[0,0.5]}, \forall a \neq a^*$. Les résultat présentés sont moyennés sur 1000 expérimentations. La performance d'une approche est mesurée comme étant le pourcentage de fois où l'action optimale a été choisie, en fonction du nombre d'interactions entre l'agent et le bandit. Une représentation tabulaire est adoptée pour KTD-SARSA, et les paramètres suivants sont utilisés : $N = 10$, $P_{0|0} = 0.1I$, $\theta_{0|0} = I$, $P_{n_i} = 1$, $\epsilon = 0.1$, $\alpha = 0.3$, $\beta_0 = 1$ et $\beta = 10$. Comme le bandit considéré a $N = 10$ bras, une politique aléatoire a une performance de $0,1$. Notons qu'une politique purement gloutonne choisirait systématiquement la première action pour laquelle l'agent aurait observé une récompense positive.

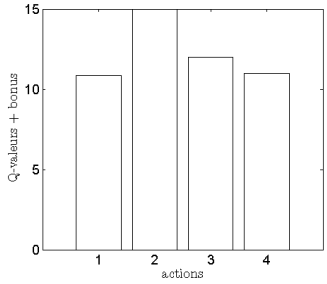
Les résultats de la figure 3.16 comparent les quatre schémas introduits. La politique ϵ-gloutonne sert de référence et tous les schémas proposés, qui utilisent l'information d'incertitude disponible dans le



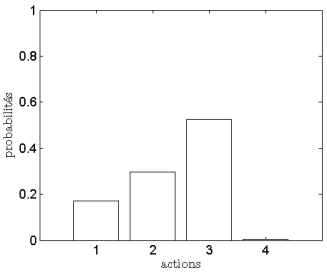
a. ϵ-gloutonne.



b. Confiante-gloutonne.



c. Bonus-gloutonne.



d. Thompson.

FIGURE 3.15 – Politiques.

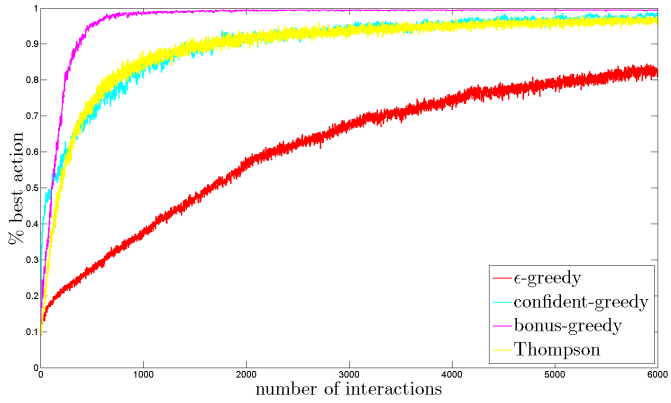


FIGURE 3.16 – Résultats du bandit.

cadre de KTD, montrent de meilleures performances. La politique de Thompson et la politique confiante-gloutonne présentent des résultats similaires, et les meilleurs résultats sont obtenus par la politique bonus-gloutonne. Bien sûr, ces résultats préliminaires ne permettent pas de conclure sur l'efficacité générale des approches proposées. Cependant, ils tendent à montrer que l'information d'incertitude fournie par KTD a du sens et peut s'avérer utile pour traiter du dilemme entre exploration et exploitation.

3.4.5 Discussion

Nous mettons ici KTD en perspective avec les approches les plus similaires. La méthode la plus proches est sans doute celle exposée dans [49] où l'auteur propose une approche à base de processus gaussiens (GPTD [13].). Il modélise la fonction de valeur comme un processus gaussien [208] et adopte un modèle génératif basé sur l'équation d'évaluation de Bellman. Avec une paramétrisation linéaire et un bruit d'évolution nul, KTD-V se réduit à GPTD et XKTD-V à MC-GPTD [14]. Néanmoins, KTD permet de prendre en compte les non-stationnarités (GPTD pourrait sans doute être étendu pour en faire autant mais ce n'est pour l'instant pas réalisé) et surtout permet de gérer les non-linéarités sans nécessiter le calcul de dérivées. Ceci permet d'utiliser des paramétrisations non-linéaires comme dans [86, 73] et de traiter l'équation d'optimalité de Bellman pour dériver des algorithmes *off policy* comme KTD-Q. Des paramétrisations non-linéaires devraient permettre de rendre plus compacte la représentation de la fonction de valeur et résulter en une compensation de la complexité de l'algorithme. Le cadre développé par Engel permet de construire en ligne une paramétrisation basée sur des noyaux ce qui est un avantage en comparaison à KTD qui doit disposer de la paramétrisation à l'avance. Néanmoins, cette méthode peut être intégrée à KTD (voir [86]).

Le filtrage de Kalman est fortement lié aux méthodes des moindres carrés (dans le cas linéaire, le premier est une généralisation du second). C'est pourquoi l'approche proposée présente des similarités avec LSTD [20]. Néanmoins, elle ne fait pas appel au concept de variables instrumentales [243], qui est utilisé pour traiter le cas des transitions stochastiques (pour éviter le problème de biais). On peut par ailleurs montrer que XKTD-V converge vers la même solution que LSTD(1) dans le cas d'une paramétrisation linéaire et un bruit d'évolution nul [75]. Dans [35], les auteurs introduisent un filtre de Kalman pour traiter la résolution par point fixe dans le cas d'une paramétrisation linéaire. Cette méthode peut être vue comme une version de KTD-V utilisant du *bootstrapping* plutôt qu'une approche résiduelle. Dans [168], un banc de filtres de Kalman est utilisé pour apprendre une paramétrisation linéaire par morceaux de la fonction de valeur. Cette hypothèse de linéarité par morceaux n'est pas faite dans l'approche KTD qui est donc plus générale.

L'approche proposée ici possède plusieurs avantages. Tout d'abord, il est possible de traiter des PDM non-stationnaires [97], comme nous l'avons montré sur la chaîne de Boyan. Mais une application plus intéressante de la gestion de la non-stationnarité est le problème du contrôle où le fait que la politique change constamment induit une non-stationnarité de la fonction de valeur évaluée. Par exemple, LSTD est connu pour se comporter mal quand il est combiné à un schéma d'itération optimiste de la politique (politique ϵ-gloutonne par exemple, voir [168]). Pour cette raison, dans [15] les auteurs préfèrent utiliser un simple TD plutôt que LSTD. Il existe assez peu de méthodes d'approximation de la fonction de valeur qui permettent de prendre en compte la non-stationnarité, la méthode décrite dans [168] étant l'une d'entre elles. D'autres approches existent comme celles exposées dans [116] ou dans [15] par exemple.

Ensuite, comme KTD représente le vecteur de paramètres comme un vecteur aléatoire, il est possible de calculer une information d'incertitude associée à l'estimation. Cela peut être utile dans le cadre du dilemme entre exploration et exploitation. La plupart des méthodes de gestion de l'incertitude ne traitent pas de l'approximation de la fonction de valeur (voir [45] ou [239]) et *vice versa*. L'approche de Engel [49] peut être utilisée en ce sens, bien que cela n'ait pas été fait dans la littérature.

KTD partage un inconvénient avec les autres méthodes de minimisation du résidu de Bellman : les quantités estimées sont biaisées si le PDM est stochastique. Il y a plusieurs méthodes pour s'affranchir de ce problème dans la littérature. Les algorithmes résiduels de [12] proposent un double échantillonnage qui nécessite de disposer d'un simulateur. Pour dériver LSTD [20], le concept de variables instrumentales [243] est utilisé. Néanmoins, cela ne fonctionne que parce que LSTD est basé sur une

13. *Gaussian Process Temporal Differences*
14. Monte Carlo GPTD

résolution au sens des moindres carrés linéaires ce qui implique de ne pas gérer les non-linéarité ni les non-stationnarités. Dans [8], les auteurs proposent d'introduire une fonction auxiliaire dont le rôle est d'annuler le biais. Cette méthode permet de gérer les non-linéarités et a été utilisée avec des réseaux de neurones [230]. Les algorithmes basés sur GPTD [49] utilisent un bruit colorés comme nous l'avons fait dans KTD.

Enfin, notons que KTD a aussi été utilisé dans un cadre d'architecture acteur-critique [78] ce qui permet de s'affranchir de calculer un argmax sur les actions pour améliorer la politique dans un cadre de contrôle.

Chapitre 4

Applications du contrôle optimal

4.1 Dialogue homme-machine

Comme indiqué dans l'introduction de ce document, c'est le dialogue homme-machine [173, 191] et donc l'interaction avec l'humain qui nous a mené à étudier l'apprentissage par renforcement et à essayer d'apporter des solutions aux problèmes du passage à l'échelle et de l'efficacité des algorithmes existants. C'est donc naturellement la première application à bénéficier de nos travaux sur ce sujet. Nous avons par ailleurs eu l'occasion de poursuivre nos travaux sur ce thème dans le cadre de différents projets comme le projet européen FP7 CLASSiC ou le projet FEDER ALLEGRO en collaboration avec des académiques (Universités de Cambridge, Edinburgh, Genève, Saarbrücken ou l'INRIA) mais aussi avec des industriels comme France Telecom. Nous consacrons donc quelques pages de ce manuscrit à cette thématique importante dans notre projet scientifique et qui présente toutes les contraintes de la présence de l'humain dans le processus de conception de systèmes intelligents.

Les systèmes de dialogue vocaux (SDS) [1] permettent l'interaction entre une personne et une machine par le biais de la parole. Ces systèmes ont généralement pour but l'accomplissement d'une tâche spécifique comme l'accès à des informations sur les horaires de bus [209], la réservation de billets d'avion [127], l'accès à une base de données [178], la communication homme-robot [218], l'interaction dans les environnements intelligents, les systèmes d'interaction main-libre pour l'automobile [122], l'aide aux personnes dépendantes [17], *etc*. Pour réaliser ce type de système, il ne suffit pas d'assembler des modules de reconnaissance et de synthèse de parole. Il faut non seulement leur adjoindre des modules de compréhension et de génération de langage naturel mais aussi (et surtout) un gestionnaire de l'interaction. Celui-ci est responsable de gérer l'échange d'information entre la machine et l'utilisateur dans le but d'accomplir la tâche. Il peut être considéré comme l'organe d'ordonnancement du système de dialogue. Il est en particulier responsable des stratégies de confirmation qui sont indispensables du fait du caractère imparfait des systèmes de reconnaissance et de compréhension de la parole. Une mauvaise gestion des sous-dialogues de confirmation (par exemple, produisant des demandes de confirmations intempestives) peut rendre l'interface très peu naturelle et dissuader les utilisateurs d'en faire usage. La conception d'un tel module est donc rendue assez complexe étant donné l'incertitude qui peut résider dans l'acquisition des informations par le système (bruit, erreur de reconnaissance ou de compréhension automatique de la parole) mais aussi le caractère aléatoire des réponses des utilisateurs (la même question peut engendrer différentes réponses d'un utilisateur à un autre ou même pour un utilisateur donné à différents instants) ou la variabilité des tâches qui peuvent être considérées. Ce travail reste pour l'instant souvent réservé à des experts du traitement automatique des langues (personnel hautement qualifié et donc rare et coûteux) et il est difficile d'obtenir des systèmes robustes et naturels c'est-à-dire possédant des facultés d'adaptation à la situation et à l'utilisateur. Il est de plus difficile de porter le travail fait dans le cadre d'un système vers un autre système pour une tâche différente. Cela freine considérablement la pénétration de ce type d'interface dans un marché grand public, bien qu'elles soient de plus en plus répandues aux États-Unis par exemple.

1. pour *Spoken Dialogue Systems*.

Depuis la fin des années 1990, des approches basées sur l'apprentissage statistique ont été étudiées dans le but d'automatiser en grande partie ce travail de conception pour les gestionnaires de dialogue [126, 237, 218, 191, 255] en vue d'apprendre artificiellement des *stratégies* d'interaction. Nous verrons en effet que la gestion de dialogue peut être mise sous la forme d'un problème de décisions séquentielles et peut donc être traité dans le cadre des PDM [126, 237, 191] ou de PDM partiellement observables (PDMPO [2]) [218, 255]. On peut donc théoriquement résoudre ce problème en appliquant des méthodes d'AR. Comme les méthodes standard requièrent des volumes considérables de données (ou de réelles interactions) pour converger, cela rend néanmoins l'application pratique de l'AR assez difficile. En effet, l'acquisition et l'annotation de données dans le domaine du dialogue homme-machine est assez coûteuse en temps et en argent tout comme la réalisation d'expériences de type *Magicien d'Oz* [111, 210]. Pour palier ce problème de rareté des données, les recherches se sont essentiellement dirigées vers la génération artificielle de données par le biais de simulation de dialogues [127, 177, 228] essentiellement en suivant la mouvance initiée par [126] qui indiquait que 710 000 dialogues avaient été nécessaires pour optimiser la stratégie pour une tâche assez simple. Néanmoins, cette méthode introduit de nouvelles sources d'erreurs de modélisation et les effets dus notamment à la simulation du comportement d'un utilisateur restent assez peu connus [227].

Une alternative à cette méthode de génération artificielle de dialogues serait d'utiliser des méthodes de généralisation efficaces en termes d'échantillons permettant d'apprendre des comportements pour des situations n'ayant pas été rencontrées dans les données. Des méthodes comme la programmation dynamique avec approximation de la fonction de valeur [101, 119] ou des méthodes d'approximation en ligne comme le SARSA(λ) de [242] ou évidemment un des algorithmes KTD développés dans les chapitres précédents. Il y a très peu d'exemples allant dans ce sens dans la littérature. Dans [104], les auteurs utilisent l'algorithme SARSA(λ) [242] avec une approximation linéaire de la fonction de valeur. Cet algorithme est toutefois connu pour être peu efficace. Dans [128], LSPI [3] [119] est utilisé en combinaison avec une méthode de sélection d'unités combinée à une approximation linéaire de la fonction de valeur. Ces deux méthodes sont des méthodes hors ligne qui utilisent des jeux de données fixes.

Nous avons étudié les deux alternatives durant ces dernières années et pensons qu'elles sont complémentaires mais appelées à évoluer vers un même schéma comme nous l'indiquerons dans notre projet pour le futur. Dans le reste de cette section, après avoir brièvement décrit les recherches que nous avons menées dans le cadre de la simulation de dialogues, nous allons expérimenter les algorithmes développés précédemment sur un problème de dialogue faisant partie de l'état de l'art. En effet, KTD possède plusieurs caractéristiques nécessaires pour ce type de tâche. Tout d'abord, il permet un apprentissage *off-policy* ce qui est nécessaire car il est difficile de contrôler la stratégie pendant l'apprentissage du fait de l'interaction avec l'humain [4]. Ensuite, un apprentissage efficace et en ligne, capable de réagir aux changements dans l'environnement. En effet, un tel changement peut intervenir lorsque le comportement des utilisateurs change [5]. Enfin, il est important que l'algorithme utilise un minimum d'échantillons pour converger du fait du coût associé à la collecte de données. Nous comparons cet algorithme à d'autres schémas d'approximation comme *Fitted Value Iteration* [101] ou *Least Square Policy Iteration* (LSPI) [119] que nous avons aussi récemment appliqués au problème du dialogue homme-machine [24, 23].

4.1.1 Simulation de dialogues

Nous profitons de cette partie consacrée au dialogue homme-machine pour faire brièvement état de la recherche qui a été poursuivie dans le domaine de la simulation de dialogues. Bien que nous ayons plutôt orienté nos recherches sur la mise au point d'algorithmes permettant de s'affranchir au maximum de cette simulation ou de démontrer qu'il faut faire évoluer cette simulation, nous pensons qu'il ne sera pas possible de l'éliminer totalement dans un avenir immédiat et avons donc cherché à améliorer les méthodes de simulation mises au point précédemment. Nous référons à nos publications dans le texte pour plus de détails.

Ainsi, nous nous sommes d'abord attachés à décrire plus formellement le processus de dialogue dans

2. Processus Décisionnel de Markov Partiellement Observable.
3. *Least Square Policy Iteration*
4. Il n'est pas pensable d'appliquer une politique arbitrairement mauvaise sous peine de voir l'utilisateur arrêter l'interaction.
5. C'est souvent le cas parce que les utilisateurs s'habituent à l'utilisation d'un système.

un cadre probabiliste [175, 191]. Cette formalisation a servi de base pour la mise au point d'un simulateur d'utilisateur consistant tant du point de vue de l'interaction que de la poursuite d'un but [177, 191]. Ce modèle d'utilisateur est en fait un réseau bayésien utilisé en mode génératif. Nous avons entraîné ce réseau bayésien sur des données obtenues lors de véritables dialogues homme-machine dans le cas d'une application à la recherche d'informations touristiques [201] comme celle qui nous servira plus tard pour démontrer l'efficacité de KTD. Ce simulateur d'utilisateur à ensuite été amélioré pour permettre la simulation de comportements plus complexes comme celui du *grounding* [36], comme décrit dans [214, 215] ; ceci afin de prendre en compte cet effet lors de l'apprentissage de stratégie optimale [179]. Nous avons aussi étendu l'apprentissage des paramètres du réseau bayésien au cas où des données seraient manquantes dans la base d'exemples [110, 216]. C'est en effet le cas la plupart du temps puisque la représentation interne que se fait l'utilisateur sur l'évolution du dialogue ne peut pas être annotée par exemple. Cette représentation est pourtant nécessaire pour reproduire un comportement cohérent tout au long de l'interaction.

Pour simuler l'entièreté du processus de dialogue, il ne suffit pas de simuler l'utilisateur mais aussi le processus de traitement des entrées vocales par les différents systèmes d'acquisition (reconnaissance vocale, compréhension du langage). Ainsi, nous avions entrepris durant la thèse de construire une modélisation des erreurs introduites par les systèmes de reconnaissance de la parole [200, 188, 186]. La recherche dans ce domaine a été poursuivie pour introduire un modèle d'erreur suite au traitement automatique du langage en vue de sa compréhension (extraction des concepts et non plus des mots). C'est un modèle similaire à celui développé pour la simulation d'utilisateur et basé sur des réseaux bayésiens qui a été développé [176, 190]. Cette fois le réseau est utilisé pour réaliser de l'inférence bayésienne et plus en mode génératif.

Nous indiquerons dans le projet de recherche (Partie III) les directions futures que nous souhaitons développer dans le cadre de la simulation de dialogue et comment unifier les développements réalisés dans le cadre de l'AR à ces approches à venir.

4.1.2 Dialogue et PDM

Un dialogue (particulièrement dans le cadre d'une tâche collaborative) peut être vu à haut-niveau comme un processus de décisions séquentielles. Dans ce processus, le gestionnaire de dialogue doit sélectionner les informations qu'il doit fournir ou demander à l'utilisateur dans chaque situation donnée dans le but d'accomplir la tâche de manière naturelle. Ceci peut être traduit dans le cadre des PDM. Dans la communauté scientifique qui étudie le discours et le dialogue (homme-homme ou homme-machine) il est communément admis que les intervenants produisent des *actes de communications* ou *actes communicatifs* qui sont appelés comme tels parce qu'ils influencent le comportement de l'autre intervenant et donc *agissent* sur l'évolution du dialogue et, en fait, sur le monde extérieur. Plusieurs types d'actes de communication sont possibles comme la salutation, les questions, les affirmations, les demandes de confirmation, les demandes de relaxation de contraintes, la clôture du dialogue, *etc*. Ces actes de communications sont le pendant des actions pour un PDM. L'état d'un dialogue à un instant donné est souvent décrit par les informations qui ont été échangées jusqu'à l'instant considéré. Il est possible, dans cette optique, de représenter l'état de manière efficace en utilisant le paradigme de *l'état informatif* [121]. Dans ce paradigme, l'état du dialogue contient une représentation compacte de l'historique du dialogue en termes d'actes communicatifs (provenant du système et de l'utilisateur). Une stratégie de gestion du dialogue π cherche alors à établir une correspondance entre des états de dialogue et des actes communicatifs. Pour rester dans le cadre d'un PDM, il faut aussi définir les récompenses immédiates permettant l'optimisation. Cette récompense est souvent modélisée comme la contribution de chacun des actes communicatifs à la satisfaction de l'utilisateur [237]. Cette notion relativement subjective est généralement estimée par une combinaison linéaire de mesures objectives qu'il est possible de réaliser automatiquement lors de l'apprentissage. Ces mesures peuvent être la durée du dialogue, le nombre d'erreurs de reconnaissance vocale, la complétion de la tâche, *etc*. Les poids de cette combinaison linéaire peuvent être estimés à partir de données empiriques comme dans [135]. Néanmoins, il est fréquent que des poids heuristiques soient utilisés étant donné la difficulté de l'obtention de données.

4.1.3 Description du système

Afin de démontrer l'intérêt des méthodes d'apprentissage développées plus tôt dans le cadre de l'optimisation de stratégies d'interaction, elles sont évaluées sur une tâche de dialogue homme-machine particulière qui a été au centre du projet FP7 CLASSiC. Le système de dialogue considéré vise l'obtention d'informations touristiques, il est proche de celui proposé par [123]. Son objectif est de fournir à l'utilisateur des informations concernant un restaurant, plus précisément sa localisation dans la ville (imaginaire), le type de cuisine et la gamme de prix. Du point de vue de l'agent, l'objectif est donc de remplir les trois *slots* correspondants (localisation, cuisine et prix) afin de proposer le bon restaurant à l'utilisateur. Les actions possibles de dialogue sont :

- demander explicitement la valeur d'un slot (par exemple "Quelle type de cuisine recherchez-vous ?"), ce qui totalise 3 actions ;
- confirmer explicitement la valeur d'un slot (par exemple "Pouvez vous confirmer que vous cherchez un restaurant dans le centre ?"), ce qui totalise 3 actions ;
- confirmer implicitement un slot en demandant la valeur d'un autre (par exemple "Vous recherchez un restaurant indien, dans quelle zone de la ville ?"), ce qui totalise 6 actions ;
- clore le dialogue en proposant un restaurant ("Vous rechercher un restaurant français bon marché dans le centre, nous vous proposons...").

Il y a donc au total 13 actions possibles.

Il est nécessaire de modéliser le système de dialogue comme un PDM pour y appliquer un algorithme d'apprentissage par renforcement. Les actions sont celles présentées précédemment. Concernant l'état, il n'est pas réaliste de travailler directement avec la sortie du module de reconnaissance de parole. Nous utilisons le paradigme de l'état d'information (*State Information paradigm*) [121], qui à partir de l'historique des sorties du module de reconnaissance de parole fournit deux valeurs par slot : une probabilité de remplissage (*filling confidence*) et une probabilité de confirmation (*confirmation confidence*). Nous considérons un espace d'état à 3 dimensions, chaque composante étant la moyenne de ces deux probabilités pour un slot donné. Il est également nécessaire de définir une fonction de récompense. Cette dernière est nulle tout le temps, sauf lorsque l'action de clore le dialogue est choisie. Dans ce cas, l'agent reçoit une récompense de $+25$ par slot correctement rempli, de -75 par slot incorrectement rempli et de -300 par slot vide. Le facteur d'actualisation est choisi égal à $\gamma = 0,95$. Etant donné que nous expérimentons des algorithmes ne nécessitant pas de connaître les probabilités de transitions, ces dernières n'ont pas à être estimées.

Les expériences présentées ici sont conduites en utilisant un simulateur d'utilisateur afin de pouvoir générer autant de dialogues que nous le souhaitons. Ceci permettra de comparer les vitesses de convergence des différents algorithmes comparés. Pour ce faire, le simulateur d'utilisateur est combiné au gestionnaire de dialogue DIPPER [123].

4.1.4 Algorithmes considérés et Q-fonction paramétrée

Dans un premier temps nous cherchons à démontrer l'efficacité de l'algorithme *off-policy* dérivé du cadre KTD sur cette application, soit KTD-Q. Les algorithms Fitted-Q [101], LSPI [119] et Q-Learning [242] avec approximation de fonction sont comparés à KTD-Q. Pour ce faire, une base d'échantillons est produite à partir d'une politique assez rustre codée manuellement. Cette politique cherche simplement à remplir les 3 *slots* l'un après l'autre en posant des questions directes à leur sujet et sans chercher à les confirmer. Des légères variantes à cette politique sont générées en tirant aléatoirement des actions dans un cas sur dix.

Ensuite l'intérêt des schémas d'exploration décrits dans la Section 3.4.4 est étudié. Nous considérons alors trois algorithmes : LSPI [119], KTD-SARSA combiné à un schéma d'exploration ϵ-glouton et KTD-SARSA combiné à un schéma d'exploration bonus-glouton. L'algorithme LSPI est hors-ligne (du moins tel que nous le considérons ici), mais il est reconnu comme étant très efficace et il a déjà fait ses preuves dans le domaine du dialogue homme-machine [128, 24, 23]. Il nous servira de référence en terme de performances. Deux schémas d'explorations combinés à KTD-SARSA sont considérés : la politique ϵ-gloutonne sert de référence, et nous avons choisi la politique bonus-gloutonne car elle présente les meilleurs résultats expérimentaux sur le problème de bandit.

L'espace d'état étant continu (un cube de côté 1), il est nécessaire de choisir une architecture paramétrée pour la Q-fonction. Ici, la Q-fonction est représentée par un réseau RBF [166] par action. Il y a trois noyaux gaussiens par dimension, d'écart-type $0,25$, et ce pour chaque action. Il y a donc au total 351 (c'est-à-dire $3^3 \times 13$) fonctions de base. Tous les algorithmes considérés utilisent la même paramétrisation pour la Q-fonction.

4.1.5 Résultats

Les résultats obtenus dans le cas de KTD-Q sont montrés sur la figure 4.1 (échelle linéaire) et 4.2 (échelle semi-logarithmique). Sur ces figures sont représentées la moyenne et la variance du cumul pondéré de récompenses obtenu par les politiques apprises par les différents algorithmes plus la politique codée manuellement qui a servi à générer les échantillons qui ont servi à l'apprentissage, comme expliqué dans la section précédente. Les performances de cette politique sont indiquées pour montrer qu'il n'est pas nécessaire d'utiliser une politique évoluée pour recueillir des échantillons. Elles ne sont pas données pour servir de base de comparaison avec les performances des autres algorithmes. L'abscisse représente le nombre de transitions (ou les tours de dialogue) et non pas les dialogues. En moyenne, un dialogue généré en appliquant la politique optimale dure entre 4 et 8 tours. Pour obtenir ces statistiques, chaque algorithme est entraîné 8 fois sur différentes jeux de données contenant 5.10^4 transitions (pour chaque phase d'entraînement, chaque algorithme est entraîné sur les mêmes échantillons), ensuite chaque politique est testée 50 fois. Chaque point de la courbe est donc obtenu sur 400 essais. Notons que les dialogues de test sont stoppés s'ils n'ont pas aboutis après 100 tours et la récompense associée est donc proche de 0. Ceci veut dire qu'une récompense de 0 indique que la politique ne peut terminer le dialogue dans un délai raisonnable et c'est pourquoi la variance associée est faible (le résultat est presque toujours nul). Les courbes pour Fitted-Q et LSPI n'existent pas pour des nombres d'échantillons inférieurs à 5.10^3, parce que ces algorithmes nécessitent l'inversion de matrices qui sont mal conditionnées en deçà de ce chiffre. Plus simplement, ces algorithmes ne convergent pas avec un nombre trop faible d'échantillons.

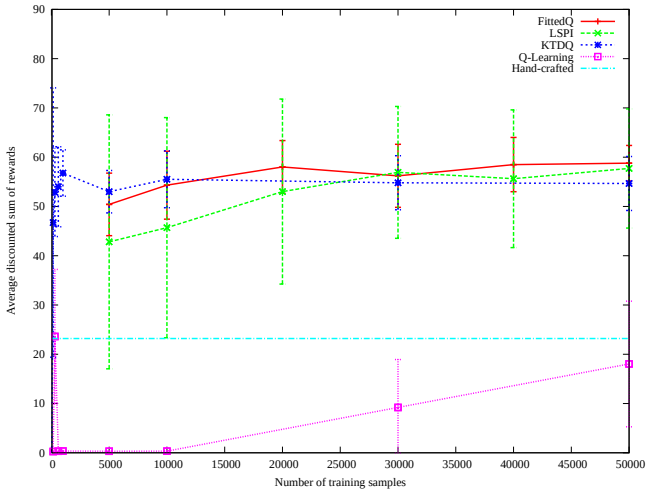


FIGURE 4.1 – Résultats en échelle linéaire

D'après les résultats de la figure 4.1, Q-learning apprend très lentement. Il est même incapable d'apprendre une politique au moins aussi bonne que la politique codée manuellement après 5.10^4 tours. Ce résultat est comparable à ce qui est rapporté dans les premiers travaux sur ce domaines [126]. KTD-Q apprend une politique quasi-optimale assez rapidement. Ses performances sont un peu en dessous de

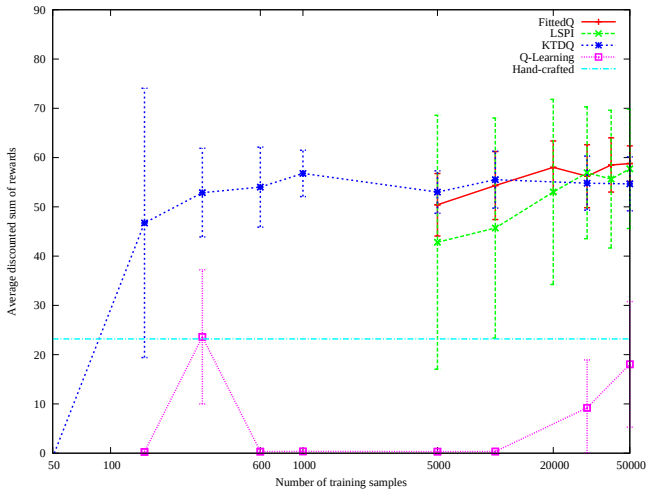


FIGURE 4.2 – Résultats en échelle semi-logarithmique

celles de LSPI ou Fitted-Q. Néanmoins, il est important de se rappeler que KTD-Q est un algorithme en-ligne alors que LSPI ou Fitted-Q fonctionnent hors-ligne. Par conséquent, KTD n'utilise qu'une seule fois chaque échantillon contrairement à ses concurrents qui traite au moins plusieurs dizaines de fois chaque échantillon pour converger.

Pour analyser plus finement les performances de KTD-Q la figure 4.2 fournit la même courbe en échelle semi-logarithmique. Sur cette figure, on peut voir qu'après 300 échantillons, Q-learning apprend une politique raisonnablement bonne (approximativement équivalente à la politique codée manuellement). Pourtant ses performances redeviennent mauvaises assez rapidement, montrant que la politique apprise par Q-learning n'est pas stable à ce stade. En revanche, KTD-Q apprend une politique assez bonne rapidement et ses performances moyennes restent stables alors que la variance associée décroît. Il est aussi intéressant de voir que KTD-Q fournit une politique assez bonne alors que les Fitted-Q et LSPI ne disposent pas encore d'assez d'échantillons pour converger. Ainsi, il serait possible d'entraîner KTD-Q sur un nombre de dialogues qu'il est possible de collecter réellement. La simulation d'utilisateur serait donc éventuellement dispensable pour ce type de tâche.

Notons que, pratiquement, l'amélioration des performances est obtenue en utilisant à bon escient les différentes stratégies de confirmation disponibles. L'algorithme apprend en effet le seuil de confiance en-dessous duquel il est nécessaire de demander une confirmation. Aussi, l'utilisation efficace de confirmations implicites réduit encore le nombre de tours nécessaires dans un dialogue ce qui permet d'augmenter encore la récompense moyenne obtenue par dialogue.

Ensuite, nous présentons l'utilisation de l'incertitude pour explorer l'espace d'état dans ce problème de dialogue. L'algorithme LSPI sert encore de référence, et présente une performance asymptotique moyenne d'environ 58 (cumul moyen pondéré des récompenses). En pratique, la politique correspondante permet de mener à terme le dialogue en seulement quelques interactions, avec satisfaction de l'utilisateur et avec une bonne robustesse au bruit lié au module de reconnaissance de la parole. KTD-SARSA présente également de bons résultats, quel que soit le schéma d'exploration considéré, et l'algorithme apprend à interagir de façon satisfaisante avec l'utilisateur en seulement quelques centaines d'épisodes. Si l'on compare les deux schémas d'exploration, on observe que le schéma utilisant l'information d'incertitude (la politique bonus-gloutonne) produit plus rapidement de meilleures politiques qui s'avèrent également être plus stables que la politique ϵ-gloutonne. Il est également à noter que KTD-SARSA combiné avec la politique bonus-gloutonne produit des politiques quasiment aussi bonnes que LSPI (performance d'environ 54 au lieu de 58). Avec une politique ϵ-gloutonne la performance oscille entre 40 et

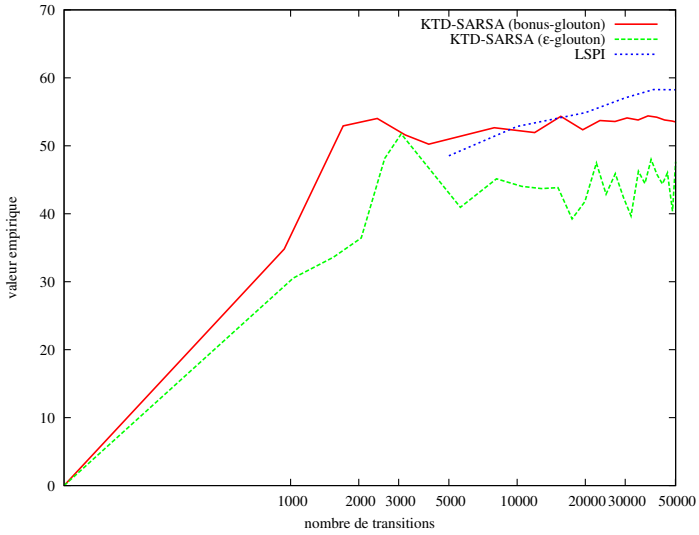


FIGURE 4.3 – Résultats sur le problème de dialogue.

47. De plus, KTD-SARSA produit déjà des politiques de qualité après un millier de transitions (ce qui correspond à seulement quelques dizaines d'épisodes, les premiers étant très longs) alors qu'il n'y a pas encore suffisamment d'échantillons pour pouvoir utiliser l'algorithme LSPI. Cela montre, que l'information d'incertitude fournie par KTD est utile, et que ce cadre de travail général permet de combiner efficacement approximation de la fonction de valeur et dilemme entre exploration et exploitation.

Par ailleurs, ces schémas d'exploration permettent d'interagir efficacement en ligne avec un utilisateur. Ceci indique que l'on pourrait améliorer les politiques en ligne tout en testant efficacement de nouvelles actions sur des utilisateurs réels. Il est important de ne pas réaliser des actions totalement aléatoires pour ce faire puisque les utilisateurs pourraient être gênés par des actions inconsistantes.

4.1.6 Conclusion

Le dialogue homme-machine est l'application qui nous a amené à tenter d'apporter des contributions à l'apprentissage par renforcement. Dans cette section nous avons donc naturellement appliqué les résultats de nos recherches sur le domaine de l'apprentissage par renforcement sur cette application. Les résultats expérimentaux montrent l'apport de ces développements pour le dialogue. En effet, le caractère *off-policy* de KTD-Q permet d'apprendre une politique optimale à partir d'échantillons recueillis avec une politique très simple et sûre. Son efficacité en termes d'échantillon permet d'apprendre des politiques optimales avec un nombre d'échantillons qu'il est possible de collecter et d'annoter. Ainsi, la simulation d'utilisateur peut éventuellement être évitée. Enfin, les schémas d'exploration utilisant l'information d'incertitude fournie par KTD et utilisable dans des algorithmes de type KTD-SARSA ont aussi prouvé leur intérêt.

Notons tout de même que le dialogue homme-machine est une application particulière puisqu'elle considère que l'utilisateur humain fait partie du système stochastique à contrôler. Celui-ci implique des comportements complexes et parfois non stationnaires. Des propriétés particulières sont donc souhaitables concernant les algorithmes de renforcement. Il est indispensable que les algorithmes soient efficaces en termes d'échantillons afin de ne pas avoir besoin d'interagir longuement avec l'humain pour apprendre. Il doit être possible d'apprendre *off-policy* afin d'utiliser une politique comportementale acceptable pour l'utilisateur. L'apprentissage en ligne doit aussi être possible pour améliorer le système au fur et à mesure de son utilisation ou pour personnaliser les stratégies apprises. Enfin, les changements dans les comportements d'un utilisateur ou d'une population d'utilisateurs doivent pouvoir être pris en compte, c'est pourquoi il est aussi nécessaire d'être en mesure de traiter les non-stationnarités.

4.2 Gestion de flux de gaz dans un complexe sidérurgique

Dans le cadre de la thèse [71] réalisée en partenariat avec ArcelorMittal par le biais du dispositif CIFRE [6], plusieurs applications ont été investiguées. Par exemple, des travaux ont été réalisés sur l'optimisation du remplacement de cylindres dans une chaîne de laminage d'acier que nous ne présenterons pas ici. En revanche, nous présentons une application à la gestion des flux de gaz dans un complexe sidérurgique. Le problème consiste à optimiser le flux de gaz entre des installations productrices (aciérie, cokerie...) et des installations consommatrices (four de recuit, usine de production d'électricité...) de gaz dans un complexe sidérurgique. Le routage devrait être fait de telle façon que chaque consommateur reçoive la quantité de gaz requise au bon moment. Si la production est trop basse, il peut être nécessaire d'en acheter à un réseau extérieur. Si elle est trop importante, il peut être nécessaire de torcher (brûler) le gaz (à certains endroits, cet excès peut également servir à produire de l'électricité). Le contrôle du réseau implique le routage de flux de gaz provenant d'une ou plusieurs sources vers une ou plusieurs destinations par le biais de nœuds diviseurs dans le réseau ou de switches (interrupteurs). Il y a également des accumulateurs de gaz (*tankers*) de capacité limitée qui peuvent être remplis ou vidés lorsque c'est nécessaire (stockage du gaz excédentaire en prévision d'une pénurie, par exemple). Il existe certaines contraintes du réseau qui doivent être vérifiées. Par exemple, certains tuyaux ont des contraintes (de flux) inférieure (flux minimal) et supérieure (flux maximal) et certains consommateurs ont des contraintes de capacité calorifique minimale (voir maximale) et de puissance pour le gaz les alimentant. Cela est important étant donné que les gaz provenant de différentes sources n'ont pas forcément les mêmes capacités calorifiques.

Les caractéristiques importantes pour traiter ce problème de gestion des flux de gaz peuvent être résumées comme suit :

contraintes du problème d'optimisation :
- infrastructure du réseau (position des mélangeurs et des switchs, tuyaux entre les différentes installations) et contraintes associées (limites de flux, capacités calorifiques minimales...),
- quantité de gaz produite par chaque installation et capacité calorifique associée,
- gaz consommé par chaque installation (puissance requise et capacité calorifiques minimale associée) ;

degrés de liberté pour l'optimisation :
- configuration du réseau (état des diviseurs, utilisation des accumulateurs, *et caetera*, mais on s'interdit d'en modifier la topologie) ;

fonction de coût :
- prix de la quantité de gaz achetée minorée du prix de la quantité de gaz revendue après transformation en électricité ;

autres informations :
- différentes mesures dans le réseau (pressions, flux, capacités calorifiques...).

Ce problème d'optimisation de la gestion des flux de gaz est traité au sein de l'entreprise par un logiciel commercial nommé `Aspen+`. Le moteur d'optimisation associé à ce produit permet de chercher une solution à la minimisation de la fonction de coût sous les contraintes du réseau. La solution obtenue a été comparée au contrôle des flux effectué par les opérateurs et des améliorations significatives peuvent être obtenues par rapport aux heuristiques employées en pratique. Cependant, `Aspen+` n'est pas tout à fait adapté à ce problème. Les résultats obtenus sont parfois plus mauvais que la solution de l'opérateur (minimum local et corruption des données peuvent être des explications). Ce logiciel ne permet pas de prendre en compte les switchs du réseau, seulement les mélangeurs et diviseurs continus. Mais surtout, l'optimisation dans le cadre d'`Aspen+` est statique. D'une part, cela ne permet pas de prendre en compte les accumulateurs de gaz. D'autre part, les opérateurs effectuent un contrôle dynamique des flux, la comparaison n'est pas totalement équitable. L'AR, quant à lui, peut prendre naturellement en compte cette composante dynamique du problème.

6. Conventions Industrielles de Formation par la REcherche

4.2.1 Modélisation conforme à l'apprentissage par renforcement

Il faut donc modéliser ce problème en termes adéquats au formalisme des processus décisionnels de Markov, c'est-à-dire spécifier ce qui doit être état, action et récompense.

Etats : l'état du système est composé :
- des contraintes de puissance (P) et de capacité calorifique minimale (c_{min}) des consommateurs contraints,
- du flux produit par les producteurs (f_0) et de la capacité calorifique associée (c_0),
- du flux (f^a) et de la capacité calorifique (c^a) courants des accumulateurs.

L'ensemble des contributions des nœuds au vecteur d'état est résumé par le tableau 4.1. Si on note N_C le nombre de consommateurs contraints, N_P le nombre de producteurs, et N_A le nombre d'accumulateurs, alors on a $(2N_c + 2N_P + 2N_A)$ composantes d'état (chaque composante étant continue) ;

Actions : l'ensemble des actions du système est composé :
- des degrés de liberté (u_i) au niveau des diviseurs ; il s'agit des proportions de flux dirigées en sortie vers les i tuyaux,
- des degrés de liberté (u^a) au niveau des accumulateurs ; il s'agit de la proportion de flux de gaz (entrée + contenu dans l'accumulateur) que l'accumulateur délivre en sortie.

L'ensemble des contributions des nœuds au vecteur d'action est résumé dans le tableau 4.1. Si l'on note N_S le nombre total de sorties (en cumulant tous les diviseurs), on a $(N_S + N_A)$ composantes continues ;

Récompenses : la récompense est égale au prix de l'électricité vendue (pour la partie du gaz transformé en électricité) minoré du prix de la quantité de gaz acheté. De plus, il faut pénaliser l'achat et le torchage de gaz là où ce n'est pas possible physiquement. On a donc une fonction de récompense de la forme :

$$r(s,a) = \sum_{i=1}^{N} \left(\alpha_T^{(i)} f_T^{(i)} + \alpha_G^{(i)} G_n^{(i)} \right) \tag{4.1}$$

Dans cette expression, N est le nombre total de nœuds et les différents termes ont les significations suivantes (ici $f_T^{(i)}$ est la quantité de gaz torché au nœud i et $G_n^{(i)}$ est le flux de gaz acheté pour ce même nœud) :
- si le nœud i dispose d'une torche, alors $\alpha_T^{(i)} = 0$. S'il ne dispose pas de torche et ne permet pas de produire de l'électricité, alors c'est un terme pénalisant (violation de la contrainte topologique) et $\alpha_T^{(i)} < 0$. Enfin, si le nœud produit de l'électricité, c'est le prix de revient par Watt produit (par exemple) et $\alpha_T^{(i)} > 0$. Notons que comme c'est la puissance qui est vendue et non le flux, ce coefficient devrait dépendre de la capacité calorifique. Nous choisissons donc dans ce cas $\alpha_T^{(i)} = \frac{c}{c_n}\alpha_{\text{PP}}$, où α_{PP} est une constante et c la capacité calorifique du gaz vendu ;
- si le nœud dispose d'un accès sur le réseau extérieur, alors $\alpha_G^{(i)}$ est le négatif du prix d'achat (par unité de flux) du gaz au réseau extérieur. Si le nœud ne dispose pas d'accès au réseau extérieur, il y a violation de la topologie du réseau, et $\alpha_G^{(i)}$ est une pénalisation (très) supérieure au prix d'achat du gaz.

Notons que pour obtenir une solution qui satisfasse les contraintes topologiques du réseau, il faut bien dimensionner ces différentes grandeurs.

Probabilités de transition : KTD définit une classe d'algorithmes *model-free*, ils ne nécessitent pas un modèle de transition. Cependant, il est important de noter que s'il y a indépendance de la transition à l'action, c'est-à-dire

$$p(s'|s,a) = p(s'|s)$$

alors maximiser le cumul de récompenses sur le long terme revient rigoureusement à maximiser la récompense immédiate. Ce n'est plus un problème d'AR mais d'apprentissage supervisé. Si on applique les algorithmes développés, cela va revenir à régresser la fonction de récompense et à en chercher le maximum. On arriverait au même résultat avec du Monte Carlo par exemple (et

de façon plus simple, étant donné qu'on dispose d'un modèle de simulation). Cela concerne uniquement le cas statique, sans accumulateur. Dans le cas dynamique, avec accumulateur, au moins les composantes d'état correspondant au contenu des accumulateurs dépendent de l'action choisie. L'évolution des grandeurs d'état relatives à l'accumulateur sont calculées simplement en fonction des flux d'entrée et de sortie. Pour les autres composantes d'état du système, nous avons choisi des modèles d'évolution temporelle en accord avec ArcelorMittal, de façon à obtenir quelque chose de proche du vrai système. Il s'agit essentiellement de modèles auto-régressifs.

Il faut noter que si cette modélisation est plus simple en termes de propagation des contraintes à travers le réseau, elle est plus complexe en termes d'espace état-action. En effet, certaines actions qui seraient impossibles avec une modélisation plus fine (mais bien plus complexe à mettre en œuvre) sont simplement pénalisées ici.

nœud	composantes d'état	composantes d'action
mélangeur	x	x
consommateur libre	x	x
consommateur contraint	(P, c_{min})	x
diviseur	x	$(u_1, \ldots, u_n)$
producteur	(f_0, c_0)	x
accumulateur	(f^a, c^a)	u^a

TABLE 4.1 – Contribution des nœuds en terme d'AR.

4.2.2 Un réseau particulier

Le réseau que nous proposons d'étudier peut sembler relativement simple. Cependant, il est intéressant pour plusieurs raisons. Tout d'abord, il représente d'un méta-point de vue assez fidèlement le type de réseaux que l'on peut rencontrer dans la pratique (les nœuds de production ou de consommation correspondant en fait à des sous-réseaux). C'est d'ailleurs également ce type de réseau qui est considéré par ArcelorMittal Research pour étudier les bénéfices des approches d'optimisation. D'autre part, comme nous le verrons, il permet de mettre en avant la gestion de l'accumulateur de gaz, qui est le point qui pose problème pour les approches utilisées en pratique.

Description du réseau

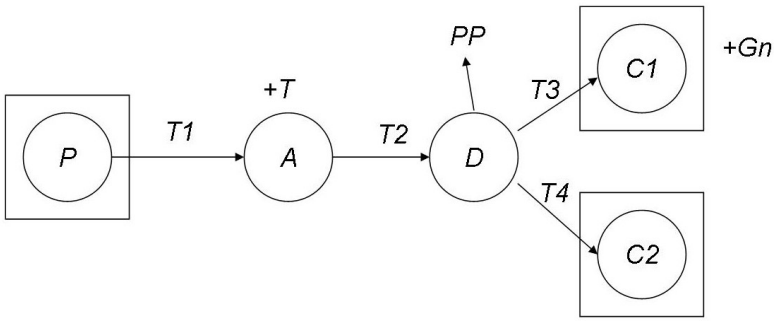


FIGURE 4.4 – Réseau considéré.

Nous considérons le réseau illustré sur la figure 4.4, qui est composé :
- d'un producteur P ;
- d'un accumulateur A qui est relié à une torche ;
- d'un diviseur D qui alimente deux consommateurs et dont le gaz torché alimente en fait une usine d'électricité ;

– d'un consommateur C_1 qui est connecté au réseau de gaz naturel ;
– d'un consommateur C_2 qui n'est pas connecté au réseau de gaz naturel.

D'après le parallèle avec l'apprentissage de la section 4.2.1, le producteur correspond à 2 composantes d'état, chaque consommateur également, le diviseur à deux composantes d'action et l'accumulateur à deux composantes d'état et une d'action. Finalement, on obtient un vecteur d'état de dimension 8 et un vecteur d'action de dimension 3. Pour résumer, nous avons comme vecteurs d'état et d'action :

$$\begin{aligned} s &= [\underbrace{s_1, s_2}_{P}, \underbrace{s_3, s_4}_{A}, \underbrace{s_5, s_6}_{C_1}, \underbrace{s_7, s_8}_{C_2}]^T \in \mathbb{R}^8 \\ &= [\underbrace{f_0, c_0}_{P}, \underbrace{f^a, c^a}_{A}, \underbrace{P_1, c_{1,\text{min}}}_{C_1}, \underbrace{P_2, c_{2,\text{min}}}_{C_2}]^T \end{aligned} \tag{4.2}$$

$$\begin{aligned} a &= [\underbrace{a_1}_{A}, \underbrace{a_2, a_3}_{D}]^T \in \mathbb{R}^3 \\ &= [\underbrace{u_a}_{A}, \underbrace{u_1, u_2}_{D}]^T \end{aligned} \tag{4.3}$$

4.2.3 Résultats

Ici nous nous proposons de comparer les politiques obtenues par KTD et Monte Carlo sur le simulateur considéré, la seconde approche étant, à l'utilisation d'heuristiques pour prendre en compte les accumulateurs près, le type d'approche qui est utilisé en pratique (c'est-à-dire maximisation de la récompense immédiate, plutôt que maximisation du cumul de récompenses sur le long terme). Nous commençons par présenter les résultats de Monte Carlo sur la figure 4.5. La politique est déterminée en choisissant, parmi un grand nombre d'actions tirées aléatoirement, celle qui maximise la récompense immédiate.

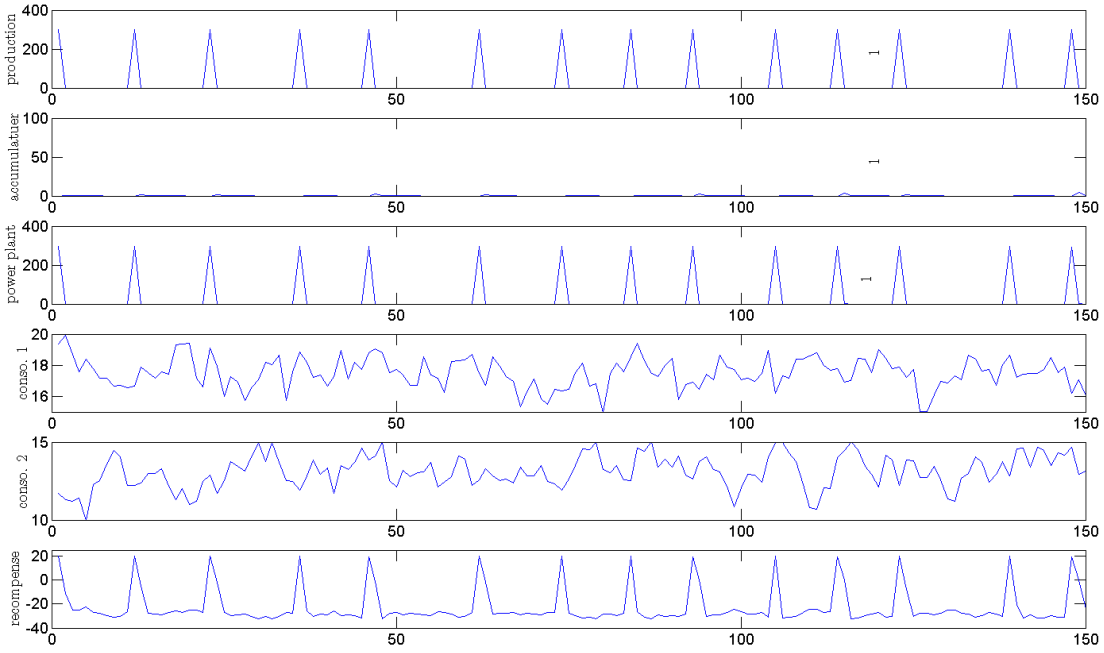


FIGURE 4.5 – Politique obtenue avec Monte Carlo.

Ce résultat, composé de six graphiques, représente une évolution typique du système lorsque la politique maximise la récompense immédiate et s'interprète comme suit. La première ligne représente le flux de gaz produit au cours du temps. On y observe des pics de production réguliers mais non identiquement espacés. La seconde ligne représente la quantité de gaz contenue par l'accumulateur au cours du temps. Il

apparaît clairement qu'il n'est pas utilisé. La troisième ligne représente le gaz vendu à l'usine de production d'électricité. On y voit que la quasi-totalité du flux injecté dans le réseau lors des pics de production est vendu. Ce comportement est cohérent, puisque c'est la récompense immédiate qui est maximisée. Les quatrième et cinquième lignes représentent respectivement l'évolution des puissances requises par les consommateurs 1 et 2. La dernière ligne représentent la récompense instantanée. Lorsqu'elle est négative, c'est qu'il n'y a pas assez de gaz pour alimenter les consommateurs, qui doivent en acheter sur le réseau extérieur. Lorsqu'elle est positive, c'est que du gaz est vendu à l'usine de production d'électricité. On voit ici aussi que l'ensemble du gaz est vendu dès que possible (moins le gaz utilisé pour l'alimentation des consommateurs au moment du pic de production) et qu'il est ensuite nécessaire d'acheter du gaz. Notons que ce n'est pas tout à fait le type de politique employée en pratique. Des heuristiques sont utilisées pour prendre en compte les accumulateurs, ce qui pourrait s'interpréter comme l'ajout d'un terme de récompense relatif à leur utilisation et qui permettrait d'obtenir un meilleur contrôle en maximisant la récompense immédiate. Nous nous intéressons maintenant au type de politique obtenu grâce à l'AR.

L'algorithme considéré est KTD-SARSA combiné à une politique de Thompson (voir section 3.4.4). Nous faisons certains choix pour la représentation de la fonction de valeur. Les capacités calorifiques étant constantes, il n'est pas nécessaire de considérer les composantes d'état associées. Au niveau du diviseur le choix des actions correspondant au routage du gaz n'a aucune influence sur l'évolution du système, uniquement sur la récompense immédiate. Les actions optimales correspondantes consistent donc à maximiser la récompense immédiate, ce qui peut être fait de manière analytique. C'est ce que nous faisons et les deux actions correspondantes ne sont pas considérées dans la Q-fonction. Sur un système réel, cela revient à dire que si une décision locale n'a aucune influence sur l'évolution du système et qu'il existe un algorithme efficace pour la prendre, autant l'utiliser. Au niveau des consommateurs, l'information de la puissance requise est redondante. On la retrouve dans les composantes d'état associées, mais également dans la récompense. En effet, si trop peu de gaz est fourni aux consommateurs, cela crée une contribution négative au terme de récompense instantanée. Nous ignorons donc ces composantes dans la fonction de qualité. Du point de vue de l'agent, cela rend l'information de récompense stochastique au lieu de déterministe. Le contrôle résultant peut être légèrement moins fin, mais cependant suffisant pour une validation du concept. Enfin, c'est la somme du flux de production et du flux propre à l'accumulateur qui est importante pour la prise de décision. Plutôt que de considérer les deux composantes d'état séparées pour la Q-fonction, nous en considérons la somme. Enfin, nous choisissons une somme pondérée de noyaux gaussiens pour la représentation de la fonction de qualité (les paramètres recherchés sont les poids de chacune des gaussiennes). Les actions étant continues, nous utilisons Monte Carlo lorsqu'il est nécessaire de calculer un maximum. Le type de politique obtenu par cette approche est représentée sur la figure 4.6.

La signification des différentes lignes est la même que pour la figure 4.5. On remarque que, cette fois-ci, la politique utilise bien l'accumulateur (deuxième ligne). Elle le remplit avec suffisamment de gaz pour assurer les besoins de consommation durant une période (période que nous rappelons ne pas être constante) et vend l'excédent à l'usine de production d'électricité (troisième ligne). Le taux de remplissage de l'accumulateur correspond environ à l'intégrale des besoins moyens entre deux pics de production. Si l'on observe la courbe de récompense (sixième ligne), il apparaît qu'elle est rarement négative, ce qui signifie que suffisamment de gaz est fourni aux consommateurs presque tout le temps. La récompense obtenue lors des pics de production (correspondant à la vente de gaz excédentaire donc) est plus faible que pour Monte Carlo. Ce type de comportement est typique des algorithmes d'AR et c'est ce que l'on souhaite observer. Il peut être nécessaire de choisir initialement une action moins récompensée qu'une autre pour obtenir un plus grand gain sur le long terme.

4.3 Conduite assistée

Les méthodes d'AR ont aussi été appliquées au problème de l'optimisation de la stratégie d'un système d'assistance à la conduite dans le cadre du projet Européen FP7 `ISi-PADAS` [245, 244] en collaboration avec le centre de recherches de Fiat et d'autres partenaires (DLR en Allemagne et l'INRETS en France par exemple). Un *Partially Autonomous Driving Assistance System* (PADAS) est généralement un système permettant de rendre la conduite plus agréable ou plus sûre pour le conducteur. Particulièrement,

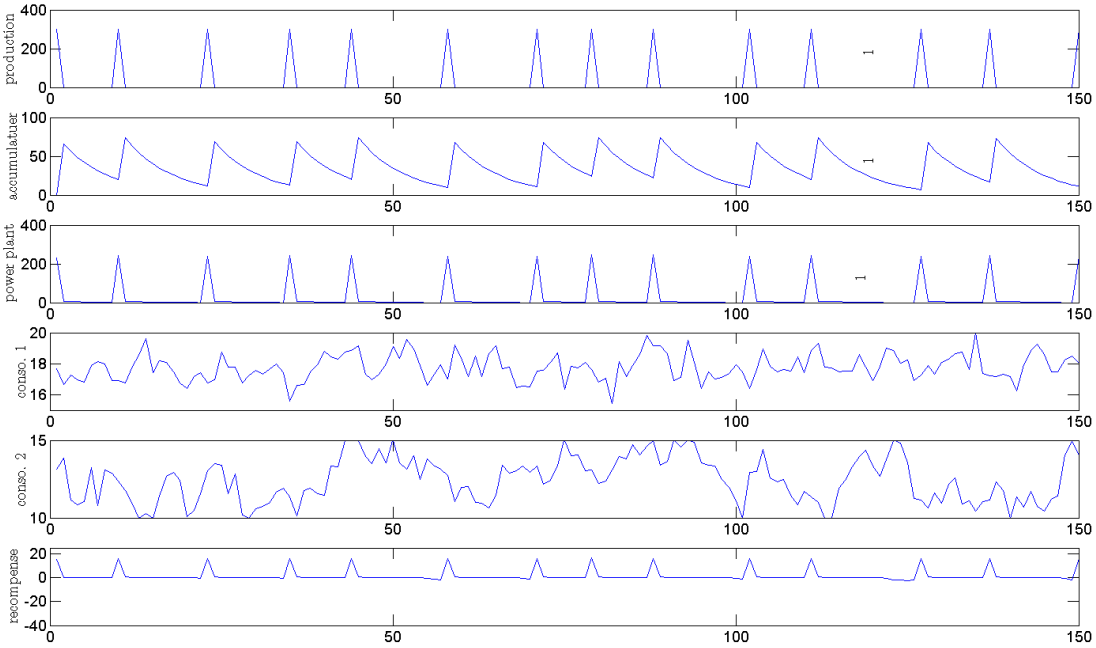


FIGURE 4.6 – Politique obtenue avec KTD.

le système étudié vise à minimiser le risque de collision avec un véhicule devançant celui pour lequel une politique de conduite assistée est cherchée. Cette politique doit atteindre cet objectif tout en minimisant les intervention du système de manière à ne pas gêner le conducteur ou rendre la conduite plus stressante. Pour cela, le système de conduite assistée peut signaler un danger en utilisant divers moyens comme des alertes audio, visuelles ou haptiques. Il peut aussi intervenir physiquement sur la vitesse du véhicule en freinant de manière plus ou moins importante.

Peu d'exemples d'application de l'apprentissage par renforcement au problème de l'optimisation de systèmes de conduite assistée ou autonome existent et la plupart de ces exemples se focalisent sur le pilotage du véhicule, c'est-à-dire le contrôle de la direction du véhicule [117, 139, 153]. Ici, nous nous intéressons à l'optimisation de l'assistance longitudinale du conducteur (sans changement de direction). Une intervention commence par des alertes de pré-collision et de collision et se termine éventuellement par une prise de contrôle du véhicule au travers d'un freinage assisté (le conducteur commence à freiner mais le système corrige ce freinage) et d'un freinage d'urgence (le système freine sans intervention du conducteur). La difficulté de cette tâche en comparaison à celle de piloter le véhicule de manière autonome est de tenir compte de la présence de l'humain dans la boucle de contrôle tant que le système d'assistance ne prend pas totalement le contrôle du véhicule. Cette présence introduit un caractère non-déterministe dans les réactions du système à contrôler qui est en fait composé du couple {véhicule+conducteur}. Cette stochasticité vient de variabilités inter- et intra-conducteur et conduit au fait qu'à une situation identique du point de vue du PADAS correspond des situations différentes suivant le conducteur ou même suivant la perception d'un même conducteur. Cette application partage ce problème avec le dialogue homme-machine décrit plus haut.

La tâche traitée dans cette section peut donc être vue comme un problème de décisions séquentielles dans lequel des séquences d'interactions {vehicule+conducteur}-PADAS sont observées. Chaque séquence $\{s_1, a_1, s_2, a_2, \ldots, s_T, a_T\}$ contient les états $\{s_i\}$ du système {véhicule+conducteur} et les réactions du PADAS $\{a_i\}$ à ces situations. Chaque séquence possède une probabilité de générer une collision et une stratégie ou politique optimale pour le PADAS est celle qui permet d'obtenir les séquences minimisant la probabilité de collision. Evidemment, nous ne connaissons pas les probabilités de collision associées à chacune des séquences. Nous disposons seulement de données enregistrées représentant un certain nombre de séquences. Il nous faudra donc apprendre hors ligne et *off-policy* une stratégie qui fonctionne aussi pour des états non-rencontrés dans la base. Ici se repose le problème de la généralisation.

4.3.1 PADAS et Processus Décisionnel de Markov

Pour transposer le problème décrit plus haut en PDM, il faut bien entendu définir les états du système {véhicule+conducteur} et les actions du PADAS ainsi qu'une fonction de récompense. L'état est en fait constitué d'une seule variable continue appelée *temps avant collision* et définie par

$$\theta_t = \frac{d_t}{v_t - v'_t}$$

où d_t est la distance au véhicule "obstacle", v_t est la vitesse du véhicule à contrôler et v'_t et la vitesse de l'obstacle. Ces variables peuvent être mesurées de manière assez précise par les instruments à bord d'un véhicule muni d'un PADAS. Les actions élémentaires que le PADAS peut réaliser sont de deux types :

- appliquer une pression de freinage égale à b fois la pression maximum autorisée par le véhicule ($b \in [0, 1]$)),
- envoyer un signal d'alerte au conducteur (les signaux pouvant être de nature auditive, visuelle ou haptique).

Les actions réelles du PADAS seront une combinaison de ces actions élémentaires. La fonction de récompense est définie sur base de 3 coûts différents :

- un coût associé au temps avant collision (inversement proportionnel à ce temps),
- un coût associé au freinage (plus le freinage est important, plus le coût est élevé)
- un coût associé à l'envoi d'un signal (à chaque signal est associé un niveau d'urgence ; plus le niveau est élevé, plus le coût est élevé).

Evidemment, un coût très important est associé à un temps avant collision de 0 puisqu'une collision est apparue. Aussi, on notera que la fonction de récompense vise à minimiser les actions du PADAS de sorte à ne pas perturber le conducteur de manière intempestive.

4.3.2 Expériences et résultats

Les expériences ont été divisées en trois phases : collecte de données, apprentissage de stratégie hors-ligne, tests. Une expérience de conduite a été réalisée sur un simulateur de voiture de type ScanerII [7]. Il s'agit d'un système fixe qui comprend une maquette de voiture, des commandes de conduites réelles (*i.e.* un siège, un volant, des pédales, un changement de vitesse et un frein à main), un tableau de bord numérique qui affiche les instruments de conduite traditionnels et un écran de projection où un environnement de conduite simulé est projeté. Ce type de simulateur permet d'effectuer des mesures sur tous les instruments avec une fréquence de 20 Hz (= 0.05 s). Cinq sujets ont participé à l'expérience qui consistait à conduire le véhicule dans des environnements extra-urbains et sur des autoroutes. Le véhicule disposait d'un PADAS implémentant une politique initiale d'intervention $p0$. Les essais ont permis de recueillir à peu près 600 minutes de données impliquant donc plusieurs conducteurs. La politique initiale p_0 était basée sur des heuristiques de bon sens (le temps avant collision a été discrétisé et à chaque intervalle était associée une action précise). Un bruit aléatoire a été introduit sur ces heuristiques afin d'explorer un peu l'espace d'état-action. Les valeurs de θ ont été calculées toutes les 300 ms ce qui a fourni un volume de données assez conséquent pour l'apprentissage. L'apprentissage s'est fait en utilisant une méthode *off-policy* évidemment puisque le contrôle du système n'était pas possible durant la collecte de données. Une politique p^* a été obtenue.

Des tests ont été effectués dans le même simulateur mais cette fois avec la politique apprise. Deux variables ont été mesurées durant cette phase de tests : le temps avant collision et la distance entre véhicules. Une collision est supposée apparue si le temps avant collision est tombé sous le seuil des 1.5 secondes ou si la distance inter-véhicules est plus petite que 1 mètre. Le Tableau 4.2 compare les résultats pour la politique apprise p^* et la politique à base d'heuristiques p_0 (mais sans le bruit stochastique sur les actions). Il est important de noter que la politique apprise a été obtenue à partir de données générées par une politique acceptable par les conducteurs qui ne se sont pas plaints des réactions du PADAS durant la phase de collection initiale (*off-policy* avec une politique comportementale acceptable). En effet, une politique initiale totalement aléatoire aurait certainement permis une meilleure couverture de l'espace d'état-action

7. www.scaner2.com

mais aurait engendré des comportements trop chaotiques pour le conducteur et donc des comportements de celui-ci qui ne se rencontreraient pas dans une situation réelle de conduite. Le Tableau 4.2 liste les pourcentage de trajectoires à l'issue desquelles une collision est apparue partant des définitions données plus haut. Ces résultats montrent que la stratégie apprise permet de réduire significativement le risque de collision ou le nombre de situations proches de la collision.

	Time to Collision	Distance
p^*	2.1%	1.5%
p_0	3.8%	4.3%

TABLE 4.2 – Résultats de la politique apprise (p^*) et de la politique heuristique (p_0)

4.3.3 Conclusion

Les résultats obtenus indiquent qu'il existe un réel bénéfice à utiliser cette approche : le pourcentage de séquences dont l'issue est une collision ou presque diminue de manière significative lorsqu'un PADAS optimisé est utilisé. L'autre avantage de cette approche est que la politique est obtenue directement à partir de données et qu'il n'est fait aucune hypothèse sur le conducteur (pas de modèle de conducteur etc.). Le caractère *off-policy* de la méthode d'apprentissage permet aussi de ne pas influencer le conducteur avec une politique changeante. De plus, des travaux sont en cours pour optimiser aussi la manière dont les signaux d'alertes sont présentés à l'utilisateur [244]. Ceci se rapproche des travaux réalisés sur les interfaces homme-machine décrits précédemment. Des résultats très prometteurs sont présentés dans [246].

4.4 Conclusions et perspectives

Les différentes applications développées dans ce chapitre sont des applications réelles faisant intervenir des industriels (France Telecom dans le cas du dialogue, ArcelorMittal dans le cas de l'optimisation de flux de gaz ou Fiat dans le cas de la conduite assistée). Les travaux effectués dans le cadre de l'AR suscitent donc un réel intérêt dans le monde industriel et les résultats obtenus sont jugés très prometteurs dans tous les cas.

Les apports de l'AR ont été clairement mis en évidence dans ces applications du fait du caractère très complexe du système à contrôler ce qui poussait précédemment à faire appel à des heuristiques (donc à l'intelligence, l'expérience et même l'intuition de l'humain) et pas à des méthodes d'optimisation. Un autre point commun à deux de ces applications (le dialogue et la conduite assistée) est l'apparition de l'humain à l'intérieur même du système à contrôler. Ceci le rend particulièrement non-stationnaire et renforce le caractère aléatoire de la dynamique du système. De plus, cela pose le problème de l'influence de la politique comportementale sur la reproductibilité des expériences et aussi de l'obligation d'apprendre à partir de trajectoires réelles et pas de transitions tirées au hasard. On retrouve dans les trois applications la nécessité de pouvoir apprendre *off-policy*.

Ces travaux ne sont donc pas restés à l'état d'expérience de laboratoire ou de test sur des problèmes jouets mais ont été réellement transposés à des applications pratiques en tenant compte des contraintes liées à la mise en œuvre dans des conditions réalistes d'exploitation. Plusieurs pistes pour poursuivre en ce sens sont envisagées au travers de nouvelles collaborations mais aussi grâce à la poursuite des collaborations en cours.

Deuxième partie

Traitement du Signal

Chapitre 5

Estimation en ligne de paramètres

Le problème d'Apprentissage par Renforcement (AR), décrit plus tôt, peut finalement être vu comme un problème d'estimation de paramètres dans le cas de l'approximation d'une fonction particulière qui est la fonction de valeur ou la Q-fonction d'un Processus Décisionnel de Markov (PDM). La particularité de ce problème était que la fonction dont on cherche les paramètres n'était pas directement observable. Dans un grand nombre d'applications autres que celle de l'AR, le problème de trouver les paramètres optimaux d'une fonction ou d'un modèle se pose de la même manière avec des contraintes différentes.

C'est le cas par exemple lorsque l'on modélise un signal par un modèle autorégressif :

$$x_i = \sum_{j=1}^{n} \alpha_j x_{i-j} + e_i$$

On cherche alors le jeu de n paramètres $\{\alpha_j\}_{1 \leq j \leq n}$ qui permet au modèle de représenter au mieux la réalité (voir Section 5.2). Ce peut aussi être le cas lorsqu l'on cherche à débruiter un signal par une modélisation de celui-ci par une somme pondérée de fonctions de base (voir Section 5.3).

Un autre exemple serait la réalisation d'une cartographie de l'intensité d'un réseau sans fil. Une manière de faire serait d'utiliser les équations de Maxwell et une connaissance précise de la topologie des lieux et des matériaux utilisés. Néanmoins, cela deviendrait rapidement assez complexe. Une manière différente de faire est de considérer un ensemble de mesures contenant des positions (p) et des intensités (i_p) et d'essayer de généraliser ces mesures à d'autres positions dans le bâtiment. On chercherait donc une fonction $f(p) = i_p$. C'est le problème général de régression qui consiste souvent en la recherche de paramètres permettant de choisir une fonction optimale parmi une famille de fonctions paramétrées.

Ainsi, il est fréquent de disposer d'une base de données $\{x_i, y_i\}$, où les x_i sont des données d'entrée et les y_i des données de sortie dont on suppose qu'elles ont été générées par un modèle $y = f(x, \theta) + b$ ou $y = f_\theta(x) + b$. Dans ce modèle, θ représente un vecteur de paramètres qui sont à déterminer à partir des données et b un bruit. Nous nous posons donc le problème d'estimer le modèle par une fonction approchée $\hat{f}_\theta$.

Dans ce chapitre, nous exposons un travail de modélisation de problèmes pratiques de traitement du signal pour ensuite bénéficier de solutions issues de travaux en filtrage comme précédemment. Nous nous intéresserons à plusieurs applications de l'estimation de paramètres en ligne faisant utilisation du filtre de Kalman et de ses extensions. Cela requiert donc une phase préliminaire de modélisation de notre problème dans le formalisme *espace d'état*. Deux problèmes pratiques particuliers sont vus dans le domaine biomédical et particulièrement focalisées sur le traitement du signal électrocardiogramme (ECG) dans un système d'acquisition d'Imagerie par Résonance Magnétique (IRM). Ici, la prise en compte de l'humain est très profonde puisqu'il sera nécessaire de comprendre le fonctionnement du cœur pour réaliser la modélisation. Ces travaux ont été réalisés dans le cadre de la thèse de Julien Oster [154], en collaboration avec l'équipe INSERM IADI située au Centre Hospitalier Universitaire (CHU) de Nancy et la société Schiller Medical qui a financé la thèse. Un travail important de modélisation de ce signal et de ses variations sera nécessaire et constitue la contribution majeure de ce travail. La thèse s'étant réalisée dans un environnement clinique et industriel, les applications ont été importantes et les contraintes réelles ont piloté le travail.

5.1 L'estimation comme un problème de filtrage

5.1.1 Reformulation du problème

Comme dans la Section 2.2, nous adoptons ici un point de vue statistique permettant de reformuler le problème comme l'estimation de l'état caché d'un système. Ceci nous permettra d'estimer les paramètres de notre modèle en ligne et ainsi d'améliorer notre estimation après chaque nouvelle information. Ainsi, nous considérerons que le système que nous cherchons à modéliser est régi par un certain nombre de paramètres que l'on rangera dans un vecteur θ. Les manifestations de l'influence de ces paramètres sur le système se traduisent par un certain nombre d'observations $\{y_i\}$ qui seront mises en relation avec le fonctionnement du système par le biais de l'équation caractérisant le modèle f_θ. Ces observations seront valables pour des entrées $\{x_i\}$ particulières du système. Nous pouvons donc reformuler notre problème d'estimation de paramètres sous la forme d'*espace d'état* décrite dans la Section 2.2 et donc donner les équations d'évolution et d'observation liées à ce problème :

$$\begin{cases} \theta_i &= \theta_{i-1} + v_i \\ y_i &= \hat{f}_{\theta_i}(x_i) + n_i \end{cases} \tag{5.1}$$

Ici encore nous adoptons un modèle de marche aléatoire pour les paramètres que nous cherchons. Ce modèle n'a pas vraiment de justification physique mais possède plusieurs avantages. Il est tout d'abord assez simple, il permet d'éviter les minima locaux en imposant une recherche aléatoire dans l'espace des paramètres et enfin il permet éventuellement de suivre l'évolution temporelle des paramètres si le modèle devait être non-stationnaire. Les variables v_i et n_i sont, comme dans la Section 2.2 des bruits additifs que l'on considérera blancs, indépendants et centrés.

$$\begin{aligned} E[v_i] = E[n_i] = 0, \\ E[v_i \cdot n_j] = 0 \quad \forall i, j, \\ E[v_j \cdot v_i] = E[n_j \cdot n_i] = 0 \quad \forall i \neq j. \end{aligned}$$

Les hypothèses de blancheur des bruits peuvent être éventuellement levées en introduisant des bruits Auto-Régressif (AR) (comme dans la section 3.3.1) ou même des bruits *Auto-Regressive Moving Average* (ARMA).

Une solution à ce problème est donnée par le filtrage bayésien qui calcule la densité de probabilité *a posteriori* $p(\theta_i|y_{1:i})$ des paramètres étant donné les sorties $\{y_k\}_{0\leq k\leq i}$ observées jusqu'au temps i (notée $y_{1:i}$) en utilisant les équations suivantes basées sur la règle de Bayes :

$$\begin{aligned} p(\theta_i|y_{1:i-1}) &= \int_\theta p(\theta_i|\theta_{i-1}) p(\theta_{i-1}|y_{1:i-1}) d\theta_{i-1} \\ p(\theta_i|y_{1:i}) &= \frac{p(y_i|\theta_i) p(\theta_i|y_{1:i-1})}{\int_\theta p(y_i|\theta_i) p(\theta_i|y_{1:i-1}) d\theta_i} \end{aligned} \tag{5.2}$$

Une estimation des paramètres que l'on cherche à l'instant i est alors donnée par l'espérance des paramètres étant donné cette distribution *a posteriori* :

$$\hat{\theta}_{i:i} = E[\theta_i|y_{1:i}] = \int_\theta \theta_i p(\theta_i|y_{1:i}) d\theta_i$$

En pratique, les équations 5.2 ne peuvent pas être résolues analytiquement dans la plupart des cas, même en connaissant l'expression de $\hat{f}_\theta$. C'est pourquoi on s'attache en général à n'estimer que les moments d'ordres un et deux de ces distributions ce qui donne lieu au filtrage de Kalman [109]. Des estimations plus complexes de ces densités sont possibles en utilisant des méthodes comme le filtrage particulaire [27] mais nous en resterons à l'approche de Kalman. Rappelons que le filtre de Kalman vise, dans notre cas, à calculer le gain optimal K_i qui permette une mise à jour linéaire optimale de notre estimation des paramètres :

$$\hat{\theta}_{i|i} = \hat{\theta}_{i|i-1} + K_i(y_i - \hat{y}_{i|i-1}) = \hat{\theta}_{i|i-1} + K_i e_i, \tag{5.3}$$

avec

$$\begin{aligned}\hat{\theta}_{i|i-1} &= E[\theta_i|y_1,\dots,y_{i-1}] = E[\theta_{i-1}+v_{i-1}|y_1,\dots,y_{i-1}] \\ &= E[\theta_{i-1}|y_1,\dots,y_{i-1}] = \hat{\theta}_{i-1|i-1}, \qquad (5.4)\\ \hat{y}_{i|i-1} &= E[y_i|y_1,\dots,y_{i-1}] = E[f_{\theta_i}(x_i)+n_i|y_1,\dots,y_{i-1}] \\ &= E[f_{\theta_i}(x_i)|y_1,\dots,y_{i-1}] = E[f_{\hat{\theta}_{i-1|i-1}}(x_i)], \qquad (5.5)\\ K_i &= P_{\theta e_i} P_{e_i}^{-1}. \qquad (5.6)\end{aligned}$$

et où $P_{\theta e_i} = E[(\theta_i - \hat{\theta}_{i|i-1})e_i|y_1,\dots,y_{i-1}]$ et $P_{e_i} = \mathrm{cov}(e_i|y_1,\dots,y_{i-1})$.

Nous ne revenons pas sur les développements qui ont permis d'aboutir à ces résultats puisqu'ils ont été réalisés dans le cas de l'approximation de la fonction de valeur dans la Section 3. Rappelons tout de même que, comme dans le cas des différences temporelles de Kalman ou *Kalman Temporal Differences* (KTD), le filtrage de Kalman permet d'estimer en permanence l'incertitude sur les paramètres que nous cherchons et donc de rétro-propager cette incertitude sur la fonction elle-même.

5.1.2 Mise en pratique

La mise à jour (5.3) nécessite le calcul du gain de Kalman K_i qui implique le calcul des statistiques $P_{\theta e_i}$ et P_{e_i} ainsi que de l'espérance de $f_{\hat{\theta}_{i-1|i-1}}(x_i)$. Ces statistiques sont encore une fois difficilement calculables dans le cas général. Dans son article fondateur, Kalman [109] suppose toutes les équations linéaires et déduit une expression analytique du gain. Dans notre cas, cela supposerait que f_θ soit linéaire en les paramètres θ. Nous souhaitons éviter de faire ce type de suppositions puisque nous ne connaissons *a priori* rien des modèles pour lesquels nous cherchons des paramètres. Pour s'affranchir de cette contrainte de linéarité de la fonction f_θ, plusieurs solutions sont possibles.

Filtre de Kalman Etendu ou *Extended Kalman Filter* (EKF)

Une première méthode, connue sous le nom de Filtre de Kalman Etendu ou *Extended Kalman Filter* (EKF) [235] consiste à linéariser (lorsque cela est possible) les équations d'état autour d'un point d'intérêt. Dans notre cas, il s'agit seulement de linéariser l'équation d'observation ce qui nous donne :

$$\begin{aligned}\theta_i &= \theta_{i-1} + v_i \\ y_i &= f_{\hat{\theta}_i}(x_i) + g(\theta_i - \hat{\theta}_{i|i}, x_i) + w_i \qquad (5.7)\end{aligned}$$

où $g(\theta, x) = \dfrac{\partial f_\theta(x)}{\partial \theta}|_{\theta=\hat{\theta}_{i|i}}$

Ce système d'équations devient alors linéaire et il peut être intégré simplement au développement qui débouche sur le gain de Kalman. Plusieurs soucis peuvent empêcher le bon fonctionnement de cette méthode. Tout d'abord, la fonction f_θ peut ne pas être dérivable (c'était le cas de l'équation d'optimalité de Bellman). Ensuite, l'approximation linéaire peut être relativement loin de la réalité.

Filtre de Kalman non-parfumé ou *Unscented Kalman Filter* (UKF)

Il est aussi possible de réaliser une linéarisation statistique de la fonction f_θ autour du point d'étude (c'est à dire l'estimation courante $\theta_{i|i}$). C'est le principe de la transformation non parfumée dont l'idée est qu'il est plus facile d'approximer une variable aléatoire qu'une fonctionnelle non-linéaire quelconque. Ce principe a été développé dans la Section 2.2 et spécialisé pour l'approximation de la fonction de valeur dans le Chapitre 3. Cette méthode est aussi connue sous le terme de *Sigma Point Kalman Filter* (SPKF) puisqu'elle se base sur le calcul de points particuliers appelés "*sigma-points*". Nous spécialisons ici cette méthode pour l'espace d'état qui nous intéresse pour l'estimation de paramètres :

$$\begin{cases} \Theta^{(0)} = \bar{\theta} & w_0 = \frac{\kappa}{n+\kappa}, \quad j = 0 \\ \Theta^{(j)} = \bar{\theta} + \left(\sqrt{(n+\kappa)P_\theta}\right)_j & w_j = \frac{1}{2(n+\kappa)}, \quad 1 \leq j \leq n \\ \Theta^{(j)} = \bar{\theta} - \left(\sqrt{(n+\kappa)P_\theta}\right)_{n-j} & w_j = \frac{1}{2(n+\kappa)}, \quad n+1 \leq j \leq 2n \end{cases} \tag{5.8}$$

où $\bar{\theta}$ est la moyenne de θ, P_θ est sa matrice de variance, κ est un coefficient d'échelle permettant de contrôler la précision de la transformation non-parfumée [107], et $\left(\sqrt{(n+\kappa)P_\theta}\right)_j$ est la $j^{\text{ème}}$ colonne de la décomposition de Cholesky de la matrice $(n+\kappa)P_\theta$. L'image de chacun de ces points par f est ensuite calculée : $\mathcal{Y}^{(j)} = f(\Theta^{(j)})$, $0 \leq j \leq 2n$. L'ensemble des sigma-points et de leurs images peut alors être utilisé pour calculer les moments d'ordres 1 et 2 de y, et même $P_{\theta y}$, la covariance entre θ et y :

$$\begin{cases} \bar{y} \approx \bar{\mathcal{Y}} = \sum_{j=0}^{2n} w_j \mathcal{Y}^{(j)} \\ P_y \approx \sum_{j=0}^{2n} w_j \left(\mathcal{Y}^{(j)} - \bar{\mathcal{Y}}\right)\left(\mathcal{Y}^{(j)} - \bar{\mathcal{Y}}\right)^T \\ P_{\theta Y} \approx \sum_{j=0}^{2n} w_j \left(\Theta^{(j)} - \bar{\theta}\right)\left(\mathcal{Y}^{(j)} - \bar{\mathcal{Y}}\right)^T \end{cases} \tag{5.9}$$

5.2 Synchronisation pour l'IRM cardiaque

5.2.1 Position du problème

L'IRM est un processus séquentiel complexe dont la durée d'acquisition est relativement importante. La formation des images provient d'un phénomène de résonance magnétique nucléaire agissant sur le spin des protons dans les molécules d'eau. Ainsi, quand ils sont placés dans un champs magnétique intense, leurs spins sont à l'origine d'une aimantation tissulaire macroscopique correspondant à un état d'équilibre statistique. Leur excitation par une onde radiofréquence à la fréquence dite de Larmor, qui dépend du champ magnétique, provoque un basculement de cette aimantation, d'un angle qui dépend de l'intensité et de la durée de l'onde radiofréquence. Lorsque l'onde radiofréquence est stoppée, les protons opèrent une relaxation les ramenant à l'équilibre en des temps différents suivant la nature des tissus (en fait, de leur concentration en eau). Ce basculement résulte en un signal qui est à l'origine de la formation des images, suivant des transformations complexes. En 1975, Richard Ernst a proposé l'utilisation de la transformée de Fourier pour l'analyse du signal IRM, ce qui lui a valu le prix Nobel de chimie en 1991. Ainsi, on peut considérer que l'IRM réalise des acquisitions dans le domaine de Fourier, remplissant au fur et à mesure les lignes de la transformée de Fourier de l'image que l'on cherche. La mesure des temps de relaxation est relativement complexe et nécessite du matériel coûteux. On ne peut donc réaliser l'imagerie entière d'un organe en une seule acquisition et plusieurs prises sont nécessaires pour reconstituer la transformée de Fourier d'une image globale. Les mouvements de l'organe entre deux acquisitions introduisent des artéfacts complexes (plus complexes qu'un flou de mouvement). Ceci implique qu'il faut, lorsque l'organe à visualiser est en mouvement (comme le cœur), mettre en place des méthodes permettant de prendre en compte ce mouvement et de réaliser des acquisitions dans des positions identiques de l'organe. C'est ce à quoi nous nous attachons dans ce chapitre qui traitera de la synchronisation de l'acquisition pour l'IRM cardiaque. Cette problématique a été traitée de manière approfondie par [232]. Nous décrivons ici le problème en termes physiologiques afin de motiver notre modélisation du problème. Le lecteur est invité à se référer à la thèse de J. OSTER [154] d'où sont issus ces travaux si des détails supplémentaires sont nécessaires à la compréhension.

Le mouvement cardiaque est communément supposé pseudo-périodique, c'est-à-dire que chaque battement cardiaque est supposé identique au précédent. Aucune hypothèse supplémentaire n'est donc émise en fonction de la respiration ou de l'apnée du patient. Le signal ECG est habituellement utilisé afin de synchroniser l'acquisition IRM avec le mouvement cardiaque. Il consiste en l'enregistrement de l'activité électrique du cœur. C'est un enregistrement non-invasif, effectué par un appareil électrocardiographique via des électrodes positionnées sur la peau du patient. C'est devenu au fil du temps un outil diagnostique primordial pour les maladies cardio-vasculaires.

Le signal ECG est la résultante de la dépolarisation successive de différentes cellules, dont la source ou excitation réside dans l'organe même. Cette excitation se produit spontanément dans le noeud sinusal,

qui est le "pacemaker" (ou métronome) du cœur. C'est la fréquence des impulsions issues du noeud sinusal qui détermine la fréquence des battements ou rythme cardiaque. En effet, bien que toutes les parties du système excitateur et conducteur du cœur aient la capacité de se dépolariser spontanément, leur fréquence propre est plus basse et la fréquence du nœud sinusal prédomine.

L'excitation se propage à partir de ce point aux deux oreillettes, avant d'être retardée dans le nœud atrioventriculaire. Cette phase correspond à la contraction des oreillettes. La propagation de l'excitation se poursuit alors le long du faisceau de His ("Common Bundle"), à travers le réseau de Purkinje ("Purkinje Fibers") pour conduire à l'excitation du myocarde ventriculaire. Ceci entraîne alors la contraction des ventricules. La figure 5.1 illustre la propagation de l'activité électrique cardiaque.

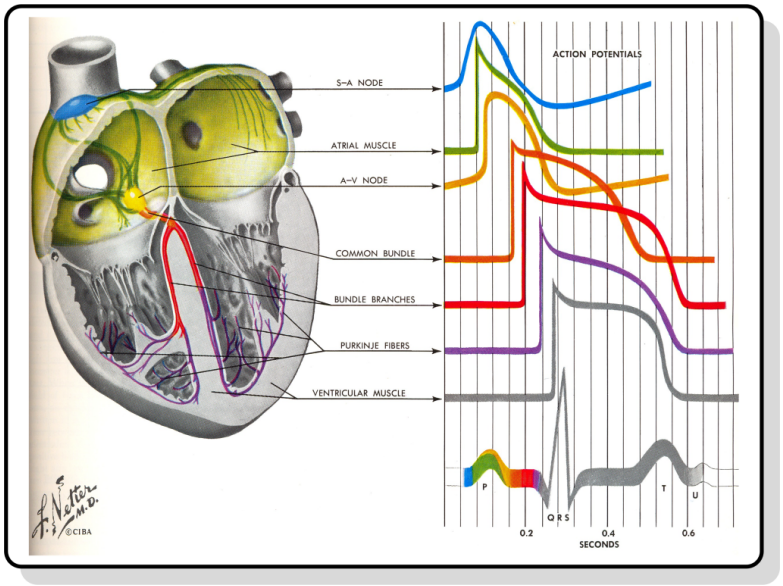


FIGURE 5.1 – Dépolarisation des cellules cardiaques et création du signal ECG [146].

Le potentiel électrique résultant de l'excitation du cœur peut être représentée par un vecteur dont l'amplitude et la direction sont caractéristiques. Le signal ECG est la résultante totale de ces vecteurs au niveau de la peau que l'on projette selon un axe, appelé dérivation, déterminé par le placement des électrodes.

Un signal ECG se caractérise par les cinq déflexions principales. L'onde P résulte de la dépolarisation des oreillettes. Le complexe QRS est constitué des trois ondes Q, R et S, qui sont le fruit de la dépolarisation des ventricules. L'onde R est la déflexion la plus facilement détectable sur le signal, avec une amplitude de l'ordre du millième de Volt (mV), selon la dérivation. Enfin, la phase de repolarisation des ventricules est à l'origine de l'onde T.

Les complexes QRS sont détectés et servent de déclencheur, ou "*trigger*", pour l'acquisition IRM. C'est pourquoi la synchronisation de l'acquisition IRM est également appelée "*triggering*". Deux modes de synchronisation existent. Soit la synchronisation est dite prospective, auquel cas une détection d'un complexe QRS lance un salve d'acquisitions IRM après un délai prédeterminé (figure 5.2). Soit la synchronisation est rétrospective, auquel cas l'acquisition IRM est faite en continu sur plusieurs phases cardiaques et les données sont triées à la fin de l'acquisition IRM en fonction des détections de complexes QRS (figure 5.3). Evidemment, la méthode préventive est plus économe mais nécessite une bonne prédiction de l'intervalle entre deux ondes R.

Néanmoins, du fait de la présence de bruit sur l'ECG, la détection des complexes QRS est une tâche délicate. Or la qualité de l'image IRM découle de manière directe de la synchronisation et donc de la détection des complexes QRS (figure 5.4), et celle-ci est actuellement jugée insuffisante. L'exemple de mauvaise synchronisation exposée sur la figure 5.4, illustre la présence de fantômes sur l'image de gauche, qui sont causées par des mouvements cardiaques entre les différentes salves d'acquisition.

Une des pistes actuellement envisagées est l'augmentation de la résolution temporelle des acquisi-

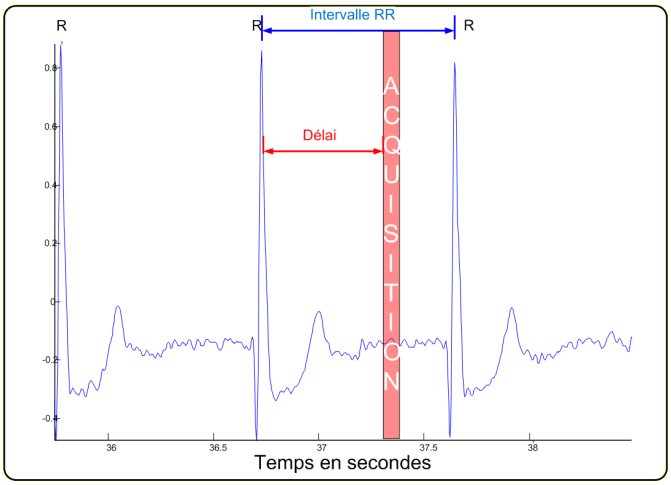


FIGURE 5.2 – Schéma de synchronisation cardiaque prospective.

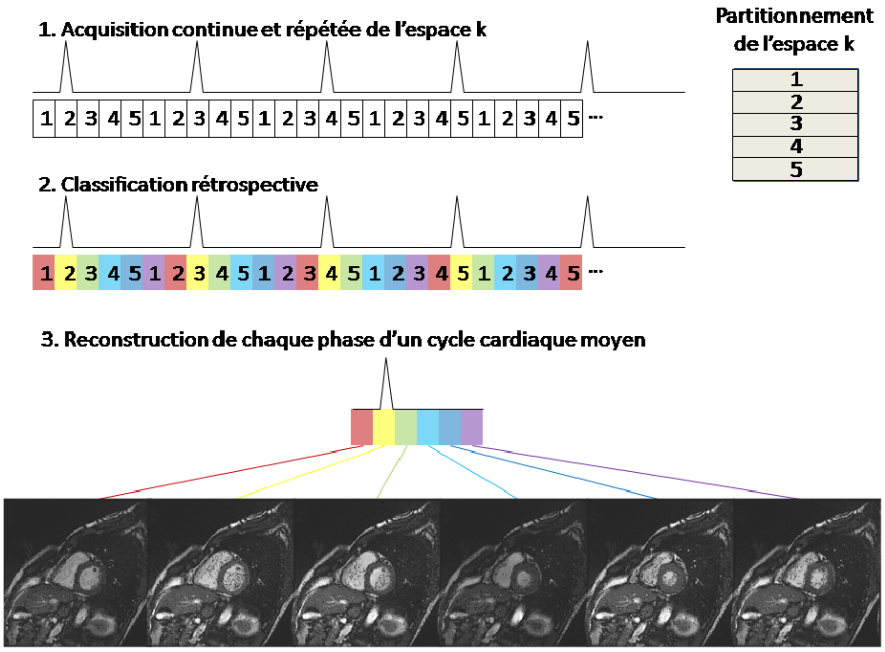


FIGURE 5.3 – Schéma de synchronisation cardiaque rétrospective.

tions IRM afin de se soustraire de la synchronisation cardiaque. Malheureusement ceci se fait au détriment de la résolution spatiale et la qualité d'image [232]. D'autres solutions sont aussi envisagées (oxymétre de pouls [47], acoustocardiogramme [63, 64, 149, 148] *etc.*) mais fournissent des résultats peu satisfaisants. L'enregistrement ECG reste donc à l'heure actuelle le meilleur outil de synchronisation de l'acquisition IRM.

La synchronisation de l'acquisition IRM avec l'activité cardiaque passe donc par la détection des complexes QRS, ou plus précisément de l'onde R. C'est pourquoi nous avons proposé une méthode originale de détection basée sur l'analyse en ondelettes [158] que nous ne détaillons pas ici mais qui s'est avérée particulièrement efficace. Néanmoins, ceci n'est pas suffisant pour une synchronisation préventive optimale.

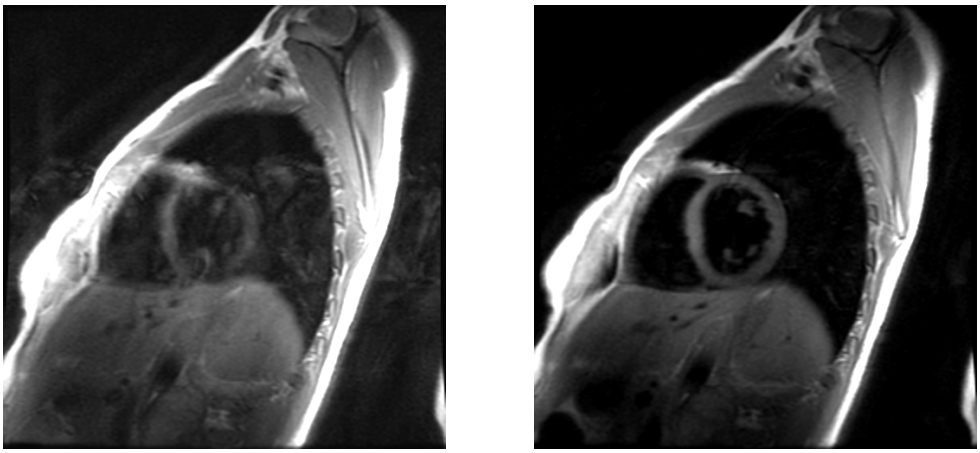


FIGURE 5.4 – Influence de la synchronisation sur la qualité de l'image, exemple de mauvaise synchronisation (à gauche) et de bonne synchronisation (à droite).

5.2.2 Prédiction de l'intervalle RR

L'imagerie cardiaque par résonance magnétique nécessite donc la mise en place de méthodes de synchronisation de l'acquisition avec le mouvement du cœur. Le signal ECG représente actuellement la meilleure solution pour la réalisation de cette tâche.

L'hypothèse qui est généralement émise pour la prise d'image est que le mouvement cardiaque est reproductible d'un cycle sur l'autre [232]. Le rythme cardiaque est donc supposé constant. Cette hypothèse est bien évidemment erronée et ce modèle est une simplification trop grande de la réalité. En effet, le rythme cardiaque est variable au cours du temps. Sur une échelle de quelques minutes, cette variation, mesurée par la durée entre deux ondes R et appelée "*intervalle RR*", est principalement due à la respiration. Ce phénomène, appelé *Respiratory Sinus Arrythmia* (RSA) [41], induit une variation sinusoïdale du rythme cardiaque de même fréquence que la respiration. Or, afin de minimiser les artefacts de mouvement sur l'image, les acquisitions sont essentiellement réalisées pendant que le patient est en apnée. Cette technique, qui permet de supprimer les artefacts de mouvement sur une image, ne permet pas de réduire les variations du rythme cardiaque [207]. En conclusion, le rythme cardiaque ne peut pas être supposé constant durant une acquisition IRM (figure 5.5).

En IRM, différentes séquences d'acquisition permettent de mettre en évidence telle ou telle propriété de l'organe ou d'annuler le signal émis par certains tissus. Certaines séquences IRM nécessitent un temps de préparation, dépendant des temps de relaxation des tissus à visualiser ou dont le signal veut être supprimé. Par exemple pour les séquences *Double Inversion Recovery Fast Spin Echo* (DIR-FSE) [236], un jeu d'impulsions radio fréquence doit être lancé un certain temps TI (Temps d'Inversion) avant d'acquérir l'image. Les images ainsi obtenues sont dites en *sang noir* (le signal du sang étant annulé) et permettent une meilleure visualisation du myocarde (meilleur contraste). Le TI peut être calculé à l'aide de la formule de Fleckenstein [62] et est généralement de l'ordre de $500ms$. L'acquisition d'images en sang noir est de ce fait impossible actuellement en systole (phase de contraction du cœur) du fait de la durée restreinte de cette phase cardiaque, à moins de lancer les impulsions *Double Inversion Recovery* (DIR) avant l'onde R (avant la systole).

Dans cette section, une méthode de prédiction RR, basée sur le filtrage de Kalman, va être proposée afin d'acquérir des images en sang noir en systole [155, 160].

5.2.3 Méthode

En routine clinique, une onde R détectée provoque immédiatement les impulsions DIR et l'acquisition du signal est alors réalisée après un temps imposé par le TI du sang (cf. figure 5.6). L'ensemble des acquisitions sang noir sur le cœur est alors réalisé durant la phase de repos isovolumétrique du ventricule

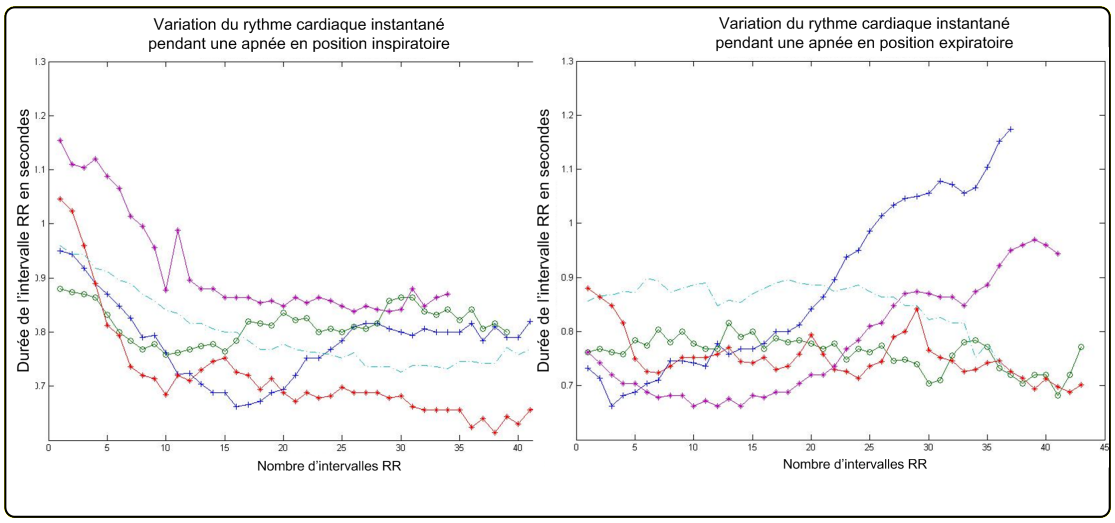


FIGURE 5.5 – Exemples de variation du rythme cardiaque instantané pour cinq sujets pendant une apnée en postion inspiratoire (à gauche) et expiratoire (à droite).

en diastole (position recherchée de repos).

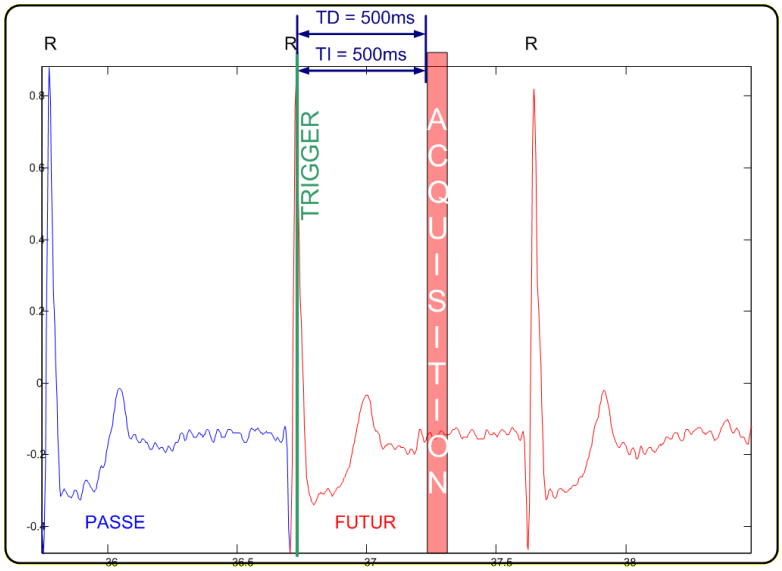


FIGURE 5.6 – Schéma de synchronisation des séquences DIR-FSE

Afin d'acquérir des images en sang noir en systole, soit un certain temps, TD, après l'onde R il est alors nécessaire de retarder le lancement des impulsions DIR de sorte qu'elles précèdent d'un temps TI, la prochaine phase isovolumétrique en systole, dans le cycle cardiaque suivant. Le retard optimal de lancement des impulsions DIR, TD_{SAEC}, doit être calculé à l'aide d'une prédiction du cycle RR à venir, $\hat{RR}_{n+1}$, par la formule :

$$TD_{SAEC} = \hat{RR}_{n+1} + TD - TI. \tag{5.10}$$

La synchronisation d'une séquence en sang noir en systole est alors réalisée suivant le schéma de la figure 5.7.

Il reste alors à déterminer une méthode de prédiction des intervalles RR. La variation de l'intervalle RR est connue pour être liée à la respiration, par le phénomène RSA. L'utilisation du signal respira-

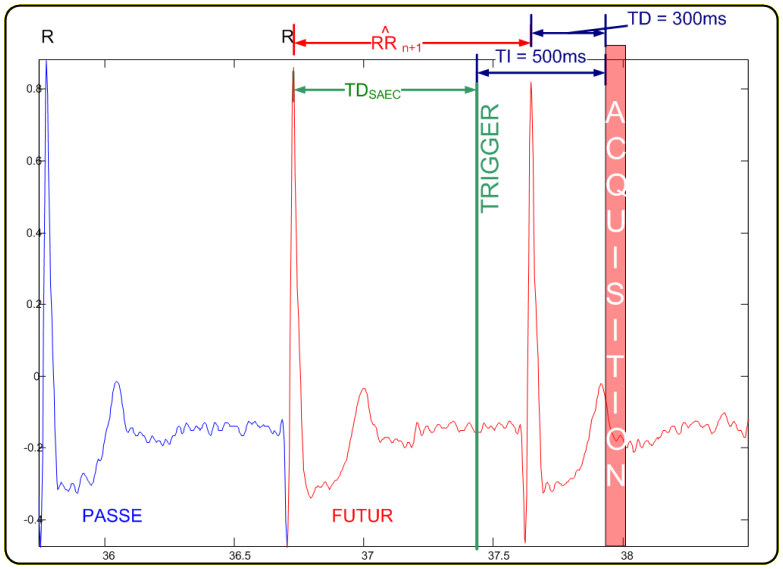


FIGURE 5.7 – Schéma de synchronisation des séquences sang noir en systole

toire peut alors s'avérer d'une grande importance pour aider à prédire le rythme cardiaque. Le signal de variation des intervalles RR peut être modélisé comme une sortie d'un filtre ARMA avec un signal de respiration en entrée [39]. Soient S_n un signal de respiration et Δ_n le signal d'intervalle RR, le signal du futur intervalle RR peut alors être modélisé par :

$$\Delta_{n+1} = \sum_{i=0}^{n_1} a_i \Delta_{n-i} + \sum_{j=0}^{n_2} b_j S_{n-j} + \eta_n, \tag{5.11}$$

où η_n est un bruit et les a_i et b_i des coefficients à trouver.

Les signaux de respiration peuvent être acquis en routine clinique par le biais de ceintures pneumatiques attachées sur l'abdomen du patient. Il est donc possible d'envisager leur intégration dans le modèle de prédiction. L'échantillonnage des signaux respiratoires est un souci pour leur intégration dans le modèle 5.11. En effet, prendre en compte un signal de respiration à chaque onde R, ou nouvelle mesure de rythme cardiaque, ne permet pas d'apporter une information complète, en particulier lors des phases de respiration rapide qui précèdent les apnées. Afin de pallier cette imperfection, il est intéressant d'intégrer également la dérivée de ce signal, dS_n, qui apporte une information sur la dynamique de la respiration. De plus, il est montré dans [51] que l'amplitude A_n des ondes R, est corrélée avec la respiration. L'utilisation de ces signaux est alors également envisageable.

Un nouveau modèle de prédiction du rythme cardiaque peut alors être formulé, celui-ci étant une régression basée sur plusieurs signaux physiologiques et que nous avons dénommé *Physiological Multi Input Regression* (PMIR) :

$$\Delta_{n+1} = \sum_{i=0}^{n_1} a_i \Delta_{n-i} + \sum_{j=0}^{n_2} b_j S_{n-j} + \sum_{k=0}^{n_3} c_k dS_{n-k} + \sum_{l=0}^{n_4} d_l A_{n-l}. \tag{5.12}$$

Il est alors possible de prédire l'intervalle suivant, en estimant les paramètres (a_i, b_i, c_i, d_i). Cette estimation peut être mise en place par une technique de filtrage de Kalman.

En effet, ce problème peut être exprimé dans une formulation d'espace d'état. Les paramètres, a_i, b_i, c_i, d_i, peuvent être considérés comme un vecteur d'état suivant une marche aléatoire. La formulation du problème est donc la suivante :

$$\begin{cases} \theta_k = \theta_{k-1} + w_{k-1} \\ \Delta_k = C_k \theta_k + v_k \end{cases},$$

Le vecteur d'état et la matrice d'observation sont définis comme suit :

$$\theta = [a_{k,0}, \ldots, a_{k,n_1}, b_{k,0}, \ldots, b_{k,n_2}, c_{k,0}, \ldots, c_{k,n_3}, d_{k,0}, \ldots, d_{k,n_4}]^T$$

et

$$C_k = [\Delta_k, \ldots, \Delta_{k-n_1}, S_k, \ldots, S_{k-n_2}, dS_k, \ldots, dS_{k-n_3}, A_k, \ldots, A_{k-n_4}].$$

La formulation espace d'état étant complètement linéaire, il est alors possible de mettre en place le filtre de Kalman simple. L'application étant tournée vers une acquisition IRM en apnée, l'horizon d'observation est court. L'incertitude sur le modèle, qui est déterminée par le bruit de la marche aléatoire, doit être élevée, afin que les paramètres de prédiction évoluent rapidement. De plus, l'initialisation du premier élément du vecteur d'état est $a_{0,0} = 1$ et les autres éléments sont initialisés à 0. Un schéma de la méthode PMIR est représenté sur la figure 5.8.

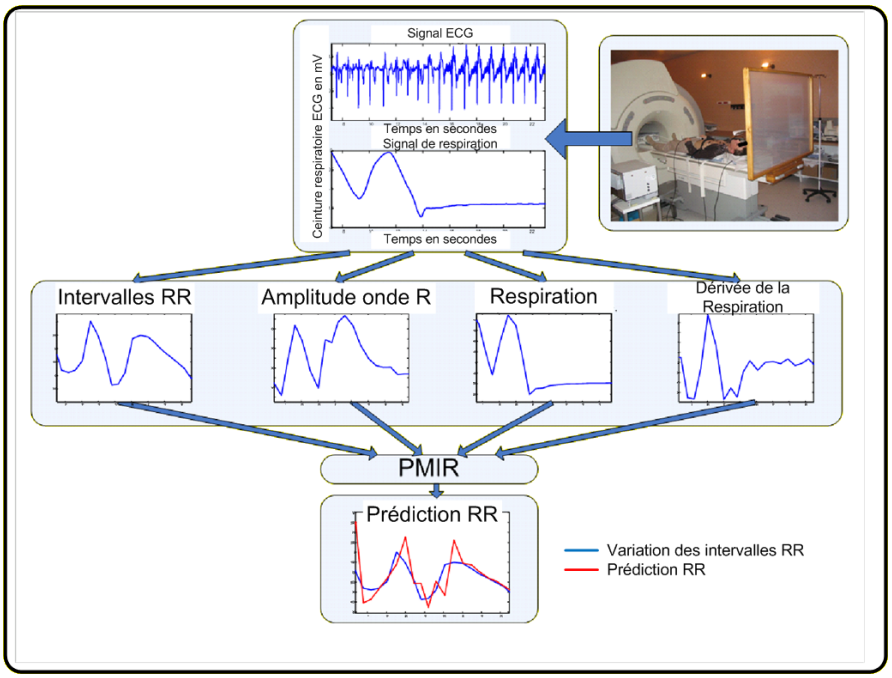


FIGURE 5.8 – Schéma de la méthode PMIR.

5.2.4 Résultats

Nous nous sommes attachés, comme précédemment, à démontrer l'intérêt de cette méthode sur des cas pratique. Ainsi l'approche abordée a été validée dans un premier temps, lors d'une étude contenant 5 sujets au CHU de Nancy. Chaque sujet a réalisé deux séries de deux apnées, chaque série étant réalisée dans une position respiratoire différente. Les apnées ont été réalisées en fin d'expiration ou en fin d'inspiration.

La méthode de prédiction a été alors mise en place, en post-traitement, la première apnée servant de calibration pour la prédiction. A la fin de cette première apnée, les paramètres du filtre de Kalman sont conservés et réutilisés pour la deuxième apnée, sur laquelle la qualité de prédiction est déterminée.

La méthode de prédiction a été évaluée, en mesurant l'erreur en moyenne quadratique normalisée ou *Normalised Mean Squared Error* (NMSE) de la prédiction qui est déterminée par :

$$\text{NMSE} = \frac{\sum_{i=1}^{n} \left(\hat{\Delta}_i - \Delta_i\right)^2}{\sum_{i=1}^{n} \left(\Delta_i - \bar{\Delta}\right)^2}, \tag{5.13}$$

où $\bar{\Delta}$ correspond à la moyenne des intervalles RR pendant l'apnée $\bar{\Delta} = 1/n \sum_{i=1}^{n} \Delta_i$. Ce critère permet de comparer aisément la qualité de la prédiction avec une méthode, "$RR_{=}$", pour laquelle l'intervalle RR est supposé constant et où $\forall i,\ \hat{\Delta}_i = \bar{\Delta}$. Pour cette méthode, le NMSE est constant et égal à 1.

Ainsi, la méthode de prédiction idéale fera tendre le NMSE vers 0, qui suppose une prédiction parfaite. Un NMSE proche de 1 signifie que la méthode de prédiction est équivalente à considérer une variation du rythme cardiaque nulle et estimer le prochain intervalle RR par le RR moyen. Toute valeur de NMSE supérieure à 1 est à proscrire.

Trois jeux de paramètres différents pour les ordres du PMIR ont été testés. PMIR 1 rassemble les résultats pour lesquels $n_1 = n_2 = n_3 = n_4 = 2$, c'est-à-dire que l'ensemble des signaux respiratoires est utilisé. Le PMIR 2 est la méthode pour laquelle, les signaux d'amplitude des ondes R ne sont pas utilisés et $n_1 = n_2 = n_3 = 2$. Enfin, PMIR 3 est une méthode où seuls les signaux d'intervalle RR sont utilisés $n_2 = n_3 = n_4 = 0$ et $n_1 = 2$.

Les résultats obtenus sur les apnées en inspiration sont rassemblés dans le tableau 5.1.

Sujet N°=	PMIR 1	PMIR 2	PMIR 3	RR =
$S1$	0.11	0.09	0.08	1
$S2$	0.55	0.45	0.45	1
$S3$	0.09	0.08	0.07	1
$S4$	0.06	0.05	0.04	1
$S5$	0.35	0.29	0.25	1

TABLE 5.1 – Comparaison des résultats de NMSE obtenus par les méthodes PMIR pour les apnées en inspiration

L'amélioration de la qualité de prédiction est visible : les résultats du PMIR sont bien meilleurs que lorsque l'intervalle RR est supposé constant, ce qui prouve la nécessité de prendre en compte les variations du rythme cardiaque pendant une apnée.

L'utilisation des signaux respiratoires, avec les méthodes PMIR 2 et 3, se révèle également plus efficace que lorsque seuls les signaux d'intervalle RR sont considérés. Cela peut paraître étonnant du fait que la prédiction est mise en place en apnée, alors que par définition la respiration est nulle. Néanmoins, pendant leur apnée, les sujets ont des mouvements involontaires de l'abdomen pour retenir leur respiration. Ces mouvements, qui modifient le volume et/ou la pression abdominale, peuvent influer sur la variation du rythme cardiaque. L'intégration des signaux de ceinture, qui mesurent ces mouvements, permet un apport d'information utile à la prédiction.

Les méthodes PMIR 2 et 3 permettent d'obtenir une qualité de prédiction sensiblement similaire, avec une légère amélioration pour le PMIR 3. L'utilisation des signaux d'amplitude de l'onde R, qui s'obtiennent difficilement sur des signaux ECG bruités par l'environnement IRM, n'apporte donc pas d'améliorations majeures de prédiction.

Les résultats obtenus sur les apnées en expiration sont rassemblés dans le tableau 5.2.

Sujet N°=	PMIR 1	PMIR 2	PMIR 3	RR =
$S1$	0.2	0.15	0.24	1
$S2$	1.22	0.88	0.91	1
$S3$	0.53	0.44	0.45	1
$S4$	0.39	0.29	0.3	1
$S5$	0.03	0.02	0.02	1

TABLE 5.2 – Comparaison des résultats de NMSE obtenus par les méthodes PMIR pour les apnées en expiration

Les résultats des méthodes PMIR sont également meilleurs que lorsque l'intervalle RR est supposé constant. Cette amélioration est néanmoins moins marquée que pour les apnées en inspiration, ce qui

tend à montrer que les variations de l'intervalle RR sont plus chaotiques et moins prévisibles en position de fin d'expiration plutôt que d'inspiration.

Les résultats de PMIR 2 et 3 sont encore meilleurs que PMIR 1, ce qui montre à nouveau l'utilité de l'acquisition de signaux respiratoires pour prédire les variations de l'intervalle RR pendant une apnée. Les méthodes PMIR 2 et 3 ont à nouveau des résultats équivalents, avec cette fois un léger avantage à la méthode PMIR 2.

L'efficacité des méthodes PMIR, en particulier PMIR 2, ayant été démontrée, il est possible de les implanter dans un système de synchronisation IRM temps-réel, afin de permettre l'acquisition en sang noir durant la systole. Une nouvelle étude a alors été menée pour acquérir des images en systole en sang noir sur cinq sujets. La qualité de l'image obtenue est telle qu'un diagnostic est possible et la contraction du cœur montre que cette acquisition a bien été faite en systole (cf. figures 5.9, 5.10) et 5.11.

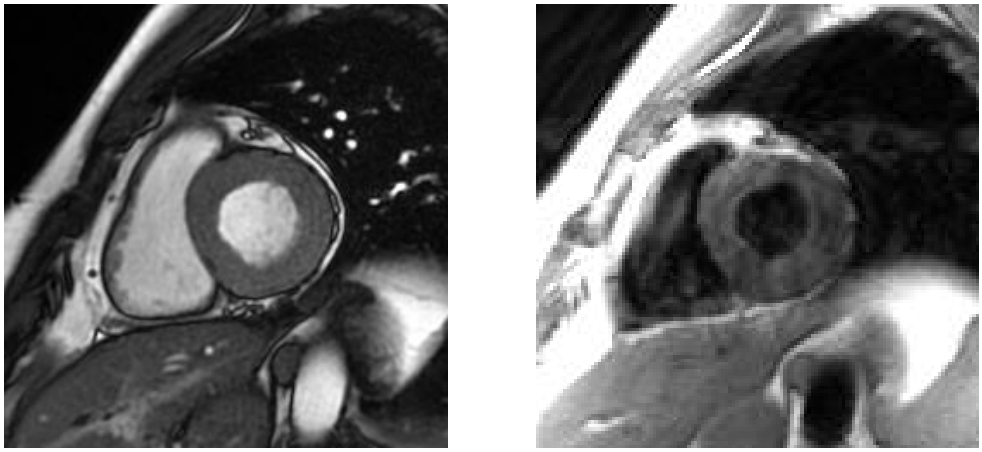


FIGURE 5.9 – Comparaison d'une acquisition en sang noir en systole (à gauche) avec une image CINE en systole (à droite)

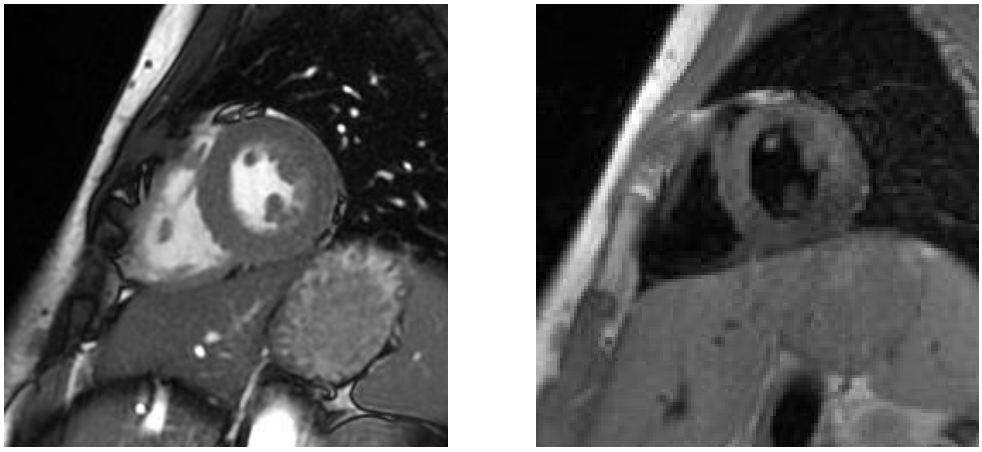


FIGURE 5.10 – Comparaison d'une acquisition en sang noir en systole (à gauche) avec une image CINE en systole (à droite)

Discussion

Une méthode de prédiction des intervalles RR (soit du rythme cardiaque) a été présentée afin de résoudre le problème de synchronisation de l'acquisition IRM avec les phases cardiaques. Celle-ci est

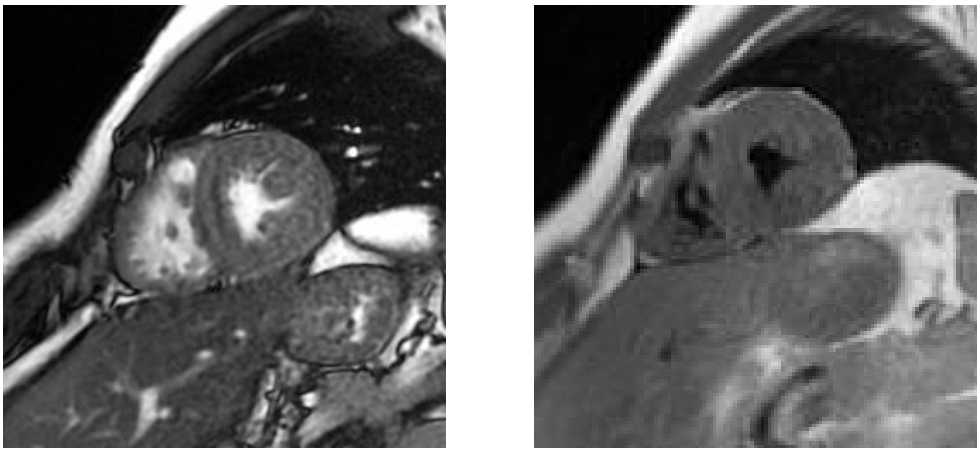


FIGURE 5.11 – Comparaison d'une acquisition en sang noir en systole (à gauche) avec une image CINE en systole (à droite)

basée sur des hypothèses, simples mais réalistes, de l'existence d'un modèle linéaire pour la prédiction de l'intervalle RR. L'utilisation de ce modèle avec une approche par filtrage de Kalman permet d'ailleurs de prendre en compte l'incertitude du modèle et le fait que celui-ci n'est qu'une modélisation partielle de la réalité. De plus, le filtrage de Kalman permet de prédire la variation du rythme cardiaque en ligne et donc de s'adapter parfaitement aux différents sujets, qui ont chacun une variabilité propre.

La méthode qui a été présentée n'en est pas moins performante et permet d'obtenir une prédiction de l'intervalle avec une erreur moyenne suffisamment faible pour permettre une acquisition non ou faiblement artefactée. Le choix des paramètres du modèle, $n_1 = n_2 = n_3 = 2$, et du filtrage de Kalman a été dicté par la nécessité d'être fortement réactif. En effet, la durée des apnées nécessaire pour une acquisition d'image est de l'ordre de 20 secondes soit environ une trentaine d'intervalles RR. Le choix de paramètres n_1, n_2, n_3, supérieurs à deux ne permettrait pas d'avoir une aussi grande réactivité et induirait un allongement non désirable de l'apnée du sujet.

Les paramètres utilisés par la méthode PMIR 2 ont induit une erreur moyenne relativement faible. Néanmoins, en raison de la forte réactivité, lors de fortes variations certains points peuvent être mal prédits. Les images réalisées après ce type de synchronisation peuvent alors être artefactées. Une méthode de reconstruction des images IRM, suite à la suppression des lignes acquises dans une mauvaise phase cardiaque, permet de réduire les artefacts induits par la mauvaise synchronisation [151].

Plusieurs études ont étendu l'utilisation de cette méthode pour réaliser également des acquisitions en sang noir et en systole. Dans [55, 58] nous suggérons de combiner la méthode de prédiction avec un modèle de cycle cardiaque. Ce dernier permet de déterminer la longueur de la systole, T_s, et de la diastole, T_d, en fonction de la durée de l'intervalle RR :

$$\begin{aligned} T_s &= 0,546 - 0,0021 \times \frac{60}{\Delta} \\ T_d &= \Delta - T_s \end{aligned} \tag{5.14}$$

L'utilisation de ce modèle cardiaque permet de déterminer le placement de la fenêtre d'acquisition, avec le délai T_D, de manière optimale et adaptative.

De même, dans [131] nous suggérons d'acquérir les images en sang noir et en systole alors que le patient respire librement. Ces acquisitions sont rendues possible par une méthode de reconstruction basée sur l'utilisation de signaux physiologiques [151].

De nombreuses applications nouvelles sont également envisageables pour cette méthode de prédiction. Dans le cadre de l'imagerie IRM, l'ensemble des applications cardiaques pourrait en bénéficier. La combinaison de la méthode de prédiction et d'un modèle cardiaque (équation (5.14)) pourrait permettre un placement optimal des fenêtres d'acquisition comme nous l'avons montré dans [55, 58].

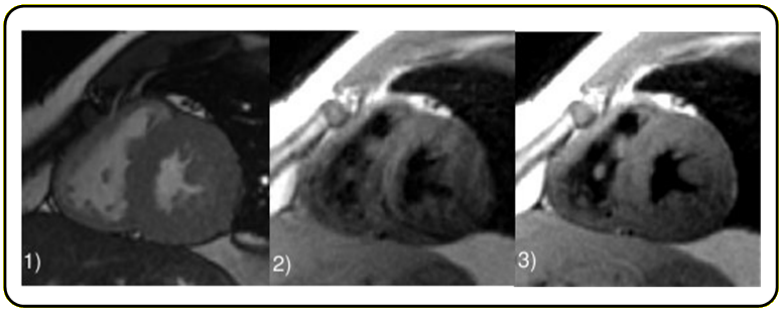


FIGURE 5.12 – Exemple d'acquisition d'images en sang noir en systole (à gauche : CINE, milieu : acquisition avec supposition d'un RR constant et à droite : acquisition adaptative) [55]

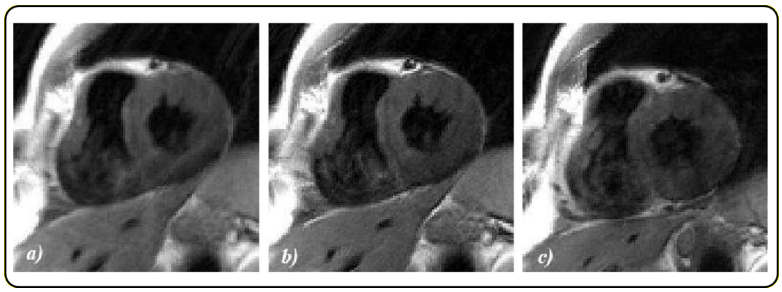


FIGURE 5.13 – Exemple d'acquisition d'images en sang noir en systole (à gauche : respiration libre et reconstruction fourier moyennée, milieu : respiration libre et reconstruction adaptative et à droite : acquisition en apnée) [131]

De plus, comme cela sera évoqué dans la section suivante, cette méthode de prédiction de l'intervalle RR pourrait servir à l'attribution du signal de phase cardiaque pour permettre un débruitage du signal ECG acquis en IRM avec un filtrage bayésien réalisé en ligne.

Enfin, une telle méthode de prédiction de l'intervalle RR pourrait servir à apporter une information sur la localisation du prochain complexe QRS. Cette information pourrait être intégrée dans un détecteur QRS afin de pouvoir augmenter la prédictivité positive de celui-ci. En effet, si une détection est réalisée en dehors d'un intervalle de confiance déterminé par l'estimation de l'intervalle RR, il pourra être supposé être soit une extrasystole soit une détection d'un artefact.

5.3 Débruitage de signaux ECG acquis en IRM

5.3.1 Position du problème

Nous présentons ici les particularités, en termes de *bruits*, du signal ECG acquis en IRM de manière à proposer une modélisation permettant un traitement optimal. Ainsi, une description (simplifiée) de la physique des phénomènes que nous cherchons à analyser est tout d'abord fournie de manière à réaliser à nouveau un effort de modélisation.

Une particularité physique de l'environnement IRM est la présence d'un champ magnétique variable à basse fréquence ($1 - 30kHz$). En effet, le champ magnétique est modulé spatialement afin que les particules d'eau émettent un signal en fonction de leur position. Cette modulation du champ magnétique est appelée gradient ("de champ magnétique"). Il existe trois gradients, gradient en x, y et z, selon la direction de modulation du champ magnétique. Idéalement, cette modulation de champ magnétique est une fonction linéaire de la direction et n'induit qu'une variation du champ dans l'axe du champ magnétique statique, $\vec{B}_0$, *i.e.* $\vec{u}_z$, telle que la valeur du champ magnétique au point (x_0, y_0, z_0) soit :

$$\vec{B}(x_0, y_0, z_0) = (B_0 + G_x x_0 + G_y y_0 + G_z z_0)\,\vec{z}, \tag{5.15}$$

lors de l'instauration de gradients de valeur G_x, G_y et G_z en $T.m^{-1}$. Afin d'avoir une acquisition d'image la plus courte possible, l'instauration de ces gradients doit être très rapide. Des vitesses de balayage allant jusqu'à $200T.m^{-1}.s^{-1}$ sont disponibles sur les imageurs du commerce. De plus, afin d'améliorer la qualité de l'image, l'amplitude de ces gradients est également sans cesse augmentée, pouvant ainsi atteindre $50mT.m^{-1}$.

La présence d'un champ magnétique variable induit la création d'un champ électrique dans le corps. De par leur contenu fréquentiel, ces champs induits peuvent provoquer une réponse physiologique, une stimulation nerveuse [233, 226], voire une influence plus problématique sur l'activité cardiaque [129, 141]. La Stimulation Nerveuse Périphérique (SNP) ou *Peripheral Nerve Stimulation* (PNS) a été l'une des considérations de sécurité majeure de ces dernières années [233, 226]. En tout état de cause, des champs électriques induits en surface de la peau du patient vont gêner l'acquisition du signal ECG.

Afin de réduire au maximum l'acquisition de ce signal indésirable pour le traitement de l'ECG, il a été proposé de réduire la distance entre les électrodes et donc la distance de mesure [52, 254], avec un tressage des câbles [1, 254]. Ceci induit une variabilité inter-patient importante. De plus, la bande passante du signal a également été réduite et est actuellement de l'ordre de $[1 - 20Hz]$ pour les capteurs disponibles dans le commerce [229].

L'acquisition d'un signal ECG reste néanmoins fortement perturbée par l'environnement IRM. Outre l'effet des gradients, on distingue aussi l'effet *MagnetoHydroDynamique* (MHD). En effet, on trouve aussi dans l'environnement IRM la présence d'un champ magnétique statique intense de l'ordre de 1.5 à 3 T pour les appareils cliniques actuels. Ceci représente 30 000 à 60 000 fois le champ magnétique terrestre en France ($47\mu T$). La présence de ce champ affecte la qualité du signal ECG principalement du fait de l'effet MHD, également appelé effet Hall, qui est la conséquence de l'interaction entre le champ magnétique statique et un fluide conducteur en mouvement, en l'occurrence principalement le sang circulant dans la crosse aortique, qui donne lieu à la création d'une tension. La mesure de cette tension comme mesure du flux sanguin a d'ailleurs été proposée, bien avant l'utilisation de l'IRM en médecine [114]. La figure 5.14 permet d'observer l'influence de ces perturbations sur un signal ECG.

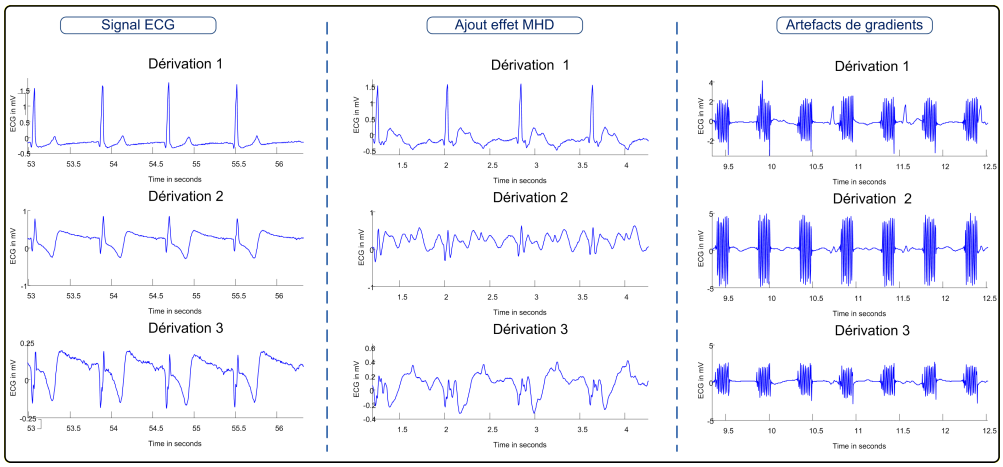


FIGURE 5.14 – Influence du champ statique et des gradients sur trois dérivations d'un signal ECG.

La mesure des gradients (ou au moins des instants d'émission de ces gradients) permet de débruiter de manière efficace les signaux ECG acquis en IRM [53]. On parle alors de méthode de débruitage avec information de gradient. Une localisation précise des artefacts est en outre possible et des techniques diverses, dont une approche adaptative, permettent de déterminer la réponse impulsionnelle d'un gradient sur l'ECG [53]. Néanmoins, toutes ces méthodes souffrent d'un même inconvénient : elles ne prennent pas en compte le signal ECG lors de l'estimation des réponses impulsionnelles. Elles supposent que le signal mesuré est constitué d'un bruit blanc sur lequel se superposent les artefacts et qu'un gradient est appliqué uniquement pendant la phase de repos électrique du cœur, en diastole. Or cette hypothèse est erronée pour de nombreuses applications, synchronisation rétrospective par exemple. De plus en IRM, même lors de la phase de repos électrique, un signal du à l'effet MHD est présent sur l'ECG. Afin d'améliorer la qualité

de débruitage des méthodes avec information de gradient, l'ensemble des contributions qui caractérisent l'ECG acquis en IRM doit être pris en compte lors de l'estimation de la réponse impulsionnelle. Ce fut l'objet d'une partie de la thèse de Julien OSTER [154, 164]

5.3.2 Théorie

Dans [142], les auteurs ont présenté une méthode permettant de synthétiser des signaux ECG réalistes. Leur approche avait été initialement proposée pour déterminer la qualité d'une méthode de débruitage, puisqu'en synthétisant un signal et en y rajoutant artificiellement un bruit, le rapport signal sur bruit est facilement calculé.

Ce modèle d'ECG repose sur la propriété de pseudo-périodicité d'un signal ECG normal. Dans le cadre de battements cardiaques normaux, une pseudo-période ou cycle cardiaque peut être considérée comme la succession de différentes déflexions, selon un ordre déterminé : ondes P, Q, R, S et T. Ce signal cyclique peut être modélisé par une somme de gaussiennes (figure 5.15). Le nombre de gaussiennes nécessaires à une bonne modélisation varie de cinq à six gaussiennes, selon que la non symétrie de l'onde T est prise en compte ou non.

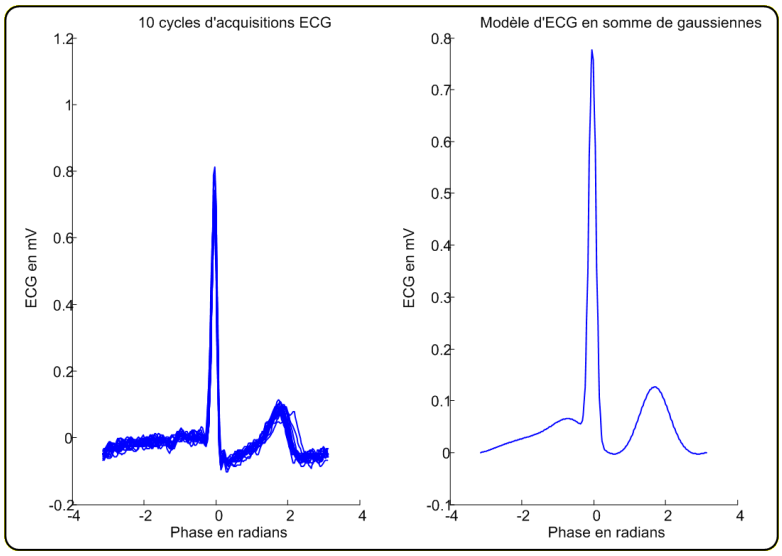


FIGURE 5.15 – Modélisation d'un cycle cardiaque par une somme de cinq gaussiennes

Afin de tenir compte de la pseudo-périodicité, ce signal est supposé avoir une trajectoire circulaire dans un plan (Oxy), dont l'angle représente la phase cardiaque, avec une amplitude dans la direction z qui représente la valeur en mV du potentiel électrique mesuré. Le cercle décrit par le modèle ECG est de rayon 1 en unité arbitraire et la vitesse de rotation est supposée constante tout au long d'un cycle donné, c'est-à-dire $\omega = 2\pi/RR$, où RR est la durée de l'intervalle RR.

De cette manière, les équations dynamiques de ce modèle ECG peuvent s'écrire en coordonnées polaires :

$$\begin{cases} \Theta_k = (\Theta_{k-1} + \omega\delta) \mod 2\pi \\ z_k = -\sum_i \delta \frac{\alpha_i \omega}{b_i^2} \Delta\Theta_{i,k-1} \exp\left(-\frac{\Delta\Theta_{i,k-1}^2}{2b_i^2}\right) + z_{k-1} + \eta \end{cases}, \tag{5.16}$$

où Θ_k est l'angle dans le plan Oxy à l'instant k, δ est la période d'échantillonnage. z_k représente une estimée de la valeur en mV de l'ECG à l'instant k. α_i, b_i et ξ_i représentent respectivement l'amplitude, la variance et la position angulaire de la $i^{\text{ième}}$ gaussienne, ce qui permet de définir

$$\Delta\Theta_{i,k-1} = (\Theta_{k-1} - \xi_i)(\mod 2\pi).$$

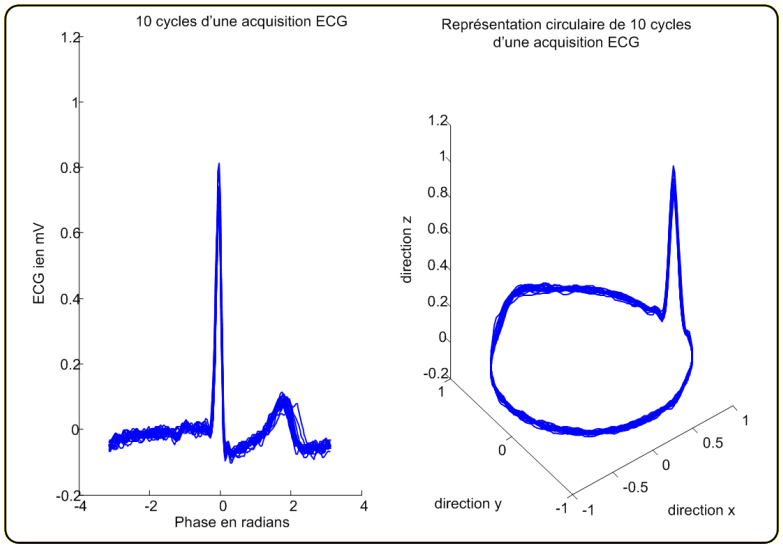


FIGURE 5.16 – Représentation circulaire de 10 cycles d'une acquisition ECG.

Ce modèle ECG a très vite trouvé d'autres applications, comme le débruitage et la segmentation du signal ECG. Il est ainsi démontré dans [40, 38] qu'en approximant les signaux réels par une somme de plusieurs gaussiennes, un débruitage et une segmentation précis peuvent être atteints. Leur approche d'ajustement par minimisation des moindres carrés ne permet cependant pas de réaliser les différentes tâches en ligne et nécessite une application en post-traitement. Dans le cadre du débruitage, le jeu de paramètres qui est le mieux ajusté aux données doit être déterminé. Pour une application optimale, cette détermination doit être réalisée en ligne, avec de surcroît une adaptation automatique au cours du temps, afin de prendre en compte la non-stationnarité du signal ECG.

Par ailleurs, l'utilisation d'un tel modèle, dont la dynamique est gouvernée par un jeu d'équations, est particulièrement bien adaptée au filtrage bayésien. C'est en mettant en forme les problèmes de débruitage, segmentation et compression, comme un problème de filtrage bayésien que de nombreuses solutions ont ainsi été proposées [223, 224, 222, 225].

La mise en place du débruitage de l'ECG dans un formalisme espace d'état est une excellente illustration. Pour ce faire, la méthode présentée par [225] sera utilisée. Les auteurs ont utilisé le modèle ECG en optant pour l'approximation du signal par cinq gaussiennes et ont défini les vecteurs d'évolution et d'observation de la manière suivante :

$$\begin{cases} \theta_k = [\Theta_k, z_k, \alpha_{i,k}, b_{i,k}, \xi_{i,k}]^t \quad i = (1..5) \\ y_k = [\varphi_k, s_k]^t \end{cases}, \tag{5.17}$$

où s_k représente le $k^{\text{ième}}$ échantillon du signal ECG mesuré et φ_k est le $k^{\text{ième}}$ échantillon d'un signal de phase, construit à partir de la connaissance des occurrences d'une onde R. Ce signal de phase a été assigné de manière linéaire entre deux ondes R. Cette phase est une droite de pente $2\pi/RR$, dont l'ordonnée à l'origine est 0. De plus, ce signal de phase est défini à 2π près et varie entre $-\pi$ et π (figure 5.17).

Les bruits d'évolution et d'observation ont été définis par :

$$\begin{cases} w_k = [\omega, \eta, \varepsilon_{\alpha,i}, \varepsilon_{b,i}, \varepsilon_{\xi,i}]^t \quad i = (1..5) \\ v_k = [v_1, v_2]^t \end{cases}. \tag{5.18}$$

La détermination des différentes fonctions est alors possible. Les deux premières composantes du vecteur d'état, z_k et Θ_k, suivent une évolution qui est gouvernée par les équations (5.16) du modèle d'ECG. Les quinze paramètres des cinq gaussiennes sont considérés comme des variables aléatoires suivant une marche aléatoire, puisqu'aucune information *a priori* sur la non stationnarité de l'ECG n'est

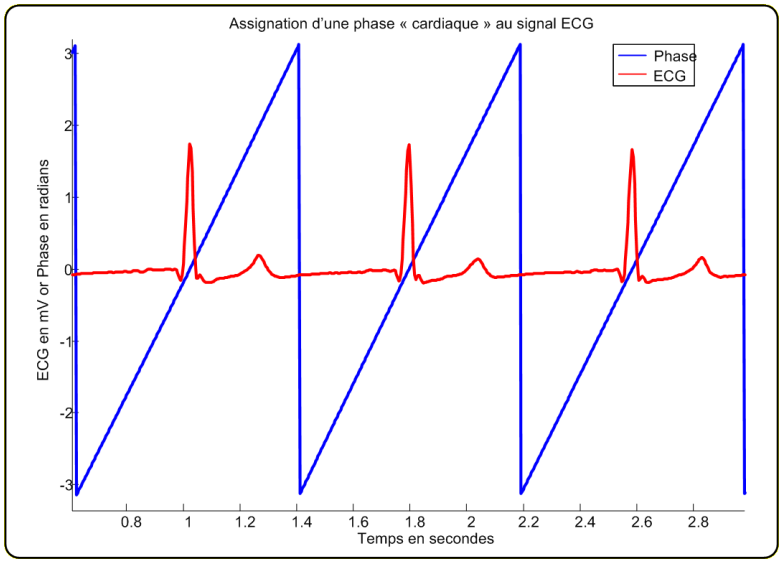


FIGURE 5.17 – Description de la procédure de création du signal de phase.

disponible. Ceci permet de suivre la solution et d'éviter de tomber dans un minimum local. Les équations d'évolution peuvent alors être mises sous la forme suivante :

$$\begin{cases} \Theta_k = (\Theta_{k-1} + \omega\delta) \mod 2\pi \\ z_k = -\sum_i \delta \frac{\alpha_i \omega}{b_i^2} \Delta\Theta_{i,k-1} \exp\left(-\frac{\Delta\Theta_{i,k-1}^2}{2b_i^2}\right) + z_{k-1} + \eta \\ \alpha_{i,k} = \alpha_{i,k-1} + \varepsilon_{\alpha,i} \\ b_{i,k} = b_{i,k-1} + \varepsilon_{b,i} \\ \xi_{i,k} = \xi_{i,k-1} + \varepsilon_{\xi,i} \end{cases}. \tag{5.19}$$

Enfin, les équations d'observation peuvent s'écrire :

$$\begin{cases} \varphi_k = \Theta_k + v_{1,k} \\ s_k = z_k + v_{2,k} \end{cases}. \tag{5.20}$$

Notons que les équations de l'espace d'état ne sont pas linéaires. Pour cette application précise, un filtrage de Kalman étendu (voir Section 5.1.2) a été utilisé.

Comme présenté précédemment, les gradients sont la cause des principales perturbations du signal ECG acquis en IRM. Leur présence empêche pratiquement toute analyse du signal et nécessite de mettre en place des méthodes de suppression spécifiques. Nous avons développé une méthode de séparation aveugle des sources (SAS) [159, 161, 162] qui permet d'obtenir des résultats encourageants, néanmoins la connaissance des signaux de commande de gradients reste un atout majeur.

En effet, le signal ECG acquis en IRM peut être modélisé comme un mélange linéaire convolutif bruité de trois sources, correspondant aux signaux de commande des trois gradients. Cette modélisation a été proposée par [53] et a été utilisée à plusieurs reprises dans le cadre de la suppression des artefacts de gradients [152, 2, 1, 3, 151]. Ceci peut se retranscrire mathématiquement de la manière suivante :

$$s_k = \sum_{i\in\{X,Y,Z\}} \left(h^i \otimes g_i\right) + \eta_k = \sum_{i\in\{X,Y,Z\}} \left(\sum_{l\in\mathbb{N}} h_l^i g_{i,(k-l)}\right) + \eta_k, \tag{5.21}$$

où s_k est le $k^{\text{ième}}$ échantillon du signal ECG enregistré, $g_{i,k}$ le $k^{\text{ième}}$ échantillon du signal de commande des gradients dans la direction i, η_k est la $k^{\text{ième}}$ échantillon de bruit et h_l^i est le $l^{\text{ième}}$ élément du filtre de création d'artefacts de gradient dans la direction i.

Ces filtres permettent de retranscrire l'ensemble des phénomènes physiques qui suivent l'instauration de courant dans les bobines de gradient, ce qui est un panel large allant de la variation du champ magnétique, en passant par l'induction de champs électriques dans le corps, par la mesure du potentiel aux électrodes jusqu'au filtrage de ce potentiel par l'électronique du capteur.

L'ensemble des méthodes de débruitage des artefacts utilisant l'information des gradients est basé sur une estimation des éléments de ces filtres, supposés être à réponse impulsionnelle finie (RIF). Ainsi, en considérant que la longueur du filtre RIF est N, l'équation 5.21 se réécrit :

$$s_k = \sum_{i\in\{X,Y,Z\}} \left(\sum_{l=0}^{N-1} h_l^i g_{i,(k-l)} \right) + \eta_k, \tag{5.22}$$

et il y a donc $3N$ éléments à estimer pour chaque voie ECG.

Il est possible de reformuler ce problème d'estimation des réponses impulsionnelles comme un problème de filtrage bayésien. En supposant que les filtres $\underline{h}^i = [h_0^i \dots h_{N-1}^i]^T$ sont des vecteurs aléatoires de taille N suivant une marche aléatoire, le vecteur état $\underline{x}$ s'écrit :

$$\underline{x} = \begin{cases} \underline{h}^X = [h_0^X \dots h_{N-1}^X]^T \\ \underline{h}^Y = [h_0^Y \dots h_{N-1}^Y]^T \\ \underline{h}^Z = [h_0^Z \dots h_{N-1}^Z]^T \end{cases}$$

L'équation d'évolution est alors $\underline{x}_k = \underline{x}_{k-1} + \underline{w}_{k-1}$.

Le vecteur d'observation est simplement défini comme le signal ECG enregistré, c'est-à-dire $y_k = s_k$ et l'équation d'évolution a été définie par l'équation (5.22), ce qui signifie que la matrice d'observation C_k est définie par :

$$C_k = [g_{X,k}, \dots, g_{X,k-N+1}, g_{Y,k}, \dots, g_{Y,k-N+1}, g_{Z,k}, \dots, g_{Z,k-N+1}].$$

Le problème d'estimation de la réponse impulsionnelle peut alors se mettre sous la forme :

$$\begin{cases} \theta_k = \theta_{k-1} + w_{k-1} \\ y_k = C_k \theta_k + \eta_k \end{cases} \tag{5.23}$$

Il est donc envisageable de résoudre le problème de suppression des artefacts de gradient par le biais d'un filtre de Kalman étendu. Cependant, cette méthode souffrirait des mêmes limitations que l'ensemble des méthodes de débruitage utilisant l'information des gradients, c'est-à-dire que l'activité électrique du cœur n'est pas prise en compte lors de l'estimation des réponses impulsionnelles. En effet, l'ensemble des contributions, hors artefacts de gradient, du signal ECG acquis en IRM est contenu dans le bruit d'observation η_k.

5.3.3 Méthode

Une nouvelle méthode de débruitage des signaux ECG acquis en IRM peut donc être proposée en unifiant les deux approches présentées dans la sous-section précédente. D'une part, une approche de filtrage bayésien a été envisagée pour le débruitage de signaux ECG basée sur l'utilisation d'un modèle de mélange de gaussiennes. D'autre part, une approche bayésienne est concevable pour l'estimation des réponses impulsionnelles des artefacts de gradient. Les deux formulations peuvent donc être regroupées dans un même état-espace. L'objectif sous-jacent à l'unification de ces deux approches est alors de pouvoir prendre en compte le signal de l'ECG lors de l'estimation des artefacts de gradient. Bien que nous nous cantonnerons une fois encore à la mise en place d'un filtre de Kalman, nous appellerons (un peu abusivement) la méthode proposée *BAyesian Gradient ARtifact REduction* (BAGARRE) [164].

Le débruitage d'ECG basé sur un modèle de mélange de gaussiennes nécessite une procédure d'attribution de la phase cardiaque (voir Fig. 5.17). Ce signal est créé à partir des détections des ondes R. Or, si pour des acquisitions standard cette détection est une tâche aisément réalisable, il n'en est pas de même dans le cas d'ECG acquis en IRM. Nous avons donc mis au point une méthode de détection fiable

du complexe QRS en milieu IRM basée sur la théorie des ondelettes et l'analyse de la régularité Lipschitzienne du signal. Nous n'exposons pas cette méthode ici pour des raisons de place et de cohésion du texte, néanmoins des détails peuvent être trouvés dans [158, 154]. Cette méthode fournit les meilleures performances de l'état de l'art à notre connaissance.

Afin d'unifier les deux approches précédemment décrites, il est intéressant de partir de leurs équations d'observation respectives. Dans le cadre du débruitage de l'ECG, le signal s_k enregistré est considéré comme la somme de deux contributions : le signal ECG z_k et le bruit de mesure $v_{2,k}$ ($s_k = z_k + v_{2,k}$). Pour les acquisitions d'ECG en IRM, ce bruit d'observation regroupe l'effet Hall, le signal dû à la respiration et aux artefacts de gradient et les bruits de mesure du capteur. Dans le cadre de la réduction des artefacts de gradient, le signal enregistré est considéré comme la somme de trois signaux de gradient filtrés et d'un bruit de mesure, η_k ($s_k = \sum_{i\in\{X,Y,Z\}} \left(\sum_{l=0}^{N-1} h_l^i g_{i,(k-l)}\right) + \eta_k$). Ce bruit de mesure contient entre autres le signal ECG et l'effet MHD. Une approche plus réaliste consiste à considérer le signal s_k enregistré comme un signal ECG, z_k, une somme des trois signaux de commande des gradients filtrés et d'un bruit de mesure, ce qui conduit à l'équation suivante :

$$s_k = z_k + \sum_{i\in\{X,Y,Z\}} \left(\sum_{l=0}^{N-1} h_l^i g_{i,(k-l)}\right) + v_{2,k}. \tag{5.24}$$

Le bruit d'observation $v_{2,k}$ ne comporte plus alors que l'effet MHD, le signal dû à la respiration et le bruit de mesure du capteur ou des électrodes. Par ailleurs, l'intégration des paramètres des gaussiennes dans le vecteur d'état permet de prendre en compte la non stationnarité de l'ECG et de modéliser en partie l'effet MHD. Il est donc plus précis de considérer que le signal $v_{2,k}$ intègre alors les artefacts de mouvement et les bruits de mesure du capteur ou des électrodes.

En partant de cette équation d'évolution, il est possible de déterminer le vecteur d'état. Celui-ci peut être construit en concaténant simplement les vecteurs d'état présentés précédemment :

$$\theta_k = [\Theta_k, z_k, \alpha_{i,k}, b_{i,k}, \xi_{i,k}, h_{1,k}^j, \ldots, h_{N,k}^j]^T,$$

où Θ_k et z_k sont la position angulaire et l'amplitude du modèle ECG, $\alpha_{i,k}, b_{i,k}, \xi_{i,k}$ représentent respectivement l'amplitude, la variance et la position de la $i^{\text{ième}}$ gaussienne. Le vecteur $\underline{h}_k^j = [h_{1,k}^j, \ldots, h_{N,k}^j]^T$ représente la réponse impulsionnelle du gradient dans la direction $j \in \{X, Y, Z\}$.

L'évolution de ce vecteur d'état est régie par le même jeu d'équations (5.19). En effet, BAGARRE repose sur le modèle d'ECG, qui modélise le signal durant un cycle cardiaque par une somme de cinq gaussiennes, les équations (5.16) restent donc valables et régissent l'évolution des deux premiers éléments de θ_k. Aucune information sur la non stationnarité de l'ECG n'est disponible, ce qui conduit à considérer les paramètres des gaussiennes, $\alpha_{i,k}, b_{i,k}, \xi_{i,k}$, comme des variables suivant une marche aléatoire de paramètres $\varepsilon_{i,k}^{\alpha}, \varepsilon_{i,k}^{b}, \varepsilon_{i,k}^{\xi}$. De même, les vecteurs $\underline{h}_k^j = [h_{1,k}^j, \ldots, h_{N,k}^j]^T$ de réponses impulsionnelles sont supposés suivre une marche aléatoire de paramètre $\underline{\sigma}_k^j = [\sigma_{1,k}^j, \ldots, \sigma_{N,k}^j]^T$, puisqu'aucune information *a priori* sur leur évolution n'est disponible. L'équation d'évolution de ce système peut alors se mettre sous la forme :

$$\begin{cases} \Theta_k = \Theta_{k-1} + \omega\delta \quad (\mod 2\pi) \\ z_k = z_{k-1} - \sum_i \delta \frac{\alpha_i \omega}{b_i^2} \Delta\Theta_{i,k-1} \exp\left(-\frac{\Delta\Theta_{i,k-1}^2}{2b_i^2}\right) + \eta \\ \alpha_{i,k} = \alpha_{i,k-1} + \varepsilon_{i,k-1}^{\alpha} \\ b_{i,k} = b_{i,k-1} + \varepsilon_{i,k-1}^{b} \\ \xi_{i,k} = \xi_{i,k-1} + \varepsilon_{i,k-1}^{\xi} \\ \underline{h}_k^j = \underline{h}_{k-1}^j + \underline{\sigma}_{k-1}^j \end{cases}, \tag{5.25}$$

où $\omega = 2\pi/RR$ représente la vitesse angulaire du modèle, δ la période d'échantillonage du signal ECG et η l'incertitude sur la dynamique du modèle d'ECG. Le bruit d'évolution associé est alors

$$w_k = \left[\omega, \eta, \varepsilon_{i,k}^{\alpha}, \varepsilon_{i,k}^{b}, \varepsilon_{i,k}^{\xi}, \sigma_{1,k}^j, \cdots, \sigma_{N,k}^j\right]^T$$

Afin de mettre en place le filtrage de Kalman, la procédure de création de signal de phase cardiaque, φ_k, présentée dans la figure 5.17 est utilisée. Le vecteur d'observation, $\underline{y}_k = [\varphi_k, s_k]^T$ peut alors être mesuré et les deux équations d'observation de ce système sont :

$$\begin{cases} \varphi_k = \Theta_k + v_{1,k} \\ s_k = z_k + \sum\limits_{j\in\{X,Y,Z\}} \left(\underline{h}_k^j\right)^T \cdot \underline{g}_{j,k} + v_{2,k} \end{cases}, \tag{5.26}$$

où le vecteur $\underline{g}_{j,k} = [g_{j,k}, \ldots, g_{j,k-N+1}]$ représente un horizon de N échantillons du signal de commande des gradients dans la direction $j \in \{X, Y, Z\}$.

La longueur des réponses impulsionnelles a été fixée à $44ms$, ce qui correspond à 11 échantillons pour une fréquence d'échantillonnage de $250Hz$. La taille du vecteur d'état est donc $50 = 2 + 3 \times 5 + 11 \times 3$. Ce choix est un bon compromis entre une taille de vecteur d'état minimale, pour limiter la complexité calculatoire, et une longueur de réponse impulsionnelle suffisamment grande pour décrire le processus physique à l'origine des artefacts de gradient.

Les méthodes de filtrage bayésien en générale sont sensibles à l'initialisation des paramètres. Cette étape doit donc être réalisée avec soin, tout en étant automatique afin de ne pas perdre de vue l'application industrielle (rappelons que ce travail a été réalisé dans le cadre de la thèse de Julien OSTER [154] réalisée en collaboration avec Schiller Medical). Cette phase de calibration peut être réalisée pendant l'installation du patient. Des enregistrements d'ECG sont alors réalisés en dehors de l'aimant IRM et peuvent remplir cette tâche. Ainsi, les trente premiers cycles cardiaques sont utilisés pour l'initialisation des paramètres, $\alpha_{i,0}, b_{i,0}, \xi_{i,0}$ des cinq gaussiennes, mais aussi des éléments diagonaux de la matrice R_0 correspondant à ces paramètres. Cette procédure a été réalisée en approximant les trente premiers cycles avec une méthode d'optimisation non linéaire des moindres carrés [164]. Les vecteurs $\underline{h}_0^j$ ont été initialisés à $\underline{0}$.

Comme les fonctions f et g sont non-linéaires mais dérivables, le choix de la technique de filtrage s'est tourné vers le filtre de Kalman étendu. Les fonctions gaussiennes sont de plus assez douces pour que leur linéarisation locale en utilisant un développement de Taylor du première ordre soit une approximation réaliste.

Afin de pouvoir implémenter le filtre de Kalman étendu, il est nécessaire de calculer la dérivée des fonctions d'évolution et d'observation.

Tout d'abord, les équations d'observation sont déjà linéaires et sont définies par :

$$\begin{cases} C_k(i,i) = 1 \quad \forall i \\ C_k(2, 18..28) = \dfrac{\partial g_2(\theta, v, k)}{\partial h_{l,k}^X} = g_{X,k-l+1} \\ C_k(2, 29..39) = \dfrac{\partial g_2(\theta, v, k)}{\partial h_{l,k}^Y} = g_{Y,k-l+1} \\ C_k(2, 40..50) = \dfrac{\partial g_2(\theta, v, k)}{\partial h_{l,k}^Z} = g_{Z,k-l+1} \\ C_k(i,j) = 0 \quad \forall i \neq j \end{cases} \tag{5.27}$$

et

$$G_k = \frac{\partial g(\theta, v, k)}{\partial v} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.28}$$

L'équation d'évolution peut quant à elle être linéarisée avec les deux matrices A_k et F_k définies comme suit :

$$
\left\{
\begin{aligned}
A_k(1,1) &= \frac{\partial f_1(\theta,w,k)}{\partial \Theta_k} = 1 \\
A_k(2,1) &= \frac{\partial f_2(\theta,w,k)}{\partial \Theta_k} \\
&= -\sum_i \delta \frac{\alpha_i \omega}{b_i^2}\left(1-\frac{\Delta\Theta_{i,k}^2}{2b_i^2}\right)\exp(-\frac{\Delta\Theta_{i,k}^2}{2b_i^2}) \\
A_k(2,2) &= \frac{\partial f_2(\theta,w,k)}{\partial \eta} = 1 \\
A_k(2,3..7) &= \frac{\partial f_2(\theta,w,k)}{\partial \alpha_{i,k}} \\
&= -\delta \frac{\omega \Delta\Theta_{i,k}}{b_i^2}\exp(-\frac{\Delta\Theta_{i,k}^2}{2b_i^2}) \\
A_k(2,8..12) &= \frac{\partial f_2(\theta,w,k)}{\partial b_{i,k}} \\
&= -\delta \frac{\alpha_i \omega \Delta\Theta_{i,k}}{b_i^3}\left(1-\frac{\Delta\Theta_{i,k}^2}{b_i^{\S}}\right)\exp(-\frac{\Delta\Theta_{i,k}^2}{2b_i^2}) \\
A_k(2,13..17) &= \frac{\partial f_2(\theta,w,k)}{\partial \xi_{i,k}} \\
&= \delta \frac{\alpha_i \omega}{b_i^2}\left(1-\frac{\Delta\Theta_{i,k}^2}{b_i^2}\right)\exp(-\frac{\Delta\Theta_{i,k}^2}{b_i^2})
\end{aligned}
\right. \tag{5.29}
$$

et

$$
\left\{
\begin{aligned}
&F_k(1,1) = \frac{\partial f_1(\theta,w,k)}{\partial \omega} = \delta \\
&F_k(i,i) = 1 \quad \text{for } i \neq 1 \\
&F_k(2,1) = \frac{\partial f_2(\theta,w,k)}{\partial \omega} = -\sum_i \delta \frac{\alpha_i}{b_i^2}\Delta\Theta_{i,k}\exp\left(-\frac{\Delta\Theta_{i,k}^2}{2b_i^2}\right) \\
&F_k(i,j) = 0 \quad \text{pour tous les autres couples } i \neq j
\end{aligned}
\right. . \tag{5.30}
$$

Il existe deux variantes à la méthode BAGARRE selon que son application soit tournée vers le monitorage ou la synchronisation. Elles seront appelées respectivement BAGARRE-M (pour monitorage) et BAGARRE-T (pour triggering). La nécessité de développer deux variantes de cette méthode est dictée par les contraintes temporelles de chacune de ces applications. La création du signal de phase φ_k nécessite en effet de connaître la position des ondes R et induit donc un retard sur la mise en place du filtrage d'au moins un intervalle RR.

BAGARRE-M peut être appliquée comme cela a été présenté pour les méthodes bayésiennes de débruitage ECG [223, 224, 222, 225], les délais induits n'empêchant pas un monitorage efficace. BAGARRE-M comporte alors trois phases différentes.

- les complexes QRS sont détectés par la méthode présentée dans [158]. Ce détecteur est appliqué sur des fenêtres de deux secondes, et qui glissent d'une demie seconde.
- A chaque nouvelle détection QRS, le signal de phase φ_k est créé sur l'intervalle RR le plus récent.
- Le filtre de Kalman étendu peut alors être appliqué sur cette période du signal et l'estimée z_k du signal ECG est mise à jour et constitue le signal ECG débruité.

BAGARRE-T, qui se veut être une méthode servant à la synchronisation, ne peut pas se permettre un tel retard. Le débruitage du signal ECG n'est donc pas basé sur l'estimée z_k. Lors du processus du filtrage de Kalman étendu, les réponses impulsionnelles, $\underline{h}_k^j$, sont également estimées et vont servir au débruitage du signal ECG.

Supposons $n-k$ le délai maximum induit par BAGARRE-M pour le débruitage. Il est alors possible de débruiter le signal ECG au temps n avec les estimées des réponses impulsionnelles estimées au temps k, en utilisant l'équation (5.24) :

$$\hat{z}_n = s_n - \sum_{j\in\{X,Y,Z\}} \left(\underline{h}_k^j\right)^T \cdot \underline{g}_{j,n}. \tag{5.31}$$

Ce débruitage n'induira alors aucun délai supplémentaire et est donc compatible avec la synchronisation. Cette approche présente des similitudes avec les techniques présentées par [53] et [152], à ceci près que l'estimation des réponses impulsionnelles est mise à jour régulièrement : BAGARRE-T est donc adaptatif (figure 5.18).

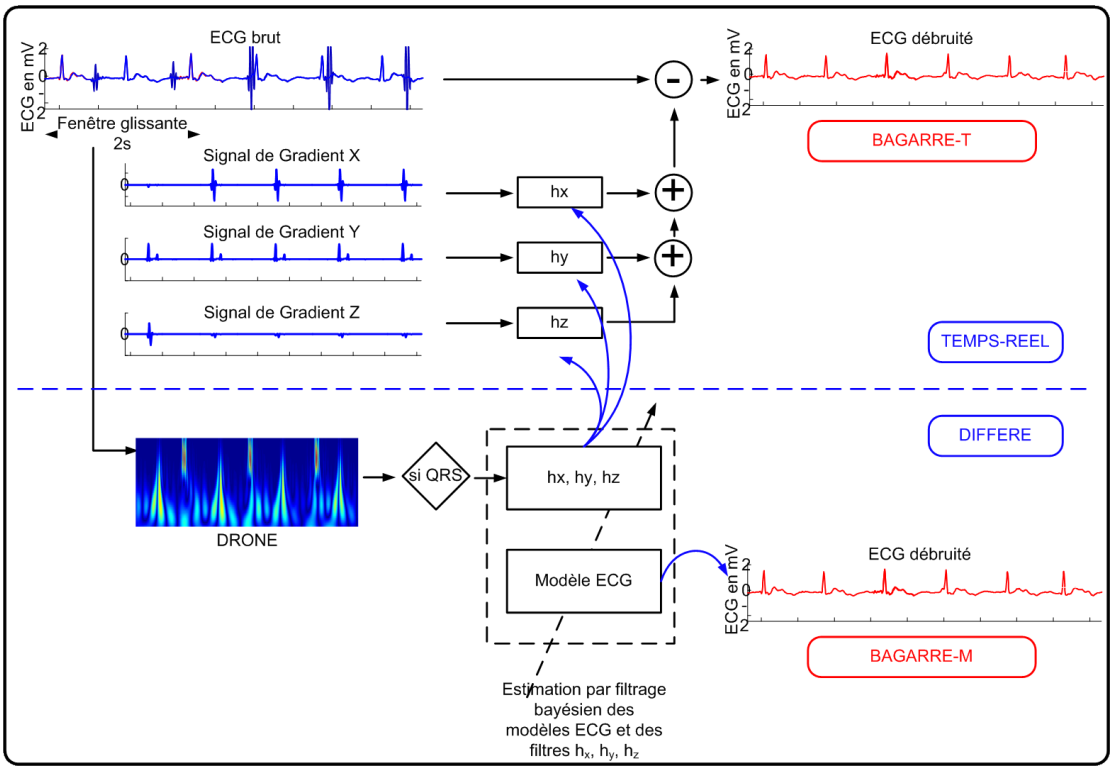


FIGURE 5.18 – Schéma explicatif des méthodes BAGARRE (en haut : débruitage temps-réel avec les estimées des réponses impulsionnelles (BAGARRE-T), en bas : estimation bayésienne des paramètres des modèles : réponses impulsionnelles pour BAGARRE-T et modèle ECG pour BAGARRE-M).

5.3.4 Résultats

L'évaluation des méthodes BAGARRE a été réalisée en quantifiant la qualité de la détection QRS. La détermination de la qualité de détection des battements cardiaques a déjà été standardisée [7]. Le cas particulier des signaux ECG acquis en IRM n'induit pas, *a priori*, de modifications dans l'évaluation de cette tâche. Nous nous sommes donc inspirés de cette norme pour calculer les paramètres de sensibilité (Se) et de prédictivité positive ($+P$) des méthodes BAGARRE.

$$\begin{cases} Se = \dfrac{TP}{TP+FN} \\ +P = \dfrac{TP}{TP+FP} \end{cases}, \tag{5.32}$$

où TP (“*True Positive*”) signifie vrai positif, FP (“*False Positive*”) signifie faux positif et FN (“*False Negative*”) signifie faux négatif. Ces différentes valeurs prenant les significations usuelles (adaptées en suivant la norme [7] dont les détails sont hors de propos ici).

Un algorithme implémenté dans un appareil de monitorage disponible dans le commerce, Argus (Schiller, Baar, Suisse) a donc été appliqué aux signaux produits par BAGARRE. Ces résultats ont été comparés avec des méthodes de l'état de l'art :

- les détections, sur le signal brut, issues d'une méthode industrielle présente dans un appareil de monitorage Argus (Schiller, Baar, Suisse), qui seront désignées par l'acronyme *Brut*.
- les détections sur les signaux débruités par la méthode *Least Mean Square* (LMS) [3] issues du même détecteur.
- les détections réalisées par la méthode VectoCardioGramme (VCG) [60].
- les détections issues du détecteur basé sur les ondelettes présenté dans [158] désignées par DRONE
- les détections sur les signaux issus de la méthode de débruitage présentée par [225], réalisées par le détecteur industriel présent dans l'appareil Argus (Schiller, Baar, Suisse), désignées par *Bayésien*.

Les résultats obtenus sur la base de données sont rassemblés dans le tableau 5.3.

	Base complète		Enregistrements peu perturbés		Enregistrements très perturbés	
Méthode	Se	$+P$	Se	$+P$	Se	$+P$
Brut	98.4	92.6	99.6	98.0	91.8	69.5
VCG	97.1	86.7	97.5	89.3	94.9	74.2
DRONE	99.0	98.8	99.4	99.6	96.8	94.1
LMS	99.5	97.6	99.7	99.2	98.3	89.6
Bayésien	98.8	95.3	99.6	98.8	93.9	78.9
BAGARRE-T	99.6	98.3	99.7	99.2	98.9	93.2
BAGARRE-M	99.7	99.2	99.8	99.6	99.4	97.0

TABLE 5.3 – Tableau comparatif de la qualité de la détection QRS.

La méthode *Bayésien* est efficace puisque la détection QRS est sensiblement améliorée sur la base complète par rapport à la méthode *Brut* ($+0.4$ points de sensibilité et $+2.7$ de prédictivité positive). Cependant, cette méthode n'est pas adaptée aux fortes perturbations de l'environnement IRM, les artefacts de gradient en particulier, c'est ce que montre la prédictivité positive sur les enregistrements très perturbés (78.9%), qui est inférieure de 10.7 points par rapport au LMS. Ceci justifie l'approche abordée par les méthodes BAGARRE, qui prennent en considération la spécificité des artefacts de gradient dans le débruitage.

BAGARRE-T obtient les meilleurs résultats sur l'ensemble des critères parmi les méthodes compatibles avec la synchronisation (Brut, VCG, LMS). Cette méthode surpasse le LMS sur les enregistrements très perturbés avec en particulier une amélioration de 3.6 points de prédictivité positive. Cela montre l'importance de la prise en compte du signal ECG dans l'estimation des réponses impulsionnelles.

Enfin, BAGARRE-M surclasse l'ensemble des méthodes présentées sur la totalité des critères de qualité. Les résultats obtenus sur les enregistrements très perturbés sont très bons avec une sensibilité à 99.4% et une prédictivité positive à 97.0%. Cette méthode permet donc d'obtenir une excellente qualité de monitorage même durant les périodes de forte perturbation de l'environnement IRM (figure 5.19).

Par ailleurs, les méthodes BAGARRE ont été également évaluées selon l'amélioration de la qualité du signal. Afin d'évaluer de la manière la plus réaliste possible la qualité de débruitage, il est nécessaire d'introduire un nouveau critère spécifique aux acquisitions ECG en IRM. Ainsi, il est important de rappeler que, pour ces signaux, le niveau de bruit est très élevé. L'effet MHD est un bruit présent dès lors que le sujet se trouve dans le champ magnétique statique. Ce bruit ne laisse alors ressortir que les battements cardiaques et déforme les ondes P et T, entre autres. L'évaluation de l'énergie du signal peut donc être restreinte à l'énergie des battements cardiaques (complexes QRS et autres extrasystoles). Ces instants sont annotés dans la base de données ECG IRM constituée (voir figure 5.20). Les annotations de la base de données ECG IRM informent aussi des périodes temporelles lors desquelles de tels artefacts

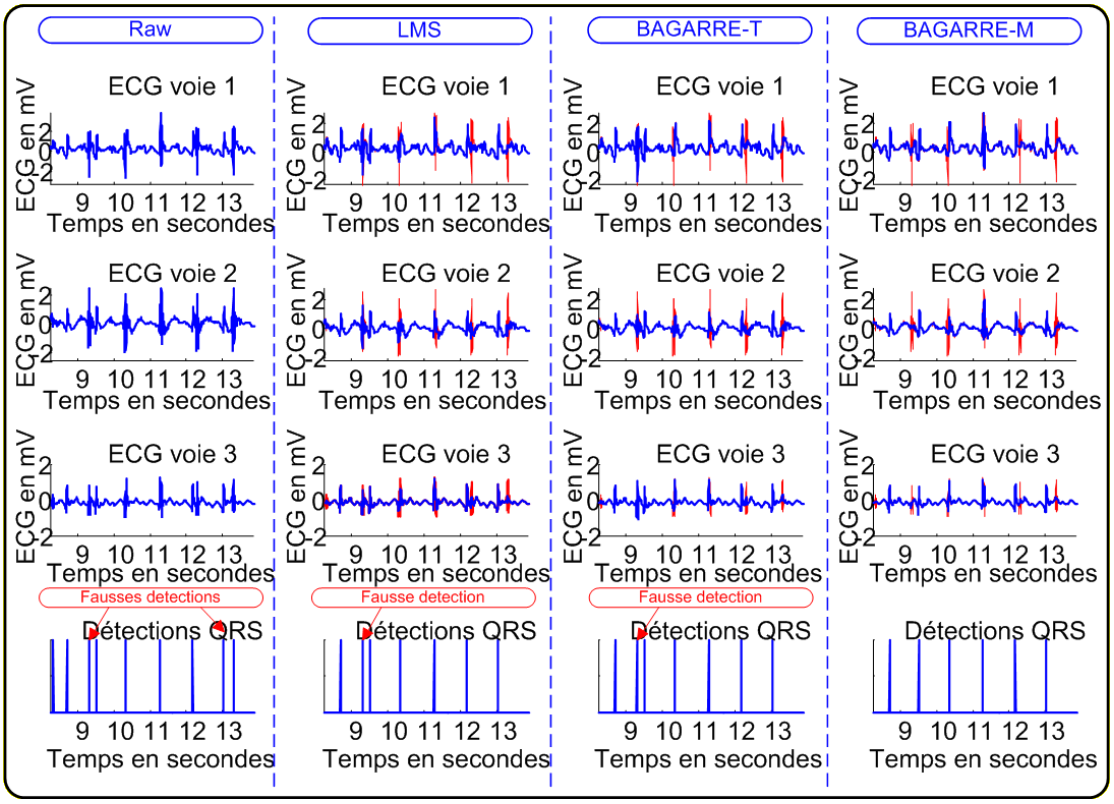


FIGURE 5.19 – Comparaison des méthodes Brut, LMS, BAGARRE-T et BAGARRE-M sur une séquence DW-EPI. (De haut en bas : dérivations 1 à 3 et détections QRS associées, en rouge signaux bruts, en bleu signaux débruités)

sont présents sur le signal, ce qui rend une évalution de leur énergie posssible.

Soient x_{brut} le signal ECG à débruiter, $x_{debruite}$ le signal résultant du débruitage, Bat l'ensemble des périodes temporelles lors desquelles se déroule un battement cardiaque et Art l'ensemble des périodes de présence d'artefacts de gradient (figure 5.20). Il est alors possible de définir l'évolution de l'énergie du signal (*Signal Energy Evolution* (SEE)), l'évolution de l'énergie du bruit (*Noise Energy Evolution* (NEE)) et l'évolution d'un pseudo rapport signal sur bruit (*Pseudo Signal to Noise Ration Evolution* (PSNRE)) par :

$$\begin{cases} \text{SEE} = 10 \times \log_{10}\left(\dfrac{\sum\limits_{i\in Bat}|x_{debruite}(i)|^2}{\sum\limits_{i\in Bat}|x_{brut}(i)|^2}\right) \\ \text{NEE} = 10 \times \log_{10}\left(\dfrac{\sum\limits_{i\in Art}|x_{debruite}(i)|^2}{\sum\limits_{i\in Art}|x_{brut}(i)|^2}\right) \\ \text{PSNRE} = \text{SEE} - \text{NEE} \end{cases}, \tag{5.33}$$

La qualité de débruitage d'une méthode de réduction des artefacts de gradient sera donc d'autant meilleure que son PSNRE sera élevé et son NEE faible. Néanmoins, une méthode sera aussi évaluée sur sa capacité à ne pas altérer les signaux ECG à proprement parler, ceci pourra être évalué grâce à son SEE, qui devra idéalement être proche de zéro.

Ce critère d'évaluation de la qualité de débruitage n'est cependant pas parfait. En effet, il part du

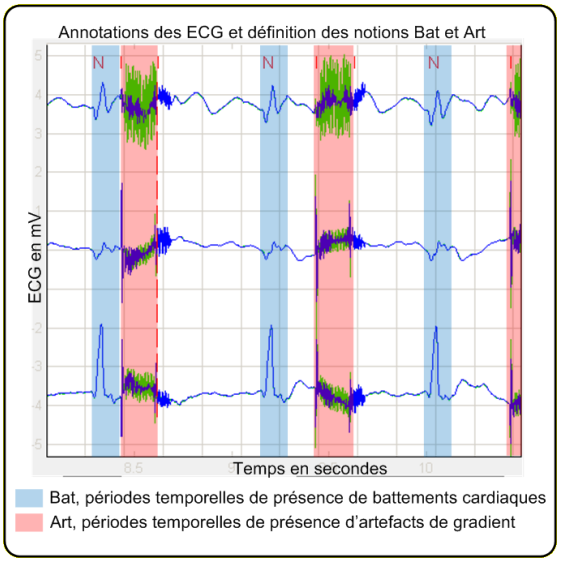


FIGURE 5.20 – Définition des périodes temporelles de présence des battements cardiaques (Bat) et d'artefacts de gradient (Art).

principe que, dans chaque région d'intérêt, la contribution énergétique d'un signal prend le pas sur l'ensemble des autres contributions. Cette approximation est possible lorsqu'on considère le SEE, car lors d'un battement cardiaque (QRS en particulier) l'effet MHD est négligeable et la contribution du signal induit par la respiration est minime bien que non négligeable. De plus, étant donné que les méthodes mises en place consisteront en la réduction des artefacts de gradient, l'ensemble des contributions prises en compte pour le calcul SEE doit idéalement être conservé.

L'approximation faite pour l'évaluation du bruit par le NEE est quant à elle plus incertaine. En effet, même si dans les régions Art la contribution énergétique des artefacts de gradient est la plus importante, il est difficile de négliger l'apport du signal ECG, de l'effet MHD et du signal issu de la respiration. A titre d'exemple, un artefact de gradient peut être induit simultanément avec l'onde T, période où l'effet MHD est le plus important et ce alors que le sujet respire de manière ample pour préparer une apnée, ou alors en même temps qu'un battement cardiaque. Il faudra donc être attentif lors de l'évaluation du NEE à ce que la méthode de débruitage ne supprime que les artefacts de gradient et ne réduise pas également les autres contributions du signal. Cela s'avérera particulièrement problématique pour les signaux issus du capteur 1, particulièrement perturbés par l'effet MHD.

Les méthodes BAGARRE ont été comparées selon ces critères avec la méthode LMS, qui réduit efficacement les artefacts en utilisant les signaux de gradients et la méthode *Bayésien*, qui débruite les signaux avec l'aide d'un modèle ECG. Ces résultats sont rassemblés dans le tableau 5.4. Comme cela a déjà été expliqué, ces critères ne sont pas parfaits et souffrent de certaines limitations, comme la non prise en compte du signal de l'effet MHD. Il est alors intéressant de s'intéresser tout d'abord aux résultats obtenus sur la dérivation 3, sur laquelle l'évaluation de la qualité de débruitage est la plus réaliste.

Sur cette dérivation, le PSNRE du *Bayésien* est légèrement inférieur à celui du LMS, mais le SEE à -0.26dB est plus gênant. En effet, comme le processus de création des artefacts n'est pas pris en compte, ces derniers ont tendance à déformer les paramètres des gaussiennes censées représenter le signal ECG. Ces déformations entraînent alors un "filtrage" excessif des battements cardiaques et donc un SEE négatif. Ce constat montre bien l'intérêt de développer une méthode qui tienne compte du processus de création des artefacts de gradient.

Les méthodes BAGARRE-T et BAGARRE-M permettent une augmentation de PSNRE par rapport au LMS respectivement de 0.35dB et 0.59dB. Cette amélioration est due à une meilleure réduction des artefacts, qui se traduit par un NEE plus faible, respectivement -0.32dB et -0.62dB. La prise en compte du signal ECG dans l'estimation des réponses impulsionnelles apporte donc une amélioration indéniable sur la qualité de débruitage. De plus, le SEE est relativement proche de zéro pour les deux méthodes

Méthode	Dérivation	SEE (dB)	NEE (dB)	PSNRE (dB)
LMS	1	-0.08	-1.38	1.30
	2	-0.17	-2.96	2.79
	3	-0.06	-1.38	1.32
Bayésien	1	-0.32	-0.79	0.47
	2	-0.72	-2.15	1.43
	3	-0.26	-1.52	1.26
BAGARRE-T	1	-0.05	-0.81	0.75
	2	-0.13	-2.62	2.49
	3	-0.02	-1.70	1.67
BAGARRE-M	1	-0.05	-1.00	0.95
	2	-0.26	-2.97	2.71
	3	-0.08	-2.00	1.91

TABLE 5.4 – Tableau comparatif de la qualité de réduction des artefacts sur la base de données ECG IRM

BAGARRE, ce qui tend à prouver qu'elles ne déforment pas le signal, mais réduisent uniquement le bruit. Le NEE légèrement inférieur pour BAGARRE-M par rapport à BAGARRE-T permet d'illustrer le lissage du signal ECG lors de l'estimation de z_k par filtrage de Kalman. Ces résultats démontrent l'efficacité des méthodes BAGARRE et leur supériorité par rapport aux méthodes de l'état de l'art, le LMS en particulier .

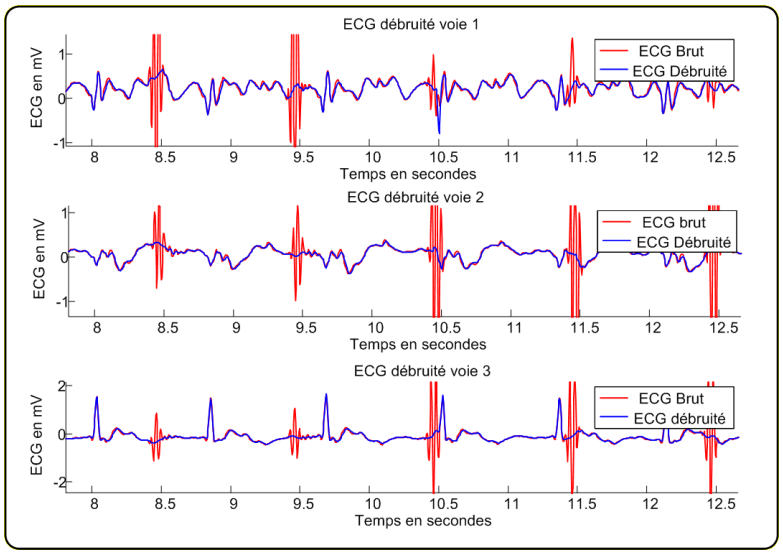


FIGURE 5.21 – Exemple de débruitage d'un signal ECG acquis en IRM pendant une séquence DW-EPI (FOV=24cm) (De haut en bas dérivation 3, 1 et 2, en rouge signaux bruts et en bleu signaux débruités)

En s'intéressant aux résultats obtenus sur les deux autres dérivations, une conclusion erronée, quant à la qualité de débruitage, pourrait être tirée. En effet, en considérant les résultats obtenus sur la dérivation 1, il peut être constaté que le PSNRE du LMS est meilleur que celui de BAGARRE-T ($+0.55$dB) et BAGARRE-M ($+0.35$dB), qui s'explique par un NEE plus faible (respectivement -0.57dB et -0.38dB). Ce résultat ne doit pas être interprété comme une réduction plus efficace des artefacts de gradient par le LMS, mais par le fait que la contribution de l'effet MHD n'est pas prise en compte dans le NEE. Ainsi, étant donné que le LMS ne considère le signal enregistré que comme une somme d'artefacts de gradient et du bruit, il s'évertue à faire tendre le signal ECG vers zéro à l'apparition d'un signal de gradient. Si celui-ci apparaît régulièrement en présence d'un effet MHD, l'estimation des réponses impulsionnelles va alors être biaisée et ceci va conduire à une déformation du signal ECG. Les méthodes BAGARRE en tenant compte de ces contributions pendant la procédure de filtrage ne déforment pas, ou peu, le signal et

ne suppriment que la contribution des artefacts de gradient (figure 5.21). Cette contribution est par conséquent moins énergétique sur cette dérivation (de l'ordre de 1dB) que sur la dérivation 3 (de l'ordre de 2dB). Le comportement des deux méthodes BAGARRE est d'ailleurs identique sur l'ensemble des voies.

5.3.5 Discussion

La qualité des deux nouvelles méthodes de débruitage a été établie dans la partie précédente. L'unification des approches espace d'état et de débruitage avec un modèle d'ECG et d'estimation des réponses impulsionnelles permet de traiter efficacement les signaux ECG acquis en IRM. L'application d'un détecteur QRS industriel aux signaux débruités permet alors de remplir les tâches de monitorage et de synchronisation avec une grande précision, même pour des acquisitions fortement bruitées, pour lesquelles les méthodes de l'état de l'art rencontrent actuellement des difficultés.

Ces deux méthodes peuvent être considérées comme les méthodes de référence pour la suppression des artefacts de gradient, tant leur résultat surclasse l'ensemble des méthodes existantes. BAGARRE-M est par ailleurs la méthode la plus efficace et la détection QRS atteinte est d'excellente qualité, sur l'ensemble des enregistrements bruités de la base de données ECG-IRM. Néanmoins, son application n'est pas compatible avec la synchronisation. Ceci est dû principalement à l'utilisation d'un signal de phase cardiaque, pour mettre en place le filtrage bayésien, qui est créé rétrospectivement à partir des détections QRS. Une méthode permettant de surpasser cette limitation a été présentée avec BAGARRE-T. Néanmoins, du fait du retard entre les estimations des réponses impulsionnelles et leur application, certains artefacts ne sont pas supprimés (figure 5.19) et la qualité de débruitage en est légèrement altérée.

Il pourrait être intéressant d'envisager d'autres pistes pour la création du signal de phase cardiaque, afin que celle-ci ne se fasse plus rétrospectivement mais de manière prospective. Une création de ce signal de phase prospective permettrait en outre d'appliquer le filtrage bayésien échantillon par échantillon. Par conséquent, la qualité de débruitage pourrait approcher celle obtenue par BAGARRE-M. Une approche potentielle est de garder le formalisme état-espace présenté dans la partie 5.3.3, mais de modifier la manière de créer le signal de phase. Ainsi, plutôt que d'attendre d'avoir détecter un QRS pour créer le signal de phase du dernier intervalle RR, il serait plus intéressant d'utiliser la méthode de prédiction de la durée de l'intervalle RR présentée dans la Section 5.2 à venir à partir de la dernière détection QRS, afin de créer ce signal de manière prospective.

Les méthodes BAGARRE ont été présentées et appliquées pour des sujets ayant des ECG normaux, sans présence d'Extra-Systoles Ventriculaires (ESV) entre autres. En présence d'ESV, le signal ECG ne peut plus être modélisé comme une somme de plusieurs gaussiennes, mais plusieurs cycles cardiaques doivent être considérées, selon la présence ou non d'ESV. Il est cependant envisageable de modéliser un tel signal, comme une succession pseudo-périodique de plusieurs cycles cardiaques. Différents modèles de cycle devraient être utilisés et le filtrage de Kalman devrait alors incorporer une étape supplémentaire de détermination du modèle à adopter à chaque cycle. Cette étape de classification supplémentaire peut d'ailleurs elle-même être effectuée par le filtrage, puisque la matrice de covariance de l'innovation (différence entre l'observation et son estimée) peut être utilisée pour vérifier la conformité des données par rapport à un modèle.

5.4 Conclusion

Comme dans la partie consacrée au contrôle optimal, nous avons montré dans cette partie notre intérêt pour la modélisation et l'application à des problèmes concrets. Les applications décrites ici imposaient un certain nombre de contraintes encore une fois en grande partie du fait de la présence de l'humain. En particulier, l'estimation en ligne des paramètres était une contrainte forte. En effet, il est indispensable de contrôler l'acquisition d'image par les imageurs IRM durant l'examen et en tenant compte de la position des organes à imager (en particulier le cœur qui est en mouvement permanent). Ainsi, le filtrage bayésien en général et le filtrage de Kalman en particulier se sont révélés être des méthodes particulièrement bien adaptées et efficaces pour résoudre ces problèmes de manière originale. Des images particulièrement nettes et jamais obtenues jusqu'à présent ont pu être acquises. Ce résultat est évidemment particulièrement motivant et les perspectives sont grandes. Les méthodes décrites permettent aussi

d'estimer l'incertitude liée aux estimations de paramètres. Cette gestion de l'incertitude peut s'avérer utile dans le cadre d'introduction de connaissance *a priori* mais aussi, lorsque c'est possible, d'explorer activement l'espace d'entrée du système dont on cherche à identifier les paramètres.

On notera encore une fois que les applications visées font intervenir l'humain. Cela induit des contraintes éthiques évidemment mais il reste que l'objectif est important. Les contraintes technologiques associées restent assez proches de celles décrites pour le contrôle optimal. Les signaux étudiés sont non-stationnaires du fait de la variation du signal ECG au cours du temps et il est indispensable de mettre au point des méthodes permettant de gérer cette non-stationnarité, c'est-à-dire de poursuivre la solution au fil du temps et pas de converger vers elle de manière globale. Il faut aussi pouvoir gérer les variabilités inter- et intra-personnes et l'évolution de ces variabilités au cours du temps et donc avoir une méthode générique et adaptative. Là encore, l'approche proposée est particulièrement bien adaptée.

Nous nous sommes aussi intéressé à modéliser le problème de régression sous la forme d'un problème d'estimation que nous avons résolu avec des méthodes de filtrage bayésien non-linéaire [85]. Dans ce même cadre, nous avons cherché à pousser ce problème de régression plus loin en considérant que nous observons l'information au travers de fonctions non-linéaires [91]. Les résultats que nous avons obtenus montrent qu'il est possible d'apprendre pratiquement aussi bien qu'avec des méthodes observant complètement les informations (au bruit près), comme les Machines à Vecteurs Supports ou *Support Vector Machines* (SVM) par exemple, mais avec une représentation plus compacte de la fonction recherchée et en affinant en ligne le résultat.

Chapitre 6

Modèles non-paramétriques et apprentissage non-supervisé

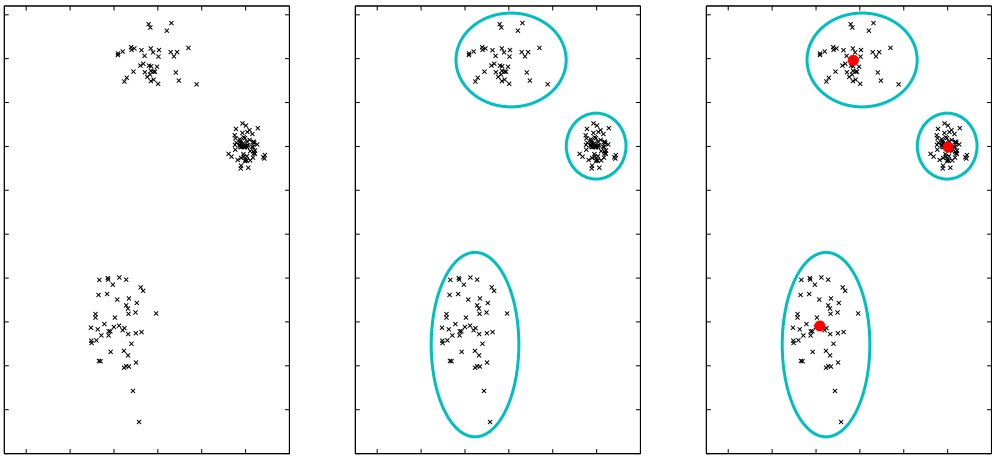
Jusqu'à présent, nous nous sommes intéressés à deux parties importantes de l'apprentissage artificiel que sont l'apprentissage par renforcement et l'apprentissage supervisé (via l'estimation de paramètres en ligne). Nous nous intéressons ici au dernier type d'apprentissage automatique, c'est-à-dire l'apprentissage non-supervisé et particulièrement à la quantification vectorielle. L'apprentissage par renforcement permet d'apprendre une politique de sélection d'actions séquentielles à partir de récompenses (apprendre à se comporter en recevant une information sur quoi faire). L'apprentissage supervisé permet d'apprendre les paramètres d'un modèle à partir d'exemples de ce que l'on cherche à modéliser. Ici, nous nous concentrons sur ce qu'il est possible de retirer de la structure ou de la distribution des données sans qu'aucune information ne vienne de l'extérieur (et donc sans en connaître un modèle dont on chercherait les paramètres). La quantification vectorielle nous servira ensuite à réaliser un *clustering* séparant l'espace des données en zones homogènes. Nous nous intéressons particulièrement à deux applications assez différentes de la quantification vectorielle. La première, qui a fait l'objet de la thèse de Béatrice CHEVAILLIER [28], consiste à segmenter des images d'IRM fonctionnelles rénales. Il s'agit de séparer automatiquement dans une image IRM les pixels représentant des compartiments différents d'un rein. La deuxième application est celle de la segmentation en locuteurs d'un flux audio (développée dans le cadre du projet LINDO). Ici, il s'agit de détecter dans un flux audio les changements de locuteur et d'ensuite affecter un label à chaque segment de manière à apparier les segments ayant été prononcés par le même locuteur. Il apparaît encore une fois que les contraintes liées à la présence de l'humain vont être importantes à prendre en compte. Ainsi, des méthodes en ligne, adaptatives et ne faisant pas d'hypothèse sur la stationnarité des distributions des exemples doivent être privilégiées.

6.1 Quantification vectorielle appliquée au *clustering*

L'objectif général de l'apprentissage non supervisé est de faire apparaître certaines structures sous-jacentes dans les données, par exemple en identifiant des sous-ensembles homogènes. La quantification vectorielle fait partie des méthodes d'apprentissage non supervisé pouvant aboutir à une partition des données, au même titre que la classification hiérarchique, l'analyse en composantes indépendantes, les machines à vecteurs supports à une classe (one-class SVM), certains arbres de décision, ... Elle trouve des applications dans des domaines aussi variés que la compression de données, la reconnaissance de formes et l'analyse par regroupement ou *clustering*.

Notons que le terme *clustering* est souvent traduit par partition, mais cela ne correspond pas tout-à-fait à l'acception courante de ce mot : s'il s'agit dans les deux cas de grouper les éléments d'un ensemble de données en différents sous-ensembles, appelés *classes* ou *clusters*, il est plus ou moins sous-entendu que, pour un *clustering*, chaque sous-ensemble contient des données qui sont homogènes, au sens où elles possèdent certains attributs similaires, ce qui n'est pas le cas en général pour une partition. On peut par exemple envisager de regrouper des objets d'après leur couleur. Un autre exemple est le suivant :

supposons qu'on ait un ensemble de points dans un plan, dont on connaît les coordonnées, et qui sont représentés sur la figure 6.1(a). De manière assez naturelle, on peut regrouper ces points d'après leur position en trois sous-ensembles (figure 6.1(b)) et attribuer à chacun un représentant (les points rouges sur la figure 6.1(c), situés « au centre » de chaque ensemble, pourraient convenir).



(a) (b) (c)

FIGURE 6.1 – Exemple de *clustering* de points d'après leur position dans un plan : données (a), regroupement en trois sous-ensembles « naturels » (b) et choix d'un représentant (c)

Les attributs seront représentés dans la suite par la variable ξ. Si les attributs sont des points de $\mathbb{R}^p$, la similarité entre deux éléments peut être mesurée par la distance euclidienne entre leurs attributs respectifs. D'autres distances peuvent être utilisées, comme celle de Hamming pour des vecteurs binaires. A chaque sous-ensemble obtenu, on peut associer un élément particulier, ou prototype, qui fait ou non partie des données classées, et qui est supposé représenter convenablement tous les éléments de ce sous-ensemble. Réciproquement, connaissant le représentant d'un des sous-ensembles, il est parfois possible de classer de nouvelles données, en les mettant dans le sous-ensemble du prototype qui leur ressemble le plus. Remarquons qu'il est fréquent que deux données soient plus proches l'une de l'autre que de leur représentant respectif, quand les classes sont mal séparées. Certaines méthodes de quantification vectorielle permettent de classer des données même dans ce cas. Remarquons que l'approche axiomatique du clustering est restée longtemps bloquée, le théorème d'impossibilité de Kleinberg concluant à l'impossibilité de trouver une fonction de clustering satisfaisant à trois propriétés simples qui semblent intuitivement indispensables [113]. Cependant, dans [132] les auteurs développent un nouveau formalisme et sont parvenus à axiomatiser convenablement le clustering.

6.1.1 Quantification vectorielle

Principes généraux

Soit $\mathcal{X} = \{\xi_1, \ldots, \xi_N\}$ un ensemble fini de points de $\mathbb{R}^p$. Nous supposons que c'est la réalisation d'une séquence de variables aléatoires vectorielles $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ identiquement distribuées de même loi qu'un vecteur aléatoire continu $\mathbf{X} : \Omega \longrightarrow V \subset \mathbb{R}^p$, où $(\Omega, \mathcal{E}, P)$ est un espace probabilisé et V une sous-variété de $\mathbb{R}^p$. Nous l'appellerons *échantillon* $\mathcal{X}$. Notons $\mathcal{B}_{(\mathbb{R}^p)}$ la tribu des boréliens de $\mathbb{R}^p$, $P_{\mathbf{X}}$ la loi de $\mathbf{X}$ sur $(\mathbb{R}^p, \mathcal{B}_{(\mathbb{R}^p)})$ et E l'espérance mathématique. En pratique, ni V, ni $P_{\mathbf{X}}$ en général ne sont connus ; seul l'échantillon $\mathcal{X}$ est donné. L'objectif des techniques de quantification vectorielle est de déterminer, grâce aux ξ_i, un ensemble fini $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ de vecteurs de référence, ou *prototypes* $\mathbf{w}_j \in \mathbb{R}^p$, $j = 1, \ldots, K$ qui représente « au mieux » l'échantillon $\mathcal{X}$ [136, 100]. Un vecteur donné $\xi \in V$ est décrit par le prototype $w(\xi)$ qui lui correspond le mieux, dans un sens à définir.

Définition d'un quantificateur vectoriel

Dans ce paragraphe, nous définissons formellement un quantificateur vectoriel et introduisons les notions de codeur et de décodeur, qui nous seront en particulier utiles pour définir par la suite la notion de carte préservant la topologie.

Définition 6.1 (Quantificateur vectoriel) *Soit $V \subset \mathbb{R}^p$ une sous-variété de $\mathbb{R}^p$, $\mathcal{I} = \{1, 2, \ldots, K\}$ un ensemble d'indices et $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ un ensemble de vecteurs $\mathbf{w}_j \in \mathbb{R}^p$, $j = 1, \ldots, K$. Un quantificateur vectoriel w de domaine V, de dimension p et de taille K est une application de V dans W qui à tout point ξ de V associe l'un des vecteurs $\mathbf{w}_j$, noté $w(\xi)$.*

$$\begin{aligned} w : V &\longrightarrow W \\ \xi &\longmapsto w(\xi) \end{aligned} \tag{6.1}$$

On le notera (V, W, w).

L'ensemble W est appelé, dans le cadre du *clustering*, ensemble de prototypes.

A tout quantificateur vectoriel de taille K, on peut associer une partition de V, voire de $\mathbb{R}^p$ quand $V = \mathbb{R}^p$, en K sous-régions ou cellules V_j, $j = 1, \ldots, K$. La $j^{\text{ième}}$ cellule est définie par :

$$V_j = \{\xi \in V : w(\xi) = \mathbf{w}_j\} = w^{-1}(\mathbf{w}_j) \tag{6.2}$$

Mesures de performance d'un quantificateur vectoriel

Pour analyser les performances d'un quantificateur vectoriel sur l'échantillon $\mathcal{X}$, nous utiliserons l'erreur quadratique moyenne, qui est l'espérance du carré de la distance euclidienne entre une donnée d'entrée et son représentant dans W. L'erreur quadratique moyenne est un cas particulier de distorsion moyenne. Nous pourrons alors définir la notion de quantificateur vectoriel optimal.

Définition 6.2 (Distorsion moyenne) *Soit d une fonction coût définie sur l'espace $\mathbb{R}^p \times \mathbb{R}^p$ à valeurs dans $\mathbb{R}$ ou $\mathbb{R}_+$. Soit un quantificateur vectoriel (V, W, w). On note $\mathbf{W}$ la matrice dont les colonnes sont les vecteurs prototypes $\mathbf{w}_j$, c'est-à-dire que $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$. La performance du quantificateur vectoriel (V, W, w) appliqué à l'échantillon $\mathcal{X}$ peut être mesurée par la distorsion moyenne*

$$\Psi(\mathbf{W}) = \mathrm{E}_{\mathbf{X}}[d(\mathbf{X}, w(\mathbf{X}))] = \int_V d(\xi, w(\xi)) \mathrm{d}P_{\mathbf{X}}(\xi) \tag{6.3}$$

ou encore

$$\Psi(\mathbf{W}) = \sum_{j=1}^{K} \Psi_j \tag{6.4}$$

où

$$\Psi_j = \int_{V_j} d(\xi, w(\xi)) \mathrm{d}P_{\mathbf{X}}(\xi) \tag{6.5}$$

La fonction d la plus utilisée correspond à l'erreur quadratique, c'est-à-dire au carré de la distance euclidienne entre deux vecteurs :

$$d(\xi, \eta) = \|\xi - \eta\|^2 = (\xi - \eta)^t (\xi - \eta) \tag{6.6}$$

C'est celle que nous choisirons pour toute la suite. D'autres possibilités pour le choix de d sont présentées dans [100].

Quantificateurs au plus proche voisin

Les quantificateurs au plus proche voisin (*nearest neighbor quantizer*) sont une catégorie particulièrement importante puisque tous les quantificateurs optimaux en font partie. Ce type de quantificateur permet de construire une partition de $\mathbb{R}^p$ en un nombre fini de polyhèdres.

Définition 6.3 (Polyhèdre de Voronoï) *Soit $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ un ensemble de vecteurs $\mathbf{w}_j \in \mathbb{R}^p$, $j = 1, \ldots, K$. Le polyhèdre de Voronoï de $\mathbf{w}_j$, noté $V_j^{(p)}(W)$, est l'ensemble des points de $\mathbb{R}^p$ qui sont plus proches de $\mathbf{w}_j$ que de n'importe quel autre $\mathbf{w}_i$:*

$$V_j^{(p)}(W) = \{\xi \in \mathbb{R}^p : d(\xi, \mathbf{w}_j) \leq d(\xi, \mathbf{w}_i)\ \forall i = 1, \ldots, K\} \tag{6.7}$$

Dans le cas où plusieurs prototypes sont à la même distance d'un vecteur $\xi \in \mathbb{R}^p$, ξ n'appartient qu'à un seul polyhèdre de Voronoï, celui correspondant au prototype de plus petit indice. L'ensemble des polyhèdres de Voronoï $\{V_j^{(p)}(W)\}_{1 \leq j \leq K}$ forme une partition de $\mathbb{R}^p$.

Quand il n'y aura pas d'ambiguïté sur W, nous noterons simplement $V_j^{(p)}$ le polyhèdre de Voronoï de $\mathbf{w}_j$.

Un exemple de polyhèdres de Voronoï d'un ensemble de vecteurs $\mathbf{w}_j$ est représenté figure 6.2.

Définition 6.4 (Quantificateur vectoriel au plus proche voisin) *Un quantificateur vectoriel (V, W, w) au plus proche voisin, dit aussi quantificateur de Voronoï, est tel que les cellules V_j sont définies par :*

$$V_j = \{\xi \in V : w(\xi) = \mathbf{w}_j\} = V \cap V_j^{(p)}(W) \tag{6.8}$$

Définition 6.5 (Cellule de Voronoï) *Dans ce cas, les $V_j, 1 \leq j \leq K$ sont appelées cellules de Voronoï de W dans V : chaque V_j est constituée de tous les points de V qui sont plus proches de $\mathbf{w}_j$ que de n'importe quel autre $\mathbf{w}_i$.*

Tout vecteur ξ appartenant à la cellule V_j est ainsi représenté par $w(\xi) = \mathbf{w}_j$, qui est le vecteur de l'ensemble W pour lequel l'erreur quadratique est minimale, i.e.

$$w(\xi) = \text{argmin}_{\mathbf{w}_j}\{d(\xi, \mathbf{w}_j)\} \tag{6.9}$$

Un exemple de cellules de Voronoï sur une variété V est représenté figure 6.3.

Pour distinguer la quantification vectorielle sur une variété V de celle dans $\mathbb{R}^p$, on parlera respectivement de la cellule de Voronoï V_j ou du polyhèdre de Voronoï $V_j^{(p)}$ d'un prototype $\mathbf{w}_j$.

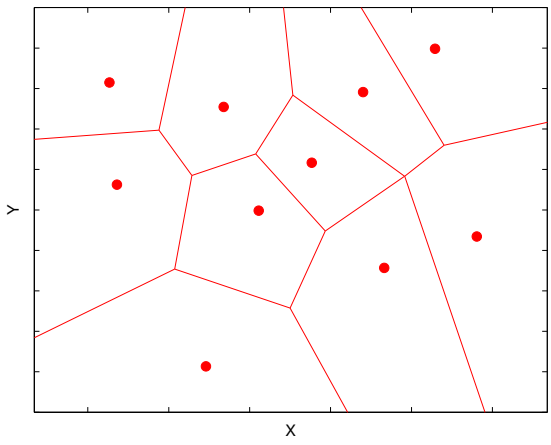


FIGURE 6.2 – Pohyhèdres de Voronoï d'un ensemble de prototypes dans $\mathbb{R}^2$

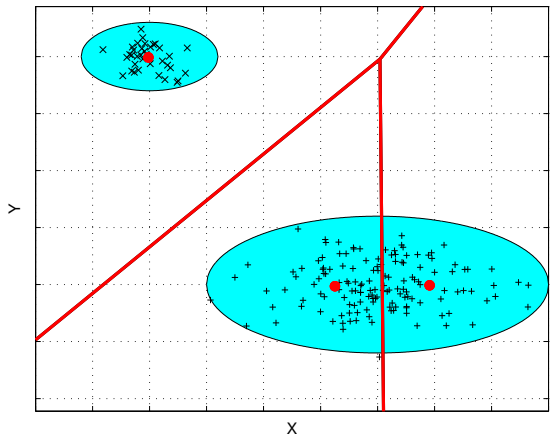


FIGURE 6.3 – Cellules de Voronoï d'un ensemble de trois prototypes sur une variété $V \subset \mathbb{R}^2$: sur les polyhèdres de Voronoï (trait continu), la cellule de Voronoï de chaque prototype (gros point) sur la variété V (zone colorée) correspond à l'intersection de son polyhèdre de Voronoï et de la variété V

Conditions d'optimalité pour la quantification vectorielle

Après avoir défini la notion de quantificateur vectoriel optimal, nous rappelons deux conditions nécessaires que doit vérifier un tel quantificateur, sachant qu'elles sont insuffisantes pour impliquer l'optimalité, même locale.

Définition 6.6 (Quantificateur vectoriel optimal) *Soit un quantificateur vectoriel (V, W, w) dont la taille K est supposée fixée. Ce quantificateur est optimal pour l'échantillon $\mathcal{X}$ et le coût quadratique d s'il minimise la distorsion moyenne*

$$\Psi(\mathbf{W}) = \mathrm{E}_{\mathbf{X}}[d(\mathbf{X}, w(\mathbf{X}))] \tag{6.10}$$

où $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ est la matrice dont les colonnes sont les vecteurs prototypes $\mathbf{w}_j$.

Pour qu'un quantificateur vectoriel soit optimal, il doit vérifier les conditions suivantes [1] :

1. Condition du plus proche voisin : pour un ensemble donné W de prototypes, les cellules de la partition optimale satisfont :
$$V_j \subset V_j^{(p)}(W) \tag{6.11}$$
2. Condition du centroïde : étant donnée une partition $\{V_j\}_{1 \leq j \leq K}$ de V, les vecteurs du code optimal pour l'échantillon $\mathcal{X}$ sont les centroïdes des V_j, c'est-à-dire qu'ils vérifient, pour un coût quadratique :
$$\mathbf{w}_j = \mathrm{E}_{\mathbf{X}}[\mathbf{X} | \mathbf{X} \in V_j], 1 \leq j \leq K \tag{6.12}$$

Un quantificateur vectoriel vérifiant les deux conditions nécessaires d'optimalité n'est cependant pas nécessairement optimal, même localement [100]. Toutefois, ces conditions suggèrent une méthode itérative d'amélioration d'un quantificateur, qui est utilisée dans l'algorithme de Lloyd généralisé [130].

6.1.2 Cartes préservant la topologie

Parallèlement à la quantification vectorielle, dans le cas où le support de la loi de $\mathbf{X}$ est une variété de $\mathbb{R}^p$, on peut construire un graphe topologique qui est une approximation de la triangulation de Delaunay (définie plus loin) des prototypes induite par V (on parlera plus brièvement de la variété encodée), et qui

1. Dans le cas où $\mathbf{X}$ n'est pas continue, une troisième condition est nécessaire : les points des frontières des cellules de Voronoï doivent avoir une probabilité nulle.

forme une carte préservant la topologie de cette variété [137]. L'ensemble peut être utilisé pour obtenir un *clustering* des données, en supposant qu'un cluster correspond à une composante connexe de la variété.

Les variétés peuvent être constituées d'un seul morceau ou bien d'une union de composantes connexes de dimensions éventuellement différentes. Quand la variété $V \subset \mathbb{R}^p$ est inconnue ou mal connue, la quantification vectorielle seule peut être insuffisante pour en séparer les différentes composantes connexes sauf dans le cas où chaque composante est incluse dans la cellule de Voronoï d'un seul prototype $\mathbf{w}_j$. Cependant, elle demeure en général insuffisante pour représenter la topologie de la variété. Une solution est d'adjoindre à l'ensemble de prototypes W une carte préservant la topologie de V, ce qui permettra de déduire de l'ensemble un *clustering* des données.

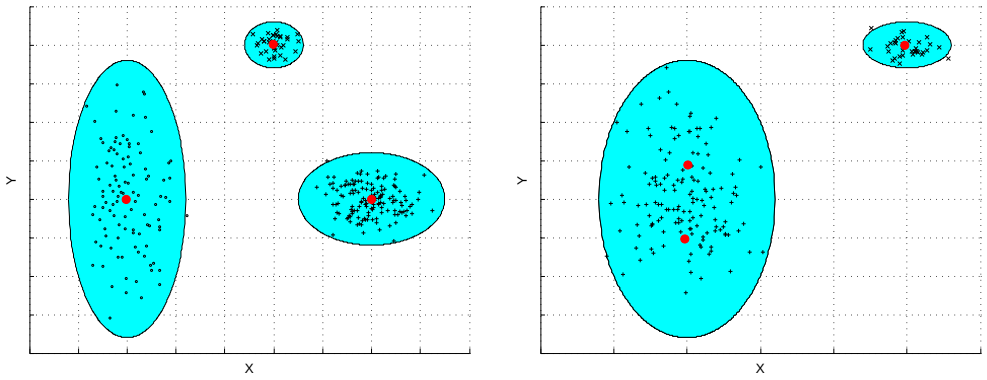


FIGURE 6.4 – Exemple de quantification vectorielle : sur la figure (a), les prototypes donnent une certaine idée de la topologie de la variété encodée puisque chaque prototype correspond à une composante connexe différente ; sur la figure (b), en revanche, il n'apparaît pas que les deux prototypes de gauche représentent des points appartenant à une même composante connexe.

Il apparaît intuitivement que, dans le graphe que nous souhaitons adjoindre à W, une relation doit exister entre deux prototypes si et seulement s'ils sont adjacents sur la variété. A un ensemble de prototypes, on peut associer le graphe qui met en relation chaque prototype et ses voisins « naturels » dans $\mathbb{R}^p$. Ce graphe, qui est la structure duale du diagramme de Voronoï, est appelé *triangulation de Delaunay*. Les sommets du graphes sont appelés neurones.

La triangulation de Delaunay complète peut être définie comme un graphe particulier qui relie les neurones dont les représentants ont des polyhèdres de Voronoï adjacents [137]. Par abus de langage, nous dirons parfois que le graphe relie les prototypes au lieu des neurones qu'ils représentent. Ainsi, deux neurones i et j sont reliés par un arc si et seulement si les polyhèdres de Voronoï des prototypes associés $\mathbf{w}_i$ et $\mathbf{w}_j$ sont adjacents.

Un exemple de triangulation de Delaunay et du diagramme de Voronoï est représenté figure 6.5. Cette triangulation est définie indépendamment de la variété encodée : tous les prototypes adjacents dans $\mathbb{R}^p$ sont reliés, même s'ils ne sont pas adjacents sur la variété.

6.1.3 Algorithmes de quantification vectorielle et construction de cartes préservant la topologie

Nous utiliserons deux algorithmes de quantification vectorielle :

- l'algorithme des K-moyennes, dit aussi de Lloyd généralisé [100, 130] que nous ne décrivons pas ici tant il est commun,
- l'algorithme du *Growing Neural Gas* (GNG) *with Targeting* (GNG-T) [65], variante de l'algorithme de GNG [67] qui est décrit ci-après.

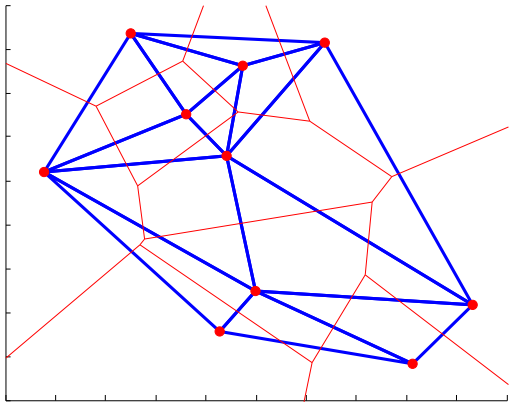


FIGURE 6.5 – Triangulation de Delaunay (trait épais) et diagramme de Voronoï associé (traits fins) d'un ensemble de prototypes (gros points) dans $\mathbb{R}^2$

Cependant, l'un comme l'autre visent à minimiser la distorsion donnée dans l'équation (6.10) par l'ajustement itératif de la position des prototypes $\mathbf{w}_j$ en utilisant seulement un nombre fini de données ξ_i, supposées identiquement distribuées selon une loi de probabilité inconnue. Le premier (K-moyennes) ne construit pas de graphe contrairement au deuxième (GNG-T).

Le principal défaut de l'algorithme des K-moyennes est que le résultat obtenu est très dépendant des conditions initiales et peut correspondre à un minimum local et non global de la distorsion [100]. Par ailleurs, le nombre de prototypes K doit être défini à l'avance, ce qui peut être particulièrement délicat pour des distributions de forme complexe dans des espaces de grande dimension.

L'algorithme GNG-T est une variante du GNG classique [67], lui-même dérivé de l'algorithme de *Neural Gas* associé à une compétition selon la règle de Hebb [136]. Si on compare cet algorithme à la version en ligne des K-moyennes, où seul le prototype le plus proche de la donnée courante ξ_i est modifié [18], l'ensemble complet des prototypes est ajusté à chaque itération, l'ampleur de la modification étant plus petite pour les $\mathbf{w}_j$ les plus éloignés de ξ_i. Ceci diminue le risque d'aboutir à un minimum local de la fonction minimisée Ψ_{NG}, laquelle tend, sous certaines conditions, vers la distorsion de l'équation (6.10). De plus, GNG-T permet de poursuivre la distribution des échantillons qui lui sont présentés. Ainsi cette distribution peut être non-stationnaire.

L'idée centrale de GNG-T est de contrôler la précision de la quantification en utilisant un argument statistique obtenu à partir de mesures calculées lorsque des exemples $\xi_i \in \mathcal{X}$ sont fournis à l'algorithme un par un. GNG-T a pour but de poursuivre la non-stationnarité de la distribution $p_t(\xi)$ sous-jacente à l'apparition des exemples, ce qui implique un échantillonnage en espace (sur $\mathcal{X}$) et en temps. Pour ce faire, définissons une époque $\mathcal{E}_t$ comme un ensemble d'exemples correspondant à un échantillonnage de $p_t(\xi)$ au temps t. Les exemples seront alors plutôt noté ξ_i^t, ce qui veut dire que ξ_i^t est le $i^{\text{ème}}$ échantillon utilisé pour estimer $p_t(\xi)$ au temps t.

Pratiquement, l'algorithme traite un graphe non-orienté G dont les nœuds (ou neurones) sont notés w, et les arrêtes sont étiquetées avec une valeur entière qui est leur "âge". Durant l'exécution de l'algorithme, chaque nœud $w_i \in \mathcal{X}$ est un prototype des valeurs prises par les échantillons, et il est associé à une accumulation d'erreur e_i et à un compteur n_i. Le nombre courant de nœuds est noté ν. La configuration de ce graphe a été adaptée de l'algorithme GNG [68], voir [66] pour plus de justifications. L'algorithme se décrit comme suit :

`Initialisation` G est un graphe vide.

`Répéter pour chaque époque`

`start_epoch` : L'époque $\mathcal{E}_t$ commence. $\forall\ 1 \leq i \leq \nu, n_i = 0, e_i = 0$.

`start_submit` : Prendre l'échantillon suivant ξ_j^t pour cette époque. Si il n'y a plus d'échantillons disponible pour $\mathcal{E}_t$, aller à `analysis`.

- Si $\nu < 2$, ajouter un nouveau prototype $w_k = \xi_j^t$ dans le graphe, avec $n_k = 0, e_k = 0$. Aller à `end_submit`.
- Trouver parmi tous les nœuds, le nœud w_{i_1} qui est le plus proche de ξ_j^t, et aussi trouver w_{i_2}, qui est le plus proche quand w_{i_1} n'est pas pris en compte.
- $n_{i_1} \leftarrow n_{i_1} + 1$ et $e_{i_1} \leftarrow e_{i_1} + \left\|\xi_j^t - w_{i_1}\right\|^2$
- Si elle n'existe pas encore, créer une arrête entre w_{i_1} et w_{i_2}.
- Augmenter l'âge des arrêtes partant de w_{i_1}, sauf celle entre w_{i_1} et w_{i_2} pour laquelle l'âge est réinitialisé à 0. Retirer les arrêtes plus vieilles que `age_max`.
- Mettre à jour le nœud gagnat : $w_{i_1} \leftarrow w_{i_1} + \alpha_1 \left(\xi_j^t - w_{i_1}\right)$.
- Mettre à jour les voisins de w_{i_1} : Pour tout w_k connecté à w_{i_1} : $w_k \leftarrow w_k + \alpha_2 \left(\xi_j^t - w_k\right)$.

`end_submit`: Aller à `start_submit`.

`analysis`: Retirer les nœuds w_k pour lesquels $n_k = 0$, puisqu'ils n'ont pas gagné pendant cette époque. Si $\nu < 2$, aller à `end_epoch`.

- Calculer $T = \frac{1}{\nu} \sum_{i=1}^{\nu} e_i$.

⋆ Trouver $i_{\max} = \text{argmax}_{1 \leq i \leq \nu} e_i$ ainsi que $i_{\min} = \text{argmin}_{1 \leq i \leq \nu} e_i$.

⋆ Si `target` $> T$, il y a trop de nœuds. Retirer $w_{i_{\min}}$. Aller `end_epoch`.

⋆ Plus de nœuds sont requis. Notons $w_a = w_{i_{\max}}$.

⋆ Trouver parmi les nœuds w_k connectés à w_a celui avec le plus grand e_k. Soit w_b ce nœud. Si w_a est isolé (*i.e* il n'a pas de voisin connecté), alors créer un nouveau nœud w_b de sort que $w_b = w_a$ (cloning).

⋆ Retirer les arrêtes (si elles existent) entre w_a et w_b.

⋆ Créer un nouveau nœud $w_c = \lambda w_a + (1 - \lambda) w_b$. Connecter w_a to w_c et w_b to w_c avec des arrêtes d'âge 0.

`end_epoch`

Après une époque $\mathcal{E}_t$, le graphe de prototypes quantifie la distribution p_t. Les nœuds fournissent exactement la même information que les K-moyennes [130]. Néanmoins, la principale différence tient en deux points. Premièrement, GNG-T ne nécessite pas de définir à l'avance le nombre de prototypes, mais plutôt d'ajuster ce nombre de manière à ce que l'erreur de quantification soit gardée constante et égale à un niveau prédéfini `target`. Lors de changements de la distribution, le nombre de prototypes peut varier dynamiquement, ce qui évite à la fois le sur-échantillonnage et le sous-échantillonnage de la distribution. Deuxièmement la différence avec les K-moyennes est que les prototypes sont construits comme un graphe, qui reflète la "forme" de la distribution, voir [138] pour des détails. Le graphe donne à GNG-T la faculté de s'adapter rapidement à des évolutions lentes ou même abruptes de la distribution.

On peut noter que seulement un nœud peut être ajouté à chaque époque avec l'algorithme décrit plus haut. Ceci peut ne pas être suffisant pour avoir une poursuite fidèle des changements de p_t dans le temps. Pour dépasser cette limitation, la même époque $\mathcal{E}_t$ peut être présentée successivement plusieurs fois à l'algorithme. Nous appellerons un *chunk* la succession de plusieurs époques. Il reste possible d'avoir de larges *chunks* tout en préservant le caractère temps-réel du processus sur un ordinateur standard. De plus, lorsque la précision de quantification est atteinte par l'algorithme dans un *chunk* le graphe devient instable durant la dernière époque du *chunk*. En effet, GNG-T ajoute et retire successivement un nœud à chaque époque dans ce cas. Ceci peut être évité en ignorant, dans la dernière époque d'un *chunk*, l'étape marquée ⋆ dans l'algorithme.

Enfin, comme c'est le cas dans beaucoup d'applications de la quantification vectorielle, mentionnons que les échantillons dans une époque ne doivent pas être corrélés. Sinon, GNG-T va poursuivre le processus qui corrèle les échantillons entre eux. C'est pourquoi il est parfois nécessaire de mélanger les échantillons dans une époque pour les présenter dans un ordre aléatoire.

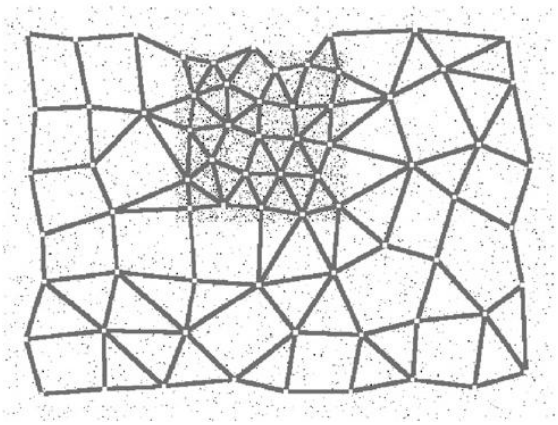
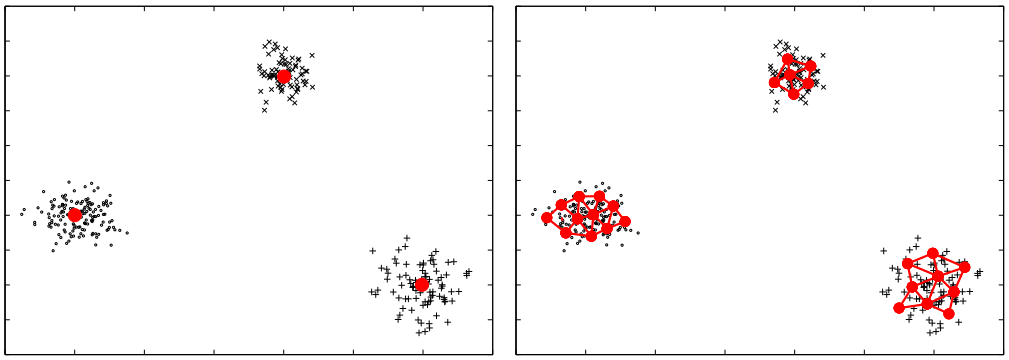


FIGURE 6.6 – Résultat de GNG-T sur une distribution artificielle de vecteurs à 2 dimensions

Exemples comparatifs

De nombreux exemples peuvent être trouvés dans la littérature, en particulier dans [66] pour GNG-T. Sur les figures 6.7 et 6.8, un exemple de quantification vectorielle d'un même ensemble de données par un algorithme de K-moyennes ou de GNG-T est présenté, les paramètres étant réglés dans l'optique d'en déduire un *clustering*.



(a) (b)

FIGURE 6.7 – Exemples de quantification vectorielle d'un même ensemble de données par un algorithme de K-moyennes (a) ou de GNG-T (b) : les croix, les signes plus et les ronds représentent les échantillons de la distribution, les gros points sont les prototypes après quantification vectorielle, reliés par des arcs pour GNG-T.

6.2 Segmentation semi-automatique d'images IRM-DRC rénales

Nous nous intéressons maintenant à un problème particulier que nous avons modélisé comme un problème de quantification vectorielle. Ce problème est celui de la segmentation automatique d'images IRM Dynamique à Rehaussement de Contraste (DRC). L'IRM-DRC permet de suivre le devenir d'un produit de contraste injecté dans l'organisme. Le produit n'est pas lui-même visible directement, mais modifie les temps de relaxation des protons des molécules d'eau voisines, donc le contraste des tissus avoisinants sur l'image reconstruite (voir Section 5.2.1 pour une explication rapide du fonctionnement de l'IRM). Ainsi, les différents tissus d'un organe apparaîtront plus ou moins intense dans l'image suivant que le

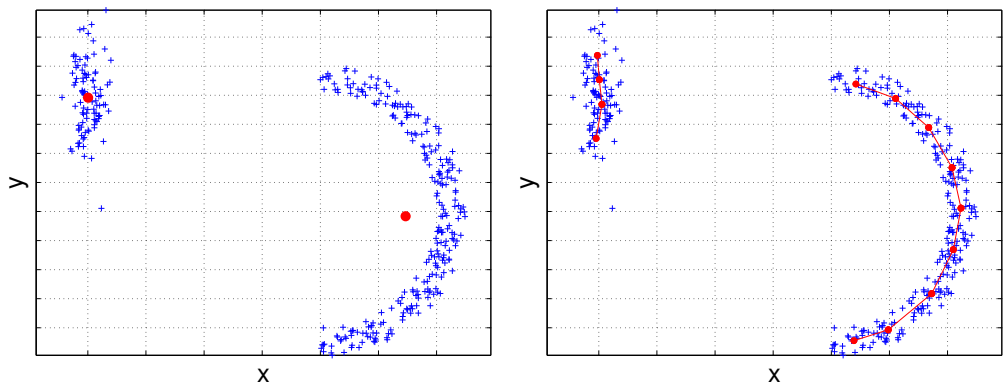


FIGURE 6.8 – Exemples de quantification vectorielle d'un même ensemble de données par un algorithme de K-moyennes (a) ou de GNG-T (b) : les croix représentent les échantillons de la distribution, les gros points sont les prototypes après quantification vectorielle, reliés par des arcs pour GNG-T.

produit de contraste sera concentré dans celui-ci ou pas. C'est un outil important d'étude de la fonction d'un organe, c'est-à-dire sa capacité à éliminer le produit de contraste.

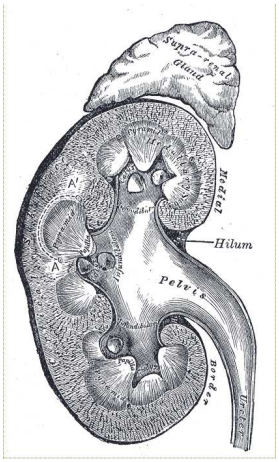
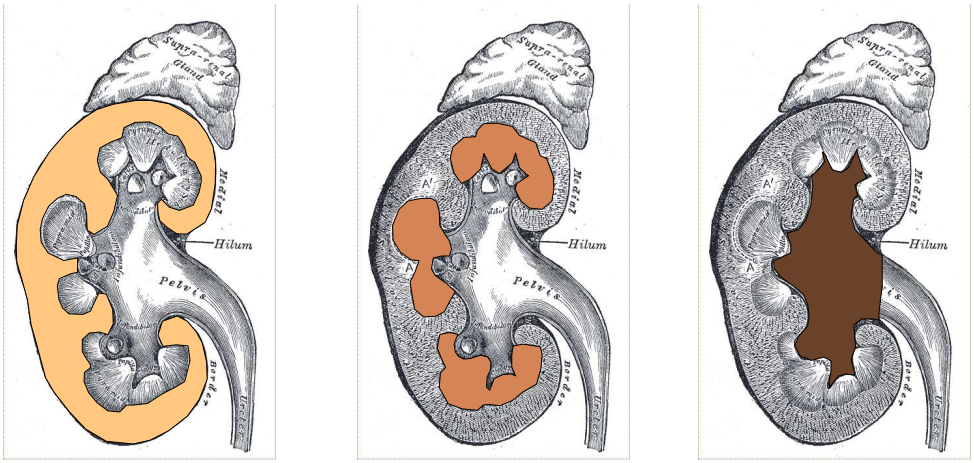


FIGURE 6.9 – Anatomie du rein (Reproduction d'une lithographie extraite de Gray's Anatomy of the Human Body).

En particulier, le rein est un organe filtrant le sang pour maintenir une composition fixe de celui-ci et produire des urines. Le produit de contraste injecté dans le sang doit donc passer par le rein et ses différents tissus avant d'être éliminé dans les urines. L'IRM est aujourd'hui préférée à la scintigraphie qui a le désavantage important d'irradier le patient. Néanmoins, il existe une relation linéaire entre l'intensité de l'image et la concentration en produit de contraste dans le tissu dans le cas de la scintigraphie. Ce lien est plus difficile à faire dans le cas de l'IRM qui offre une meilleure résolution spatiale et demande plus de temps de traitement. C'est pourquoi nous avons collaboré avec le CHU de Nancy et l'équipe IADI (Imagerie Adaptative, Diagnostique et Interventionelle) pour mettre au point une méthode de traitement semi-automatique de ces séquences d'images. Ici encore, une modélisation du problème est nécessaire. Les variabilités intra- et inter-patients doivent être prise en compte mais pas seulement. Il s'agit ici de tenir compte du fait que l'analyse doit servir au médecin dont l'avis peut aussi varier d'un spécialiste à l'autre, d'un patient à l'autre, *etc*.

Les reins sont en forme de haricot, et leur dimension approximative chez l'homme est d'environ

11 à 12 cm de long, 5 à 7 cm de large, et 2 à 3 cm d'épaisseur [102]. Leur structure interne est très complexe (cf. figure 6.9 [2]). On peut cependant distinguer trois parties principales, ou compartiments : le cortex, la médullaire et les cavités, représentés sur la figure 6.10 [3]. Les cavités sont formées du pelvis et des calices rénaux, et constituent la partie supérieure des voies urinaires (dans la suite de notre étude, nous ne considérerons pas la totalité du pelvis mais nous nous limiterons à la partie incluse dans le sinus rénal). Le cortex est situé en périphérie du rein, alors que la médullaire en est une partie interne, formée des pyramides dites de Malpighi dont les bases sont tournées vers la partie corticale [21]. Le sang passe donc par ces trois tissus rénaux afin d'être filtré et le produit de contraste suivra aussi le même chemin. Ainsi, il est possible de qualifier la qualité de la fonction rénale en suivant le trajet du produit de contraste dans ces trois compartiments rénaux.



(a) (b) (c)

FIGURE 6.10 – Anatomie du rein : cortex (a), médullaire (b) et cavités (c).

Les séries d'images que nous utiliserons seront réalisées après injection de chélates de gadolinium. Les acquisitions sont répétées sur plusieurs minutes pour obtenir une séquence complète. Trois images issues d'une même séquence sont présentées figure 6.11, mettant en évidence la modification d'intensité due au produit de contraste.

Afin de qualifier la fonction rénale, les médecins urologues et radiologues doivent analyser finement les séquences d'images. Pour cela, une segmentation de chacune des images de la séquence (qui peut en comporter plusieurs centaines) serait nécessaire. Devant l'ampleur de la tâche, ce travail est souvent résumé à l'identification d'une petite partie de l'image (appelée RoI pour *Region of Interest*) dont on suppose qu'elle est commune au même tissu sur toutes les images. Evidemment, cela induit potentiellement des erreurs d'interprétation de l'examen.

Dans cette section, nous construisons puis testons différents classificateurs pour les voxels d'une coupe rénale. Grâce à des données simulées (voir [29] pour plus d'information sur la méthode de simulation), nous donnons certaines explications sur l'origine des qualités et des défauts de chacun d'entre eux, et nous proposons des améliorations. Nous évaluons également leurs performances sur des données réelles.

Nous ne pouvons pas réaliser d'apprentissage supervisé sur cette tâche pour plusieurs raisons. La première est qu'il nous faudrait des segmentations parfaites réalisées par des radiologues pour chacune des images et pour plusieurs séquences. Comme indiqué plus haut, cela n'est pas faisable du fait du temps que cela demanderait. Ensuite, rien ne dit que les segmentations fournies représenteraient la réalité des tissus de l'organe. En effet, les variabilités de segmentation inter- et intra-radiologues sont importantes, indiquant qu'il n'existe pas réellement de "vérité". Enfin, il n'est pas évident que les résultats appris resteraient valables pour d'autres patients.

2. http ://www.bartleby.com/107/illus1127.html
3. http ://fr.wikipedia.org/wiki/Fichier :Kidney_nephron.png

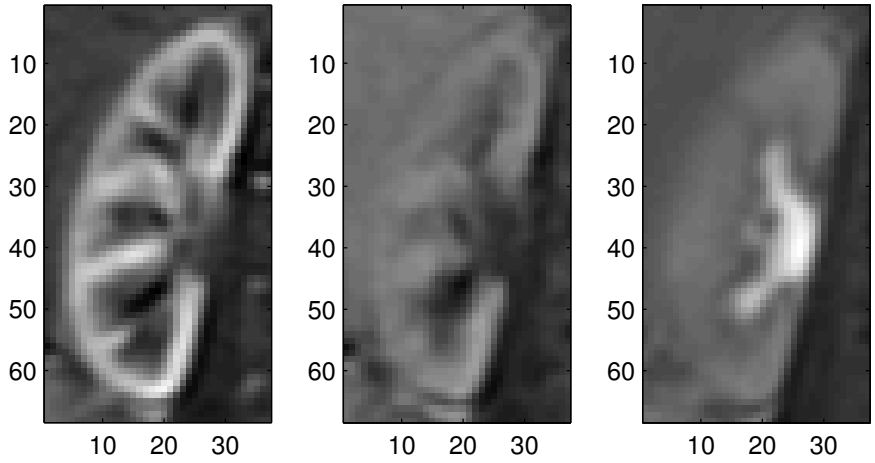


FIGURE 6.11 – Exemples d'images extraites d'une séquence d'IRM dynamique au pic artériel (à gauche), pendant la filtration (au milieu) et la phase tardive (à droite).

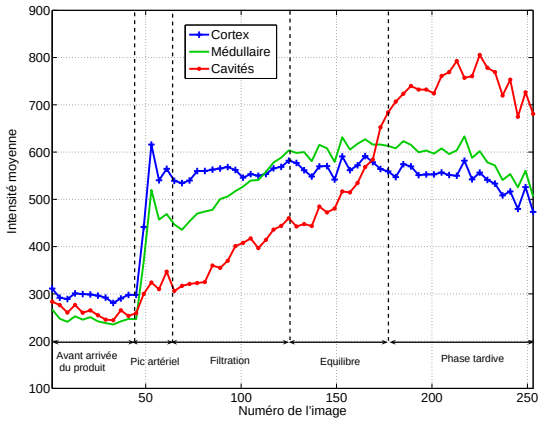


FIGURE 6.12 – Exemple de courbes temps-intensité typiques pour le cortex, la médullaire et les cavités.

Chaque pixel de l'image possède une courbe d'intensité qui évolue au cours du temps (intensité lumineuse pour un pixel variant en fonction de la présence ou non du produit de contraste). Une courbe temps-intensité moyenne pour chaque compartiment du rein peut être déduite d'une séquence : cette courbe traduit l'évolution de la concentration du produit de contraste dans chacun de ces trois compartiments. Cinq phases successives, reliées au mécanisme de filtration et d'évacuation des urines décrit ci-dessus, peuvent être observées sur la figure 6.12 [4] :

1. Baseline (environ 60 s) : l'agent de contraste n'a pas encore atteint la zone rénale ; le cortex possède un niveau de signal plus élevé car il contient une quantité d'eau plus importante ; la médullaire et les cavités ont des signaux assez comparables de sorte qu'on ne peut les distinguer durant cette phase.
2. Pic artériel (environ 15 s) : ce moment correspond à l'arrivée du produit de contraste par les vaisseaux sanguins. Le cortex présente théoriquement le pic artériel de plus grande amplitude, mais en raison de la vascularisation importante de l'ensemble du rein, des pics apparaissent également dans les autres compartiments.

4. Remarquons que l'échelle des temps est graduée en numéro d'images, et correspond donc à un temps réel non linéaire, les acquisitions dans la deuxième moitié étant plus espacées que dans la première

3. Filtration (environ 100 s) :
   – dans le cortex, le signal reste à un niveau élevé ; une légère montée peut éventuellement être observée.
   – le signal de la médullaire augmente également avec un léger retard par rapport au cortex, la pente de la courbe en étant plus raide,
   – le signal des cavités augmente un peu.
4. Équilibre (environ 150 s) : les signaux du cortex et de la médullaire deviennent difficiles à distinguer, les cavités sont encore peu rehaussées.
5. Phase tardive (environ 5 min) : l'urine est évacuée ; alors que les signaux du cortex et de la médullaire restent semblables et diminuent légèrement, les cavités se remplissent et l'intensité du signal correspondant devient très élevée.

La construction des classificateurs se fait donc grâce à un *clustering* par quantification vectorielle des courbes temps-intensité (éventuellement normalisées) des voxels de la coupe de référence, selon les principes exposés au chapitre 6. Différentes variantes, s'appuyant néanmoins sur un ensemble d'hypothèses communes, seront étudiées. En effet, si certaines peuvent donner satisfaction sur les données simulées, leurs performances peuvent s'avérer insuffisantes sur des données réelles. Il s'agit toujours de déduire d'une quantification vectorielle de leurs courbes temps-intensité, un *clustering*, puis une classification des voxels rénaux. Notre objectif étant non pas d'extraire le rein, mais d'en segmenter les structures internes (cortex, médullaire, cavités), un masque global à N voxels x_i, $1 \leq i \leq N$, est créé sur la coupe de référence avant la segmentation fonctionnelle. Seules les courbes temps-intensité des voxels de ce masque, obtenues à partir des p images de la séquence IRM, servent à la quantification vectorielle.

6.2.1 Hypothèses communes en lien avec la quantification vectorielle

La quantification vectorielle est la première étape de la segmentation fonctionnelle rénale. Un *clustering* des données et de l'espace des courbes temps-intensité en est déduit. Ce *clustering* dépend de l'algorithme utilisé et des hypothèses issues de la physiologie du rein.

Hypothèses sur les données communes aux différentes méthodes

Soit $\mathbf{I}_i = (I_{i1}, \ldots, I_{ip})$ le vecteur à p composantes associé au voxel x_i, où $I_{i\tau}$ est le contraste du voxel x_i au temps τ (on l'appellera également courbes temps-intensité du voxel x_i). A partir de ce vecteur, par normalisation, on construit le vecteur $\xi_i = (\xi_{i1}, \ldots, \xi_{ip})$ à p composantes, où $\xi_{i\tau}$ est le contraste normalisé du voxel x_i au temps τ.

Les N vecteurs ξ_i peuvent être considérés comme une réalisation d'une séquence de variables aléatoires $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ identiquement distribuées de même loi qu'un vecteur aléatoire continu $\mathbf{X} : \Omega \longrightarrow V \subset \mathbb{R}^p$, où $(\Omega, \mathcal{E}, P)$ est un espace probabilisé et V une sous-variété de $\mathbb{R}^p$. Ni V, ni la loi $P_{\mathbf{X}}$ ne sont connues ; seul l'échantillon $\mathcal{X} = \{\xi_1, \ldots, \xi_N\}$ est donné [5]. La quantification vectorielle permet de déterminer, grâce aux ξ_i, un ensemble fini $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ de prototypes $\mathbf{w}_j \in \mathbb{R}^p$, $j = 1, \ldots, K$ qui représente au mieux l'échantillon $\mathcal{X}$. Un vecteur donné $\xi \in V$ est décrit par le prototype $w(\xi)$ qui lui est le plus proche, au sens de la distance euclidienne.

6.2.2 Données

Huit séquences d'IRM de perfusion rénale à rehaussement de contraste pour des reins normaux sont à notre disposition. Ces acquisitions ont été réalisées sur huit patients avec un rein sain et un rein pathologique. Seul le rein sain a été retenu pour cette étude. Les examens ont été réalisés sur un appareil General Electric Healthcare à 1,5 T (corps entier). Des séquences LAVA (écho de gradient 3D ultra-rapide) ont été utilisées, avec les paramètres suivants :

– taille de la matrice : 256×256 pixels ;

5. Remarquons qu'on pourrait aussi considérer que chaque compartiment correspond à une distribution de loi différente, mais dans le cadre de la quantification vectorielle, on suppose une loi de probabilité commune permettant de distinguer les trois compartiments, par exemple en ayant trois modes.

– taille des voxels : entre $1,172$ et $1,875$ mm dans le plan de coupe, 10 mm d'épaisseur de coupe
– angle de bascule : $15°$;
– T_R : $2,3$ ms ;
– T_E : $1,1$ ms.

L'examen dure en moyenne une dizaine de minutes. Un rectangle recouvrant la totalité du rein est tout d'abord sélectionné, la taille de la matrice correspondante variant entre 47×35 et 84×59 pixels dans la coupe contenant la plus grande proportion de tissu rénal. Des segmentations seront réalisées par deux radiologues suivant le protocole décrit au paragraphe 6.2.5 page 107.

Le patient bougeant et respirant durant l'acquisition, il a été nécessaire de recaler les images. Pour éviter les risques de dérives, nous choisissons l'image enregistrée au pic cortical comme référence : toutes les images de la série seront recalées sur cette image. La mesure de similarité choisie est l'information mutuelle, en raison des changements de contraste pendant la perfusion et du fait que nous sommes amenés à recaler des images temporellement éloignées. Comme il nous semble impossible de distinguer localement les changements de contraste dus aux mouvements ou à la diffusion de l'agent de contraste, nous préférons nous limiter à des transformations rigides. Il est demandé à l'opérateur d'extraire un rectangle contenant le rein et le dépassant de quelques pixels à chaque extrémité. En effet, les transformations ne peuvent être considérées comme uniformes que localement, et le rein doit occuper la majorité du rectangle, pour que ce soient les pixels qui le représentent, et non ceux des organes voisins, qui influencent le plus la mesure de similarité. L'information mutuelle est estimée à l'aide de fenêtres de Parzen, selon la méthode exposée dans [140]. Après un recalage au pixel près, un recalage subpixel avec rotation est réalisé. Cette méthode a été également choisie mais avec des transformations élastiques dans [258], de manière indépendante, les résultats ayant été publiés après que nous l'avons utilisée.

6.2.3 Création des trois compartiments finaux

Si l'algorithme des k-moyennes est utilisé avec $K = 3$ prototypes, alors les prototypes résultant de l'application de l'algorithme devraient représenter les courbes temps-intensité moyenne des trois compartiments, éventuellement normalisées (cortex, médullaire et cavités). Néanmoins, il y a tout de même une grande différence entre les courbes des pixels pour un même compartiment (voir figure 6.13), même après normalisation. La distance euclidienne entre deux vecteurs peut même être plus petite entre deux courbes de pixel n'appartenant pas au même compartiment qu'entre deux pixel de compartiments différentes. On peut donc tester des valeurs plus élevée de K. Les cellules créées pourront être fusionnées manuellement. Il faut tout de même conserver une valeur de K assez faible pour que cela soit possible. Une valeur de $K \leq 6$ est raisonnable.

Avec l'algorithme de GNG-T, dans le cas idéal, la triangulation de Delaunay induite comprend trois sous-graphes connectant les neurones associés aux prototypes d'un même compartiment. En réalité, à cause du bruit et du volume partiel (le fait qu'un voxel puisse chevaucher 2 compartiment rénaux), on obtient un seul graphe avec un nombre de prototypes relativement élevé (entre 10 et 30 sur les données réelles). Un regroupement manuel n'est donc pas envisageable.

La procédure que nous avons utilisée est la suivante.

– Extraction des cavités : les cavités sont le compartiment anatomique contenant les voxels dont le rehaussement est maximal en fin de perfusion. Appelons prototype semence celui dont l'intensité moyenne est maximale en phase tardive. Tous les prototypes dont l'intensité moyenne durant cette phase est supérieure à un premier seuil s_1 et qui sont connectés dans le graphe G au prototype semence par un chemin ne passant que par des prototypes qui eux-mêmes vérifient le critère, sont supposés représenter les cavités.
– Séparation du cortex et de la médullaire : le taux de filtration dépend de la nature des tissus et est utilisé pour séparer cortex et médullaire. La pente des courbes temps-intensité pendant la phase de filtration est évaluée automatiquement par une méthode de régression linéaire standard pour tous les prototypes n'appartenant pas aux cavités. Appelons prototype semence celui dont la pente est maximale durant cette phase. Un nœud est attribué à la médullaire si la pente correspondante est inférieure à un seuil s_2 et s'il est relié au prototype semence par un chemin ne passant que par des prototypes qui eux-mêmes vérifient le critère de seuil. Les prototypes restants sont attribués au cortes.

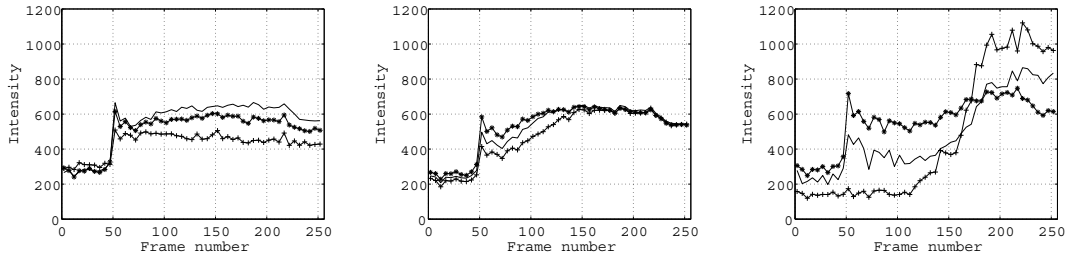


FIGURE 6.13 – Exemples de courbes temps-intensité pour le cortex (a), la médullaire (b) et les cavité (c) pour un même rein.

Les deux seuils s_1 et s_2 sont initialisés de sorte que le cortex représente approximativement 50% du rein et les cavités 20%. La seule intervention manuelle est leur réglage par un opérateur. Comme l'algorithme est très rapide, l'ajustement peut se faire en temps réel. Il est assez aisé car, à chaque changement de niveau des seuils, un nombre important de voxels avec une évolution temporelle semblable est ajouté ou retiré. Remarquons que le second critère ne pourrait être utilisé pour séparer les cavités des autres compartiments. En effet, sur la figure 6.13(c), un pic artériel, non prévu en théorie mais relativement important, peut être observé sur des courbes temps-intensité de voxels attribués aux cavités ; ce pic est dû à l'importante vascularisation du rein et également à l'effet de volume partiel pour certains voxels.

6.2.4 Normalisation des données

A priori l'algorithme des K-moyennes peut être appliqué à des données brutes ξ_i comme dans [33]. Néanmoins, chaque vecteur est divisé par sa norme de manière à tenir compte de la forme de la courbe et pour atténuer les hétérogénéités dans l'illumination due à l'acquisition. Pour GNG-T, il n'est pas possible de réaliser une telle normalisation mais toutes les intensités I sont d'abord remplacées par $(I-I_B)/(I_L-I_B)$, oùI_B est la valeur moyenne pour la *baseline* et I_L est la valeur moyenne de la phase tardive pour le rein entier. Cette manipulation permet d'obtenir une dynamique similaire pour tous les reins tout en conservant l'ordre relatif dans les intensités ce qui est nécessaire pour l'algorithme GNG-T.

6.2.5 Segmentations manuelles anatomiques de référence pour les données réelles

Deux radiologues expérimentés ont examiné les séquences pour délimiter d'abord un masque du rein entier, puis les structures internes : cortex, médullaire et cavités. La procédure retenue est la suivante :

- visualisation de la séquence complète,
- sélection d'une image en phase tardive, sur laquelle les cavités apparaissent nettement, détermination du masque global puis segmentation des cavités,
- sélection du pic cortical, segmentation du cortex, puis de la médullaire (par différence entre le cortex et les cavités précédemment délimitées).

Le masque global ainsi défini peut légèrement différer d'un radiologue à l'autre en raison du flou des images et parce que les images choisies pour réaliser la segmentation peuvent être différentes. Le masque global finalement retenu est l'intersection des masques globaux des deux radiologues. Les trois ROIs définies par la suite sont incluses dans ce masque global, qui est également utilisé par la suite pour la segmentation fonctionnelle. Un exemple de segmentation manuelle est présenté figure 6.14(a) et (b). Notons que cette manière de procéder peut « favoriser » les cavités au détriment de la médullaire : en raison du volume partiel, certains voxels composés d'un mélange de médullaire et de cavités ont un comportement dominant de cavités en phase tardive et seront considérés comme tels parce que ce compartiment est extrait en premier, alors qu'il ressemble davantage à de la médullaire aux alentours du pic cortical.

6.2.6 Validation quantitative

Une des segmentations manuelles est choisie comme référence. Toute les autres segmentations (les fonctionnelles obtenues par les algorithmes proposés et l'autre segmentation manuelles anatomique) seront comparées à cette référence. Chaque segmentation d'un compartiment est considéré comme une carte binaire pour laquelle les valeurs sont égales à 1 à l'intérieur du compartiment et 0 en dehors. Nous notons R la référence et T la segmentation testée. Quatre types de pixels peuvent alors être distingués en prenant en compte leurs labels dans R et dans T :

Type du pixel	label dans R	Label dans T
Vrai positif (TP)	1	1
Faux négatif (FN)	1	0
Faux positif (FP)	0	1
Vrai négatif (TN)	0	0

Des critères discriminants sont alors calculés pour chaque compartiment :

- Pourcentage de recouvrement (*overlap*) $PO = 100 \times TP/(TP+FN)$, i.e. pourcentage des pixels du compartiment de référence qui sont dans le compartiment test aussi.
- Pourcentage extra $PE = 100 \times FP/(TP+FN)$. Une segmentation parfaite fournirait une valeur $PO = 100\%$ et $PE = 0\%$. Des valeur élevées de PO et PE pour une segmentation donnée pourrait par exemple indiquer qu'une sur-segmentation du compartiment correspondant. Une valeur élevée de PE associée à une faible valeur de PO peut indiquer une position globalement mauvaise de la zone associée au compartiment.
- Index de Similarité $SI = (2 \times TP)/(2 \times TP + FN + FP)$. SI est sensible aux différences en taille est en position [257]. Par exemple, des zones de même taille qui partage la moitié de leurs pixels donnerait $SI = 1/2$. Une zone en couvrant une autre de taille deux fois inférieur donnerait $SI = 2/3$. Pour une segmentation parfaite on a $SI = 1$.

6.2.7 Résultats et discussion

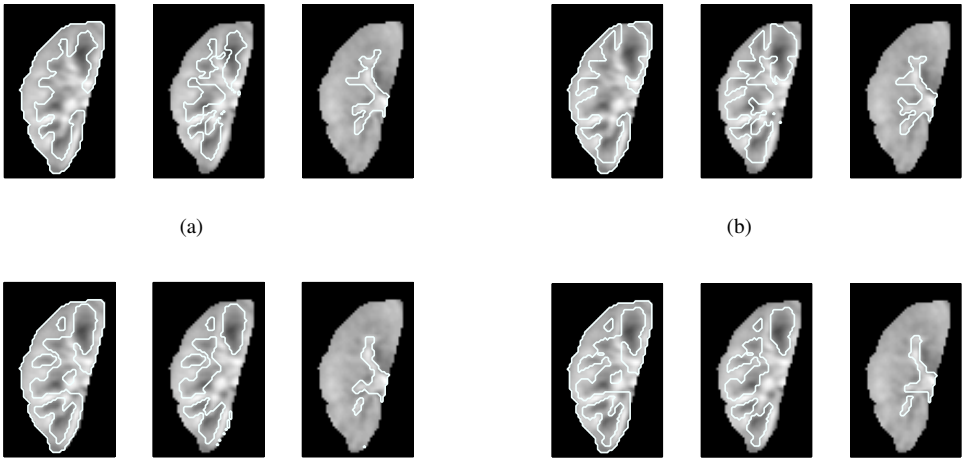
Des exemples de deux segmentations manuelles et de deux segmentations fonctionnelles automatiques sont montrées sur la figure 6.14. Dans ce cas précis, la taille des compartiments varie de 532 à 700 pixels pour le cortex, de 375 à 559 pour la médullaire, de 161 à 217 pour les cavités. Les tests sont réalisés avec les options suivantes :

- quantification vectorielle par K-moyenne appliquée sur des vecteurs normalisés ξ_i de manière que $\|\xi_i\| = 1$ et une fusion manuelle est opérée pour $K = 3$ à 6,
- quantification vectorielle par GNG-T et fusion manuelle en fixant deux seuils indépendants.

Notons que si GNG-T était appliqué aux vecteurs normalisés, la méthode de fusion définie dans la Section 6.2.3 ne sera plus valable puisque la transformation peut modifier les intensités relatives des pixels. Afin de tester visuellement la consistence qualitative des segmentations, les contours de chaque compartiment sont superposés aux image IRM (les pixels en dehors du rein sont masqués et apparaissent en noir). Les images sont sélectionnées dans la phase de perfusion pendant laquelle le compartiment est le mieux visible, soit le pic artériel pour le cortex et la médullaire et en phase tardive pour les cavités.

Le résultat qualititatif visuel entre ces images et la segmentation fonctionnelle est très bonne pour les deux méthodes semi-automatiques. Au contraire, sur l'exemple de cortex montré sur la figure 6.15, on voit qu'un simple seuillage sur les intensités sur une image unique ne fournit pas une segmentation statisfaisante : sur la figure 6.15(a) gauche, le seuil est trop bas et une grande partie de la médullaire est sélectionnée. En augmentant le seuil, le résultat ne s'améliore pas car une grande partie du cortex est alors mise de côté (figure 6.15(a) centre et droite). Un phénomène similaire peut être observé pour les cavités sur la figure 6.15(b). Cette méthode ne sera donc plus utilisée pour comparaison tant elle donne de mauvais résultats.

Des comparaisons quantitatives entre les segmentations sont présentées dans le tableau 6.1 et les figures 6.16 et 6.17. Les segmentations de l'opérateur OP1 sont choisies comme référence. Tout d'abord, les résultats moyens pour les segmentations de l'opérateur OP2 sont présentés dans le tableau 6.1. Elles peuvent être comparées aussi avec les résultats du K-means avec $K = 7$ sans normalisation qui est



(a) (b)

(c) (d)

FIGURE 6.14 – Exemples de segmentations manuelles anatomiques par les opérateurs OP1 (a) et OP2 (b), et de segmentations fonctionnelles après quantification vectorielle par K-moyennes (c) et par GNG-T (d) pour un même rein : pour chaque segmentation le cortex est montré sur la gauche, la médullaire au milieu et les cavités sur la droite.

(a) (b)

FIGURE 6.15 – Segmentations du cortex (a) et des cavités (b) en utilisant une méthode simple de seuillage de l'intensité pour 3 seuils croissants (de gauche à droite).

la meilleure méthode présentée dans [33]. Les moyennes pour les trois critères de similarité de la Section 6.2.6 pour les huit cas sont évaluées. Le pourcentage de pixels bien classifiés (WCP : *Well Classified Pixels*) est aussi calculé : pour les segmentations d'un compartiment, c'est la somme des TP pixels pour les 8 cas divisé par le nombre total de pixels pour ce compartiment. Pour le rein global, c'est la somme de tous les TP pixels, quel que soit le compartiment, divisé par la somme de tous les pixels de tous le reins. Ce pourcentage est un peu différent de la moyenne des overlap puisque les petits reins ont moins d'influence sur cette mesure. Concernant PO (overlap) et PE (extra pixels), ces critères doivent être examinés ensemble pour mettre en évidence la qualité de la segmentation complète : la plupart du temps une augmentation de PO s'accompagne d'une augmentation de PE. Par exemple, dans la première colonne des tableaux 6.1(a) et (b), on peut voir que PO et PE augmentent pour le cortex, alors qu'ils décroissent pour la médullaire : ces évolutions montre qu'il existe un léger chevauchement entre le cortex et la médullaire mais aussi une augmentation de la sursegmentation du cortex au détriment de la médullaire.

Les comparaisons quantitatives des résultats entre la segmentation manuelle de référence et n'importe laquelle des segmentations fonctionnelles sont similaires à celles obtenues en comparant la référence et la segmentation de l'autre radiologue, quel que soit le compartiment considéré. En d'autres termes, les segmentations fonctionnelles sont aussi ressemblantes à la référence que les segmentations anatomiques le sont entre elles. Néanmoins, les résultats sont moins bons avec l'algorithme des K-moyennes sans normalisation.

Pour tous les types de comparaisons, l'index de similarité est du même ordre de grandeur. Le pourcentage de pixels globalement bien classés est de toute façon plus élevé pour la plupart des segmentations fonctionnelles et varie entre 74.4 % et 77.6 % pour les K-moyennes avec normalisation et GNG-T, alors que sa valeur est de 74.9 % pour la segmentation manuelle par l'autre opérateur. Le plus mauvais score pour tous les critères est obtenu globalement pour la médullaire : la frontière cortico-médullaire n'est en effet pas très bien délimitée et ces deux compartiments restent difficiles à isoler objectivement.

Concernant les K-moyennes appliquées sur des vecteurs normalisés, des résultats très satisfaisants sont obtenus directement pour $K = 3$ pour presque tous les reins, et ne sont pas nécessairement améliorés par l'augmentation de K. Ceci tend à prouver que dans la plupart des cas, les courbes temps-intensité normalisées sont des caractéristiques discriminantes pour distinguer les trois types de pixels. De plus, aucune étape de fusion n'est nécessaire. Ceci n'est pas le cas pour les courbes non normalisées comme cela est indiqué dans [33]. Les résultats des K-moyennes avec $K = 3$ et pour GNG-T sont très semblables : ils sont légèrement meilleurs pour GNG-T grâce à sa flexibilité qui en fait une méthode plus adaptée aux cas plus difficiles. Dans le tableau 6.1, les valeurs des critères apparaissent en gras s'ils sont au moins égaux à ceux obtenus pour la segmentation manuelle par OP2. Ces résultats montrent que chacun des trois compartiments est bien identifié, *i.e.* la segmentation complète est en effet satisfaisante. Parmi les différentes variantes des algorithmes testées, ces deux méthodes sont les plus performantes pour la segmentation de reins sains.

Dans une seconde étape, OP1 étant toujours la référence, la dispersion des résultats peut être observée sur la figure 6.16 pour SI et sur la figure 6.17 pour PO et PE ; seulement les résultat K-moyennes avec $K = 3$ sont présentés parmi les méthodes avec normalisation, puisque les segmentations obtenues sont les meilleures. Chaque rein est testé seulement une fois. La variance sur les huit cas est bien plus importante pour les K-moyennes avec $K = 7$ sans normalisation que pour les segmentations manuelles. D'un autre côté, elle est la plupart du temps bien plus faible que pour les autres méthodes et particulièrement pour les K-moyennes avec $K = 3$.

6.2.8 Discussion

Les sections précédentes ont montré qu'il était possible de segmenter une coupe de manière à mettre en évidence les différents compartiment du rein eu utilisant l'évolution temporelle de l'intensité des pixels.

Parmi les variantes testées, les K-moyennes sur vecteurs normalisés avec $K = 3$ et le GNG-T semi-automatique paraissent les mieux adaptées pour la segmentation des structures rénales internes. En effet, n'importe quelle segmentation anatomique et n'importe quelle segmentation fonctionnelle obtenue par l'une ou l'autre de ces deux méthodes se ressemblent autant que les deux segmentations anatomiques.

TABLE 6.1 – Valeur des critères de comparaison pour les segmentations des trois compartiments et le résultat global (référence : OP1) pour OP2, K-moyennes avec $K = 3$ à 6 avec normalisation (KM3n à KM6n), GNG-T et K-moyenne avec $K = 7$ sans normalisation (KM7). Les résultats qui sont au moins équivalents à OP2 (gras italique) pour les K-moyennes avec $K = 3$ et pour GNG-T apparaissent en gras. Les critères affichés sont les pourcentages de pixels bien classifiés (WCP), overlap (PO), extra pixels (PE) et l'index de similarité (SI).

Test	OP2	KM3n	KM4n	KM5n	KM6n	GNG-T	KM7
WCP (%)	***69.8***	**83.6**	86.6	88.8	89.2	**83.7**	75.9
PO (%)	***71.8***	**82.8**	85.2	88.3	88.5	**83.2**	74.9
PE (%)	9.2	19.3	23.4	29.0	27.8	21.9	17.3
SI	***0.79***	**0.82**	0.82	0.81	0.82	**0.81**	0.77

(a) Cortex

Test	OP2	KM3n	KM4n	KM5n	KM6n	GNG-T	KM7
WCP (%)	84.8	72.2	65.5	62.1	62.2	73.0	75.7
PO (%)	84.0	72.8	66.1	62.7	62.8	73.0	76.7
PE (%)	***56.6***	**36.2**	30.3	24.9	24.9	**33.7**	52.1
SI	***0.70***	**0.70**	0.67	0.66	0.66	**0.71**	0.68

(b) Médulaire

Test	OP2	KM3n	KM4n	KM5n	KM6n	GNG-T	KM7
WCP (%)	74.9	70.5	71.5	69.1	71.2	68.7	68.2
PO (%)	73.9	69.1	69.8	65.9	69.5	69.8	66.2
PE (%)	***16.1***	**14.5**	16.2	13.9	14.2	**11.1**	18.0
SI	***0.77***	0.75	0.75	0.73	0.75	**0.77**	0.71

(c) Cavités

Test	OP2	KM3n	KM4n	KM5n	KM6n	GNG-T	KM7
WCP (%)	***74.9***	**77.5**	77.0	76.5	77.1	**77.6**	73.5

(d) Rein entier

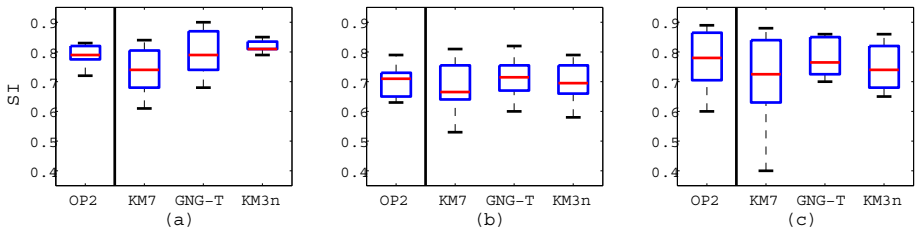


FIGURE 6.16 – Index de similarité SI (la référence étant OP1) : de gauche à droite, les diagrammes en *boîte à moustaches* pour la segmentation par OP2, K-moyenne avec 7 classes sans normalisation (KM7), GNG-T et K-moyennes avec 3 classes avec normalisation (KM3n) pour le cortex (a), la médullaire (b) et les cavités (c). La boîte a des lignes pour le quartille inférieur, médian, et supérieur. Les moustaches indiquent la plus petite et la plus haute valeur.

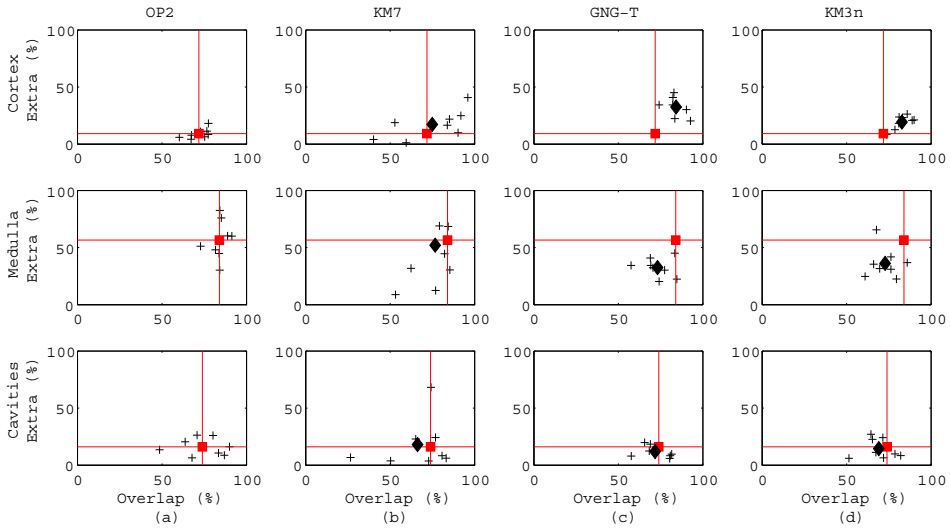


FIGURE 6.17 – PE en fonction de PO (référence : OP1) pour le cortex, la médullaire et les cavités (de haut en bas) pour la segmentation manuelle par OP2 (a), K-moyenne à 7 classes sans normalisation (b), GNG-T (c) et K-moyennes avec 3 classes avec normalisation (d) ; $\square$, et $\Diamond$, représentent les valeurs moyennes sur les huit reins respectivement pour la segmentation manuelle par OP2 ou pour les méthodes semi-automatiques.

L'hypothèse de séparation des composantes connexes n'est pas critique pour ces deux méthodes.

Pour les reins sains testés, les résultats donnés par les K-moyennes à trois classes sur des vecteurs normés sont la plupart du temps satisfaisants. La quantification vectorielle des courbes temps-intensité permet ainsi d'aboutir directement à la segmentation attendue, en un processus entièrement automatique. Cependant, en cas de résultat aberrant, il n'y a pas de possibilité de correction par un opérateur. Il est cependant possible de tester un autre nombre de classes (inférieur à 7), dont certaines sont par la suite fusionnées manuellement par un opérateur, ou bien d'utiliser la méthode semi-automatique avec GNG-T. Pour les K-moyennes avec $K \leq 3$, nos résultats sont un peu différents de ceux exposés dans [258] avec une méthode similaire ; sur quatre examens formant la base de données, deux ont été réalisés avec des séquences similaires aux nôtres mais acquises à 3 T au lieu de $1,5$ T, et pour 55 instants au lieu de 256. Les auteurs considèrent que 5 à 7 classes sont nécessaires pour obtenir une segmentation convenable ; seules 2 ou 3 classes sont conservées, les autres étant considérées comme dues au volume partiel ou à des morceaux d'autres organes conservés dans le masque rénal. Pour l'un des cas, la médullaire et les cavités sont regroupées dans une même classe, et pourraient être séparées par un post-traitement parce que les classes sont disjointes sur la segmentation ; ce qui n'est généralement pas le cas sur les séries dont nous disposons. La segmentation est directement réalisée sur les données 3D, sachant que les coupes extrêmes du rein ne sont pas conservées car les images ne sont pas d'assez bonne qualité.

Pour la méthode semi-automatique avec GNG-T, la quantification vectorielle permet, en tenant compte de l'évolution temporelle globale, de réduire la taille des données et de diminuer le bruit, afin que l'étape de fusion, basée sur des propriétés physiologiques, soit facile à faire pour un opérateur : elle consiste uniquement en un réglage assez grossier de deux seuils indépendants ; la méthode peut être une solution alternative offrant davantage de flexibilité que les K-moyennes pour le traitement des cas difficiles, car le nombre de prototypes représentant les données est nettement plus élevé. On peut même envisager de modifier la valeur de la cible T pour accroître la souplesse de la méthode.

Des tests supplémentaires devront être menés sur une base de données plus importante contenant des reins sains et des reins pathologiques. Pour ces derniers, comme de nouveaux comportements temporels risquent d'apparaître, une adaptation des critères de fusion pour GNG-T ainsi qu'une recherche d'un nombre de classes K optimal pour les K-moyennes devront être effectuées. Il est probable que la différence de représentativité des voxels due à la dilatation des cavités et à l'amincissement du cortex et

de la médullaire, justifiera l'usage de GNG-T plutôt que celui des K-moyennes.

Les courbes fonctionnelles diffèrent très peu en fonction de la technique utilisée ; il faudra cependant également vérifier que c'est également vrai pour les paramètres rénaux fonctionnels qui en sont tirés.

Notons enfin que le WCP traduit bien la qualité des segmentations obtenues, en accord avec les autres critères. Nous pourrons ainsi utiliser l'erreur de classement $1 - \text{WCP}$ comme critère de risque pour estimer cette qualité dans le cadre de la généralisation.

6.2.9 Conclusion

Dans ces travaux l'humain est omniprésent. En effet, les développements présentés dans cette section ont pour but d'analyser des images permettant d'interpréter le bon fonctionnement des organes internes d'une personne. Mais aussi, le résultat de ce traitement vise à aider le radiologue à accomplir son travail d'analyse. De ce point de vue, le gain de temps est considérable : alors qu'il faut une dizaine de minutes pour réaliser une segmentation manuelle, seulement une vingtaine de secondes est nécessaire pour une segmentation fonctionnelle.

D'un point de vue algorithmique, nous avons du, une fois encore, utiliser des méthodes répondant à certaines contraintes imposées par cette présence de l'humain aux deux bouts de la chaîne. Tout d'abord, l'apprentissage non-supervisé était de mise du fait de l'absence d'une vérité terrain unique. En effet, nous avons vu qu'il existait une variabilité d'annotation non-négligeable entre opérateurs et que cette variabilité était au moins aussi importante entre deux opérateurs qu'entre un opérateur et la méthode automatique. Cette méthode d'évaluation a d'ailleurs permis convaincre les radiologues du bien-fondé de cette technique qui, non seulement n'introduit pas plus de différence sur les segmentations que la variabilité inévitable entre opérateurs, mais en plus fournit des résultats reproductibles. Nous n'avons pas eu l'occasion, du fait du temps important que cela nécessite, de faire une étude de la variabilité intra-annotateur c'est-à-dire de demander à un même opérateur d'annoter manuellement une même séquence à plusieurs jours ou semaines d'intervalle. Néanmoins, la littérature montre que cette variabilité existe dans d'autres domaines et n'est pas négligeable. Notons que nous avons travaillé sur des méthodes de génération artificielle d'image IRM dans le but de quantifier plus précisément les performances de nos algorithmes indépendamment d'annotations manuelles [29].

Afin de permettre aux radiologues de garder la main sur une partie de l'annotation, des méthodes semi-automatique ont été mises au point. En effet, il est important pour un praticien de pouvoir comprendre les résultats qu'il interprète et le fait de pouvoir regrouper à la main ou de régler certains seuils en observant le résultat de ces réglages permet de concerner une compréhension suffisante pour valider le résultat.

Enfin, les méthodes développées devaient tenir compte de la variabilité des images qui diffèrent d'une personne à une autre du fait de la taille des organes et de la qualité de leur fonctionnement. C'est pourquoi nous avons opté pour des méthodes d'apprentissage non supervisé.

6.3 Segmentation en locuteurs d'un flux audio

Dans cette section, nous rapportons les résultats de travaux traitant de la segmentation en locuteurs d'un flux audio commencés avant la thèse [187] et poursuivis huit années plus tard dans le cadre du projet ITEA2 LINDO. Ce travail a été réalisé en collaboration avec nos collègues Hervé FREZZA-BUET et Jean-Louis GUTZWILLER [103, 193]

6.3.1 Position du problème et état de l'art

La segmentation en locuteurs d'un flux audio (en anglais *speaker diarization* ou *speaker clustering*) est un processus ayant pour objectif de découper et étiqueter automatiquement un flux audio en segments homogènes en termes de locuteurs. Il existe beaucoup d'applications à ce traitement qui est souvent un pré-requis à l'indexation de données multimédia, au sous-titrage automatique ou à l'aide active aux personnes mal-entendantes. Cela peut aussi permettre l'adaptation au locuteur pour la reconnaissance automatique de la parole ce qui aide à améliorer les performances de ces systèmes. Dans les applications

réelles (traitement de bulletin d'informations radiodiffusés, audio-videosurveillance), rien n'est connu *a priori* à propos des locuteurs (aucun modèle n'est disponible), de leur nombre ou du nombre de fois ou chaque locuteur parlera dans le flux. Aussi, la qualité des données audio peut varier de manière importante. Nous ferons tout de même l'hypothèse que les locuteurs ne parlent pas en même temps et que donc un seul locuteur est enregistré à la fois. Le processus de segmentation peut alors être décomposé en deux processus distincts : (*i*) les frontières des segments homogènes sont détectées (idéalement, les segments sont les plus longs possibles), (*ii*) une phase de regroupement des segments permet d'assigner une étiquette à chaque segment de manière non-supervisée.

Ce problème a été largement traité dans la littérature de ces dix dernières années. Néanmoins, si la détection des frontières de segments est réalisée au fil de l'eau (en ligne), la phase de regroupement et d'étiquetage intervient après coup (hors ligne). L'intégralité du flux audio doit alors être disponible ce qui introduit un délais considérable dans le traitement des données [46, 6]. Il existe très peu de méthodes en ligne [217] et la plupart d'entre elles nécessitent la connaissance *a priori* d'un modèle des locuteurs intervenant dans le flux. Cette limitation empêche les applications *anytime* d'utiliser des traitement de ce type. Pourtant, les applications *anytime* sont de plus en plus demandées dans des domaines tels que la surveillance ou la génération automatique des minutes d'une réunion [6].

Nous proposons donc une méthode en ligne et non-supervisée (sans modèle) pour réaliser la segmentation en locuteurs d'un flux audio. Le traitement en ligne et en temps réel est atteint parce que les informations relatives à chaque locuteur sont résumées dans un graphe de petite dimension dans l'espace des caractéristiques extraites du signal audio obtenu par des méthodes de quantification vectorielle. En particulier la méthode GNG-T [66] est utilisée pour construire ce graphe. Comme indiqué dans la Section 6.1.2, GNG-T calcule un réseau de neurones topologiques dont la taille est automatiquement adaptée à la distribution des données et aucune information *a priori* ne doit être fournie à propos de cette taille (contrairement à d'autres algorithmes comme k-moyennes ou *neural gas*). Cet algorithme est aussi capable de poursuivre la distribution des données si celle-ci change avec le temps ce qui est le cas pour l'application visée.

6.3.2 Extraction de caractéristiques

Dans ce travail, le signal de parole est échantillonné à 8 kHz et quantifié sur 16 bits. Il est analysé sur une fenêtre glissante dont la taille devrait rester suffisamment petite (de l'ordre de 30ms). En effet, il est important d'analyser le signal sur une échelle temporelle où il peut être considéré comme stationnaire ce qui est le cas pour des tranches temporelles de 10 à 100ms en fonction du son prononcé [206].

Du fait de la quantité trop importante de données obtenues après échantillonnage même sur une fenêtre d'analyse assez courte, cette quantité est réduite en utilisant la redondance naturelle du signal de parole. Des caractéristiques de dimension plus faibles sont alors extraites. Plusieurs types de caractéristiques peuvent être considérées mais un choix assez commun est l'utilisation des coefficients cepstraux ou *MEL Cepstrum Frequency Coefficients* (MFCC) [169].

L'analyse en fréquences MEL [6] fournit des caractéristiques faiblement corrélées qui sont pertinentes pour la reconnaissance automatique de parole mais aussi pour l'identification de locuteurs [112]. Ainsi, 16 coefficients MFCC sont extraits de fenêtres de 32ms ($N = 256$) qui se recouvrent de 22ms (voir Figure 6.18). Ces coefficient composent alors un *vecteur acoustique* à 16 dimensions. Toutes les 10ms ($L = 80$), un vecteur acoustique est donc extrait.

La Figure 6.19 décrit la manière dont les MFCC sont obtenus. Tout d'abord, une transformée de Fourrier rapide (FFT [7]) est appliquée à N échantillons ($s[n]_{n\in[0,N-1]}$) pour obtenir les N coefficients du spectre ($S[j]_{j\in[0,N-1]}$). Le module du spectre est alors utilisé comme entrée pour un banc de filtres MEL pour obtenir le vecteur MEL-*spectrum* ($S[k]_{k\in[0,15]}$). Le banc de filtres MEL approxime en fait le traitement non-linéaire du signal acoustique effectué par l'oreille, transformant l'échelle linéaire des *Hz* en un échelle non-linéaire [238] en utilisant la relation suivante :

$$f_{\text{MEL}} = 2595 \times \log(1 + \frac{f_{Hz}}{700})$$

6. le nom vient du mot *melody*
7. Fast Fourrier Transform

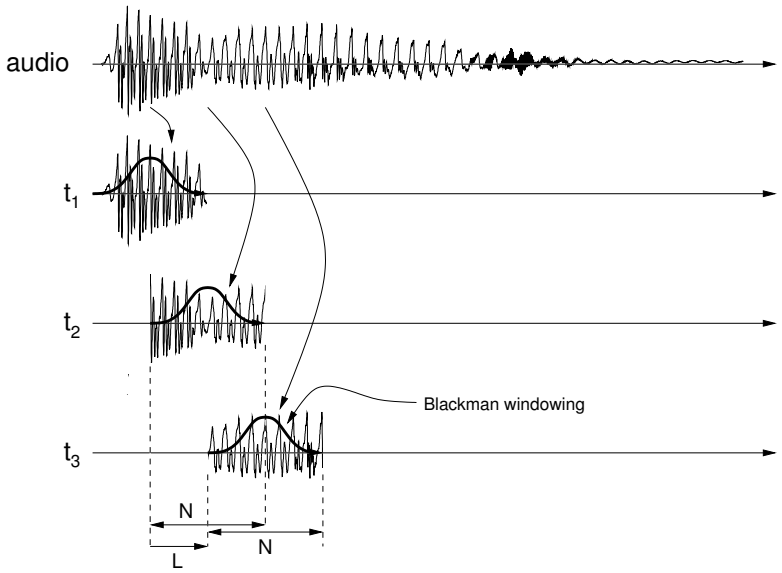


FIGURE 6.18 – L'extraction de caractéristiques est basée sur une fenêtre glissante. Une fenêtre de Blackman est utilisée pour éviter les effets de troncature. N est le nombre d'échantillons sonores dans la fenêtre, et L le délai (en termes d'échantillons) entre deux fenêtres consécutives.

$$s[n]_{n\in[0,255]} \xrightarrow{FFT} S[j]_{j\in[0,255]} \xrightarrow{\text{MEL}} S[k]_{k\in[0,15]} \xrightarrow{\log|\,|} \log|S[k]|_{k\in[0,15]} \xrightarrow{\text{IDCT}} c[i]_{i\in[0,15]}$$

FIGURE 6.19 – Calcul des MFCC.

La réduction de dimensionalité (de N à 16) vient du filtrage du spectre. Le banc de filtres MEL consiste en un ensemble de 16 filtres triangulaires linéairement espacés sur l'échelle MEL. Ceci est résumé sur la Figure 6.20.

Une transformation *logarithmique* suivie d'une transformation en cosinus discrets inverse ou *Inverse Discrete Cosine Transform* (IDCT) fournit finalement le vecteur de coefficients MFCC ($c[i]_{i\in[0,15]}$) pour la fenêtre courante.

L'idée à la base de l'analyse cepstrale (aussi appelée transformation homomorphique) est de tirer parti des propriétés de la transformée de Fourrier et du logarithme pour transformer un produit de convolution en une somme. En effet, le signal de parole peut être approximativement modélisé par la convolution du signal glottique (pression d'air à la sortie de la glotte) et de la réponse impulsionnelle du conduit vocal. Après la transformée de Fourrier, cette convolution devient un produit simple. Après la transformation logarithmique, le produit simple se transforme en la somme des contributions du conduit vocal et du signal glottique. Ce qui nous intéresse réside dans la contribution du conduit vocal qui est caractéristique du locuteur (cela contient une information sur la forme du conduit vocal). Les coefficients cepstraux sont représentatifs de cette contribution et cette transformation permet l'extraction plus simple de la contribution du conduit vocal (soustraire est plus simple que déconvoluer).

6.3.3 Détection des changements de locuteurs

La division du flux audio en segments peut être réalisée de diverses manières (voir [6], Section 2.2). Dans ce travail, nous avons choisi une méthode basée sur le calcul d'une distance entre deux ensembles de fenêtres contiguës comme ce que nous avions fait dans [187]. Les frontières des segments sont alors identifiées comme les maxima de cette distance. Pour calculer cette distance, deux fenêtres glissantes sont définies sur les vecteurs acoustiques (souvenons-nous que chaque vecteur acoustique est lui-même

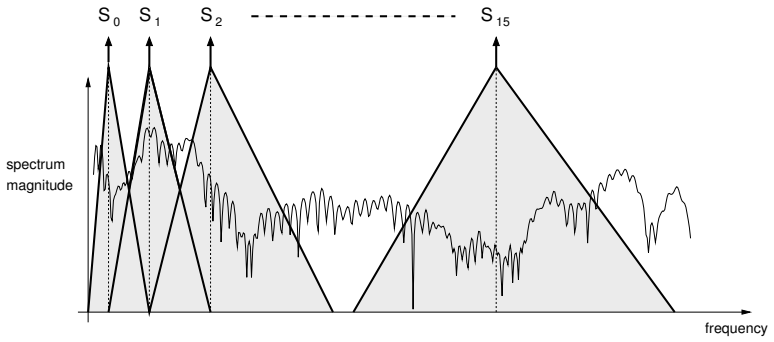


FIGURE 6.20 – Banc de filtres MEL

obtenu en traitant une fenêtre d'échantillons sonores toutes les 10 ms). De manière à faire un traitement statistiquement significatif, chaque fenêtre doit contenir à peu près 300 vecteurs acoustiques, ce qui représente 3s de parole. Ici, il n'y a pas de recouvrement entre les fenêtres (voir Figure 6.21). Notons que cela veut simplement dire que les locuteurs parlent chacun au moins 3s d'affilées.

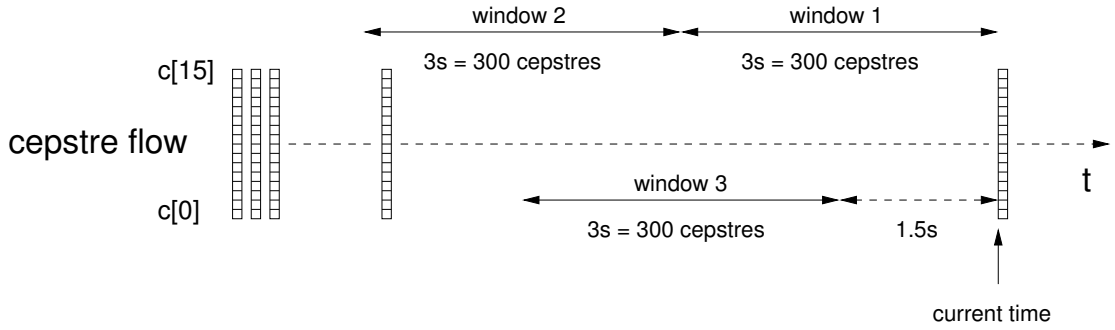


FIGURE 6.21 – Les changements de locuteurs sont détectés par le calcul d'une distance entre deux fenêtres consécutives (1 et 2 sur la figure). Une troisième fenêtre est aussi utilisée pour calculer les statistiques du vecteur aléatoire à 16 dimensions (*i.e.* la matrice Σ_{win3} utilisée dans la définition des distances).

De telles distances doivent prendre en compte les variabilités naturelles des coefficient MFCC. En effet, la figure 6.22 montre que les coefficients MFCC ont une variance très différente les uns des autres. Donc par exemple, une grande variation dans la valeur du premier coefficient est plus "*normale*" qu'une grande variation dans le huitième coefficient. Il est donc important de pondérer l'influence de chacune des composantes en fonction de cette variation intrinsèque sinon seulement les premiers coefficients seront réellement pris en compte dans le calcul de la distance. Notons que cette distance statistique ne nécessite toujours aucun modèle *a priori* des locuteurs.

Ces considérations conduisent naturellement à la définition de la distance euclidienne normalisée définie par l'équation 6.13, dans laquelle $\Sigma_{\text{win3}} = \{\sigma_{ij}\}_{0 \leq i,j \leq 15}$ est une matrice diagonale où σ_{ii} est la variance de la $i^{\text{ème}}$ composante du vecteur cepstral. Il est aussi possible de prendre en considération la corrélation entre les coefficients en utilisant la distance de Mahalanobis. Elle consiste en l'utilisation de la matrice Σ_{win3} qui est maintenant la véritable matrice de covariance comme l'indique l'équation 6.13. D'autres distances ou même des mesures de divergence ou de similarité peuvent être utilisées comme discuté dans [13].

$$d^2\left(\bar{c}_{\text{win1}}, \bar{c}_{\text{win2}}\right) = \frac{1}{16}\left(\bar{c}_{\text{win2}} - \bar{c}_{\text{win1}}\right)^{\text{T}} \Sigma_{\text{win3}}^{-1}\left(\bar{c}_{\text{win2}} - \bar{c}_{\text{win1}}\right) \tag{6.13}$$

De manière à garder le caractère "en ligne" du processus et d'éviter les dérives statistiques dues aux

changements de locuteurs, Σ est estimée sur le passé récent du signal contenu dans une troisième fenêtre qui recouvre les deux précédentes (voir Figure 6.21).

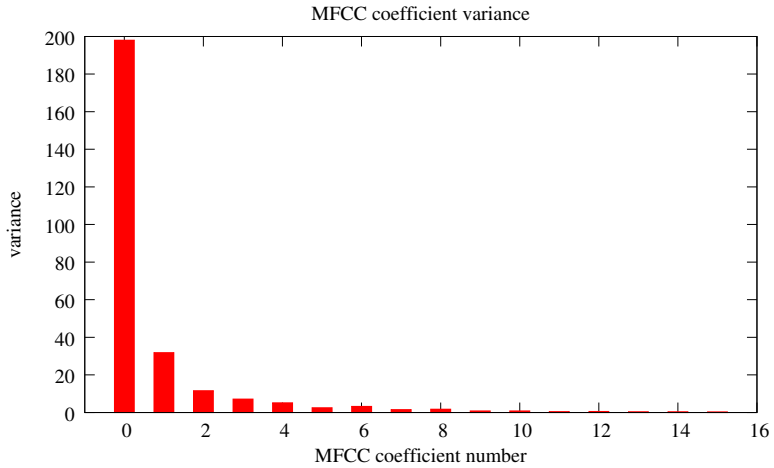


FIGURE 6.22 – Variance des coefficients MFCC coefficient. La variance décroît de manière importante avec l'ordre du coefficient. C'est pourquoi les coefficients MFCC sont normalisés en fonction de leur variance pour le calcul de la distance.

Le calcul décrit précédemment et illustré sur la Figure 6.21 est réalisé toutes les 500ms (*i.e.* le temps courant indiqué dans la figure est translaté de 50 vecteurs acoustiques). Ce processus génère donc un signal de distance (d^2) qui est échantillonné toutes les 500ms dont il faut détecter les maxima en ligne. Le signal de distance étant tout de même assez bruité, un filtre passe-bas lui est appliqué pour le lisser et éviter des fausses détections (voir Figure 6.23) :

$$d^f_{i+1} = d^f_i + \lambda(d^2_{i+1} - d^f_i),$$

où d^f_i est la distance filtrée au temps i, d_i est la distance brute au temps i et λ est le paramètre de lissage (plus λ est faible, plus la distance est lissée).

La Figure 6.23 montre que les maxima absolus ne peuvent pas être détectés en localisant simplement les maxima locaux du signal de distance. Cela reste vrai même après le filtrage passe-bas. Une procédure d'identification des maxima les plus pertinents doit être mise en place tout en respectant les contraintes de temps réel et de calcul en ligne.

La détection de maximum est alors réalisée sur base de calcul de seuils. Une fenêtre glissante de taille fixe (9 échantillons ici, c'est à dire 4 s de parole) est utilisée sur les derniers échantillons du signal de distance filtré passe-bas d^f. Si l'échantillon maximal dans cette fenêtre est à la position centrale, il est étudié plus précisément. Cet échantillon central est considéré *a priori* comme un maximum correspondant à un changement de locuteur si :

- son amplitude M est plus importante qu'un seuil β.
- il y a au moins m échantillons avant l'échantillon central et m échantillons après dont l'amplitude est en-dessous d'un seuil αM. Dans ce travail $m = 1$ procure de bons résultats.

Comme mentionné précédemment, la détection d'un changement de locuteur reste d'une complexité computationnelle faible parce que cela doit se faire rapidement et parce que notre contribution réside plus dans la méthodes d'étiquetage expliquée ci-après. De plus, les fausses détections peuvent être récupérées dans certaines mesures par la méthode de regroupement en fusionnant deux segments consécutifs ont été prononcés par la même personne.

6.3.4 Regroupement et étiquetage

La partie innovante de notre travail réside dans cette partie, spécifiquement parce que les calculs sont fait en temps réel, en ligne et de manière non-supervisée. Cela implique de gérer des flux de données. Le

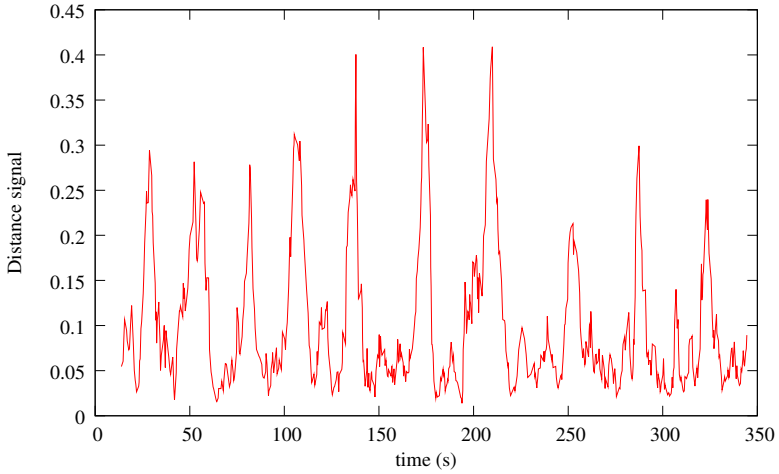


FIGURE 6.23 – Le signal de distance est généré par le procédé décrit dans la figure 6.21. Après avoir filtré passe-bas, les instants où se situent les pics importants deviennent des bons candidats pour des transitions entre locuteurs. Ces transitions peuvent en fait apparaître en réalité jusqu'à 3s avant le maxima du fait des délais induit par la fenêtre 1 sur la Figure 6.21.

locuteur courant est caractérisé par la distribution courante des vecteurs acoustiques qu'il produit. Cette distribution doit être capturée de manière à ce que la comparaison entre deux représentations permette de déterminer si elles correspondent au même locuteur ou pas. Nous basant sur des travaux précédents de notre collègue Hervé Frezza-Buet dans le domaine du traitement de flux vidéo [66], pour lesquels les distributions des pixels intéressants sont hautement non-stationnaires, une procédure a été définie pour obtenir une telle représentation dynamiquement. Une caractéristique cruciale qui fournit à l'algorithme son caractère dynamique est qu'il se base sur des graphes représentant la distribution comme indiqué dans la Section 6.1.2. Ce graphe est adapté pendant que la distribution change. Le résultat de cette adaptation est un ajustement automatique du nombre de prototypes de manière à préserver la qualité de la représentation de la distribution sous-jacente alors qu'elle change.

L'algorithme GNG-T, décrit dans la Section 6.1.3, est utilisé pour représenter la distribution des vecteurs cepstraux entre deux segments consécutifs qui ont été détectés par le processus de segmentation présenté précédemment (voir figure 6.24). La flexibilité de GNG-T évite de devoir paramétrer à la main le nombre de prototypes requis. Ceci est compatible avec la contrainte d'un traitement en ligne, non-supervisé et sans modèle.

Dans [66], GNG-T a été appliqué à l'analyse de flux vidéo, échantillonnant les contours des images au fil du temps. Dans un tel contexte, une époque [8] $\mathcal{E}_t$ est une image de la vidéo au temps t, et tous les ξ_i^t sont les pixels appartenant aux contours dans cette image. Dans le contexte d'analyse de parole qui nous intéresse, l'échantillonnage au temps t concerne seulement un échantillon.

Quantification des vecteurs acoustiques avec GNG-T

Une époque $\mathcal{E}_t$ est un ensemble d'échantillons tirés de la distribution p_t. Comme indiqué dans [66], le nombre réel d'exemples dans une époque reflète l'intégrale de la distribution sur l'espace des échantillons X. Grossièrement, une augmentation du nombre d'échantillons dans une époque $\mathcal{E}_t$ signifie qu'une partie de l'espace X est devenu plus dense, au sens de p_t.

Dans le contexte du regroupement de vecteurs acoustiques, les choses diffèrent légèrement. Au temps t, un nouveau vecteur de MFCC $\xi_t \in X = \mathbb{R}^{16}$ est disponible, mais un exemple isolé ne peut pas former une époque pour échantillonner une distribution p_t. En effet, la distribution des vecteurs de MFCC qui doit être quantifiée est celle du locuteur courant, quelle que soit la longueur du segment

8. voir Section 6.1.3 pour la définition d'une époque et d'un *chunk*

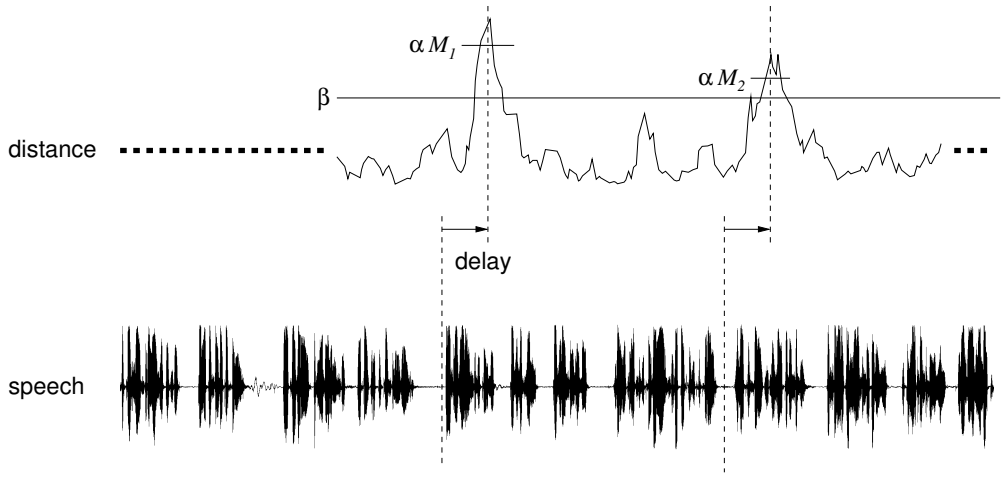


FIGURE 6.24 – Détection de maximum à base de seuils. Les processus successifs de fenêtrage et de filtrage impliquent un délai dans la détection (6 s avec nos paramètres expérimentaux).

de parole qu'il prononce. Supposons que des échantillons contigus de t_a à t_{a+n} correspondent à un unique locuteur, la collection des n veceurs MFCC $\{\xi_{t_a} \cdots \xi_{t_{a+n}}\}$ sont en fait les échantillons qui doivent être quantifiés pour caractériser le locuteur, puisque la distribution des vecteurs de MFCC est supposée stationnaire pour ce locuteur. Ainsi, la collection d'échantillons peut être utilisée comme époque, la longueur réelle de la collection ne peut plus être interprétée comme précédemment. Cette longueur reflète maintenant la précision de l'échantillonnage puisque le fait d'avoir de longues époques signifie que nous pouvons disposer de plus d'échantillons pour évaluer la distribution dont ils ont été tirés. Pour prendre cet effet en compte, l'algorithme présenté dans la Section 6.1.3 est adapté comme suit. La valeur `target` qui pilote la précision de la quantification n'est plus constante. Elle est actuellement définie comme $\texttt{target} = n \times \texttt{t}$, la constante `t` étant maintenant un paramètre qui contrôle la précision de la quantification. Les justifications de cette modification peuvent être trouvées dans [66].

Indentification des locuteurs à partir des graphes

Une méthode est proposée pour identifier les locuteurs *en ligne* à partir des graphes obtenus par GNG-T sur les vecteurs MFCC qui sont pris comme une représentation des caractéristiques acoustiques des locuteurs. Rappelons-nous que la détection en ligne des changements de locuteurs est disponible. A chaque instant t, l'ensemble des vecteurs MFCC qui ont été calculés depuis le dernier changement de locuteur est disponible. Tous les τ pas temporels, un mélange de cet ensemble est effectué et utilisé μ fois comme une époque pour GNG-T, et μ' fois pour la stabilisation, en ne considérant pas les étapes marquées d'une $\star$ dans l'algorithme (voir Section 6.1.3). Une valeur de `t` en relation avec la taille de l'ensemble est aussi définie, comme dit dans la Section 6.3.4. Lorsque le changement de locuteur suivant est détecté, le graphe courant est associé au segment temporel situé entre le changement précédent et le changement détecté. Alors, avant que le segment suivant ne commence, l'ensemble de vecteurs est rendu vide ce qui signifie que la seule trace des vecteurs contenus dans le segment qui est conservée par notre méthode est le graphe construit sur ce segment. Ce graphe constitue en effet une représentation compacte de la distribution des MFCC pour le segment et donc pour le locuteur de ce segment.

A partir de cette procédure, les locuteurs dans chaque segment peuvent être identifiés en calculant une distance D entre les graphes conservés pour tous les segments précédemment identifiés. Pour le segment courant, la distance est calculée entre le graphe courant et chacun des graphes correspondant aux segments précédents. Si la distance est plus faible qu'un seuil prédéfini γ, le locuteur du segment courant est considéré comme étant le même que celui dans un des tours précédents. Sinon, le segment courant est associé à un nouveau locuteur.

Il nous faut maintenant définir une méthode de comparaison entre les graphes. Il existe un grand

nombre de métriques pour ce faire dans la littérature [26] et nous en avons essayé plusieurs. Nous ne rapportons ici que le choix qui a fourni les meilleures performances. La méthode choisie consiste à associer chaque nœud d'un graph avec le nœud le plus proche dans l'autre graphe puis de sommer les distances euclidiennes entre les nœuds de chacune des paires ainsi constituées. Cette métrique est commutative. La similarité $D(A, B)$ entre deux graphes correspondant à des locuteurs A et B est donc calculée en utilisant l'équation 6.14, où $|G|$ est le nombre de nœuds du graphe G et $g \in G$ un nœud de G. La distance $d(a, b)$ est la distance euclidienne entre les vecteurs prototypes a et b.

$$D(A,B) = \frac{\sum_{a \in A} \min_{b \in B} d(a,b) + \sum_{b \in B} \min_{a \in A} d(a,b)}{|A| + |B|} \tag{6.14}$$

Notons que les arrêtes de chacun des graphes ne sont pas utilisées pour calculer cette similarité dans l'équation 6.14. Néanmoins, rappelons que les arrêtes jouent un rôle central dans les propriétés dynamiques de GNG-T.

Procédure de regroupement

Comme indiqué dans les sections précédentes, chaque segment est associé à un locuteur grâce à une mesure de distance entre des graphes. Deux segments consécutifs peuvent avoir été étiquetés comme ayant été prononcé par le même locuteur du fait d'une sensibilité trop importante de la procédure de détection des changements de locuteurs. Pour traiter ce type de fausses détections (faux positifs), il est possible de regrouper *a posteriori* des segments consécutifs ayant la même étiquette comme illustré sur la Figure 6.25.

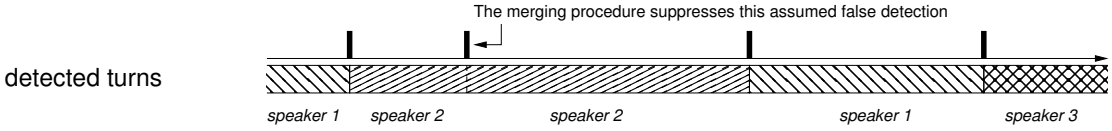


FIGURE 6.25 – Procédure de regroupement pour palier les fausses détections de la procédure liées aux changements de locuteurs.

6.3.5 Expérience

Evaluation de la méthode de détection des changements de locuteurs

Afin de rendre notre expérience reproductible, nous avons constitué des données de test à partir de la base de données BREF80 [120]. Cette base de données contient plusieurs heures de parole lue en français, prononcées par 90 locuteurs différents (50 femmes et 40 hommes). De cette base de données, 11 locuteurs ont été tirés aléatoirement pour constituer un jeu de données artificiel pour tester l'algorithme. Puisque l'algorithme est totalement non supervisé, aucune donnée n'est utilisée pour entraîner le système ce qui est un avantage de la méthode proposée.

Les segments, dont la durée varie de 10 à 40 secondes, ont été aléatoirement sélectionnés dans le contenu audio correspondant aux 11 locuteurs puis concaténés de manière à constituer un fichier audio de 10 heures. La base de données de test contient donc approximativement 1500 segments ($N = 1500$ dans la suite). Plusieurs mesures ont été calculées pour vérifier les performances. Premièrement, pour quantifier les performance de la détection de changement de locuteurs (voir Section 6.3.3), la sensibilité et la pureté ont été calculées comme suit :

$$\text{Sensibilité}(\%) = \frac{TP}{TP + FN} \times 100\% \tag{6.15}$$

$$\text{Pureté}(\%) = \frac{TP}{TP + FP} \times 100\% \tag{6.16}$$

où TP représente le nombre de vrais positifs ou *true positives* (le nombre de changements correctement détectés), FN est le nombre de faux négatifs ou *false negatives* (pas de détection alors qu'il y a un changement) et FP est le nombre de faux positifs ou *false positives* (détection alors qu'il n'y a pas de changement). La Figure 6.26 illustre le calcul de ces valeurs. La valeur optimale pour ces deux valeurs est 100% ce qui est atteint pour des détections correctes (sensibilité = 100%) et des segments longs et homogènes (pureté = 100%).

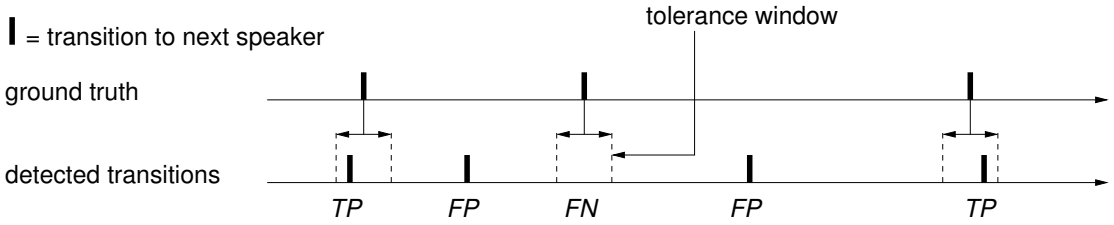


FIGURE 6.26 – Définition de TP (true positive), FP (false positive) et FN (false negative). Le délai induit par le processus de détection (voir Figure 6.24) est enlevé pour calculer les performances. La valeur TP est calculée en fonction d'une fenêtre de tolérance de 6s dans ce cas.

	Sensibilité	Pureté	α	β
Sans regroupement	96.5 %	96.8 %	1	0.13
Avec regroupement	96.5 %	97.4 %	1	0.01

TABLE 6.2 – Résultats de la détection de changement de locuteurs obtenus pour le meilleur jeux de paramètres (voir Figure 6.27). L'effet de la procédure de regroupement décrite dans la Section 6.3.4 est montré.

Les résultats sont fournis dans le Tableau 6.2. Les mesures de sensibilité et de pureté sont très satisfaisantes étant donné la simplicité de la procédure de segmentation. Les seuils optimaux α et β ont été obtenus par *grid search* pour fournir ces résultats qui sont donc représentatifs des meilleures performances qui peuvent être obtenues par la méthode décrite dans la Section 6.3.3. Bien sûr, la nécessité de trouver ces seuils est un point faible de la méthode mais la sensibilité aux paramètres (et particulièrement à ces niveaux) est montrée sur la figure 6.27. Sur cette figure, le f-score, définit par :

$$\text{f-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6.17}$$

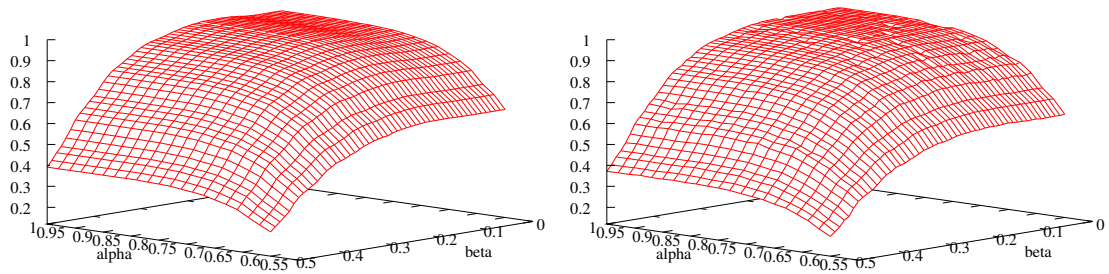


FIGURE 6.27 – f-score (*i.e.* $2 \times \text{Sensibilité} \times \text{Pureté}/(\text{Sensibilité} + \text{Pureté})$) pour la détection des changements de locuteurs. Les figures de gauche et de droite sont respectivement obtenue sans et avec la procédure de regroupement décrite dans la Section 6.3.4.

On peut voir que le f-score est relativement plat autour des valeurs optimales des seuils ($\alpha = 0.85$ et $\beta = 0.15$). Cela montre que les performances de l'algorithme ne sont pas très sensibles aux variations

des valeurs des paramètres. On peut aussi remarquer que la courbe est même plus plate encore lorsque la procédure de regroupement est appliquée, ce qui montre que les sur-segmentations sont compensées correctement par la procédure. On peut voir sur la Figure 6.27 que les résultats sont moins sensibles aux petites valeurs de β lorsque le regroupement automatique est effectué. En effet, sans cette procédure, les performances décroissent quand β est plus petit que 0.13. Cet effet disparaît lorsque le regroupement est effectué.

Notons que la détection des changements de locuteurs est un traitement préalable qui ne sert qu'à alimenter l'algorithme d'étiquetage. Donc une pureté assez faible n'est pas un véritable problème si la procédure d'étiquetage fonctionne correctement. Une condition nécessaire pour que cela soit effectivement le cas est que chaque segment contienne suffisamment de vecteurs acoustiques d'un même locuteur et donc la sensibilité ne devrait pas être trop faible. Une autre condition est que chaque segment ne doit pas contenir des vecteurs acoustiques de différents locuteurs et la pureté devrait être aussi d'un niveau suffisamment élevé.

Evaluation de la méthode d'étiquetage

Plusieurs métriques sont utilisées pour évaluer la méthode d'étiquetage. Premièrement, la mesure de *matching* calcule le pourcentage (en termes de nombres globaux (indexés par N) ou en termes de durée (indexés par T)) de segments correctement étiquetés étant donné la vérité terrain.

$$\text{match}_N = \frac{\text{nombre de segments correctement étiquetés}}{N''} \tag{6.18}$$

$$\text{match}_T = \frac{\text{temps correspondant aux segments correctement étiquetés}}{\text{durée totale du fichier audio}} \tag{6.19}$$

La valeur de N'' est définie sur la Figure 6.28.

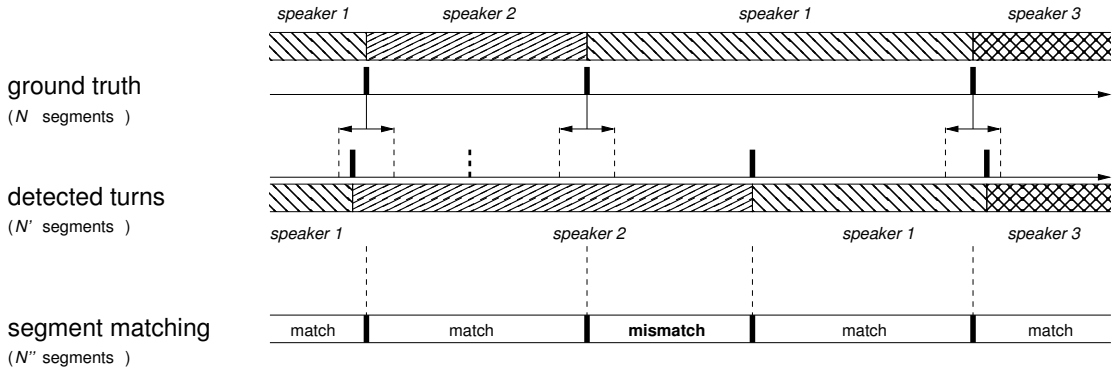


FIGURE 6.28 – Le *matching* des segments détectés avec la vérité terrain est calculé en rassemblant les transitions à la fois de la vérité terrain et de la détection (considérant une fenêtre de tolérance). Les sous-segments générés sont qualifié de "segment correctement étiqueté" si leurs étiquettes sont identiques.

Deuxièmement, l'*homogénéité des segments* h quantifie le fait que les segments correspondent bien à un seul locuteur. Nous introduisons la notion d'*étiquette principale* pour calculer cette valeur. L'étiquette principale d'un segment s est l'étiquette qui a la plus grande proportion du sous-segments correspondant dans la vérité terrain. Ce sous-segment et le segment détecté ont donc la même longueur $l(s)$. Appelons $p(s)$ cette proportion. La véritable étiquette dans un segment détecté peut éventuellement être différente de son étiquette principale si la procédure a échoué. La valeur présentée ci-après est la moyenne de $p(s)$ pour tous les segments détectés.

L'*homogénéité complémentaire* $\bar{h}$ est calculée comme la précédente mesure d'homogénéité mais les rôles de la vérité terrain et de la segmentation automatique sont inversés. L'utilisation de ces deux mesures évite de sur-évaluer les performances en cas de sur-segmentation. En effet, si les segments sont trop petits, ils ont une bonne homogénéité mais le complémentaire est faible.

match$_N$	match$_T$	h	$\bar{h}$	number of speakers	α	β
Sans regroupement						
95.7%	97.6%	99.3%	97.9%	11	0.96	0.05
95.7%	97.6%	99.3%	97.9%	10	0.98	0.02
95.3%	97.2%	99.1%	99.2%	11	1.00	0.13
Avec Regroupement						
94.3%	96.1%	98.7%	99.6%	11	0.94	0.13
94.7%	96.7%	99.1%	99.5%	12	0.92	0.05
94.6%	96.5%	99.2%	99.5%	13	1.00	0.01

TABLE 6.3 – Résultats de l'étiquetage. L'effet de la procédure de regroupement décrite dans la Section 6.3.4 est aussi montré. Il y a 11 locuteurs dans la vérité terrain. Pour chaque cas (avec ou sans regroupement), la première ligne est le meilleur match$_T$ pour lequel le nombre de locuteurs est correct (11), la deuxième ligne est le meilleur match$_T$, quel que soit le nombre de locuteurs trouvé. La troisième ligne est le jeu de paramètres utilisé pour le Tableau 6.2.

Les résultats sont montrés dans le Tableau 6.3. Ces résultats sont très bons (les segments sont pratiquement homogènes et le *matching* est très précis) en comparaison à des algorithmes similaires de l'état de l'art [46] tout en restant en ligne et non-supervisé.

6.3.6 Discussion

Nous avons montré ici que la quantification vectorielle pouvait nous permettre aussi d'effectuer la segmentation en locuteurs d'un flux audio en temps réel sur un ordinateur standard et sans aucune information *a priori* sur les locuteurs.

Le caractère temps réel est atteint parce que les informations concernant les locuteurs sont résumées dans des graphes de petite dimension appris de manière automatique par l'algorithme GNG-T. Les performances sont au moins aussi bonnes que celles que l'on peut trouver dans la littérature sur le sujet depuis une dizaine d'années. Pourtant les contraintes imposées à la méthode proposée sont moins importantes que dans les travaux rapportés jusqu'ici (pas de modèle, segmentation au fil de l'eau, temps réel).

En particulier, nous avons pu maintenir la complexité de l'algorithme de détection des changements de locuteurs relativement simple du fait du bon comportement de la méthode de regroupement basée sur la quantification vectorielle.

6.3.7 Conclusion

Le signal que nous analysons ici est évidemment le propre de l'homme puisqu'il s'agit de la parole (encore que ce point soit discutable, puisque nous ne traitons pas du langage articulé mais du signal acoustique dans ce travail). Il possède encore une fois toutes les caractéristiques d'un signal produit par l'humain (ou du moins par le vivant) et qui impose des contraintes sur les traitements qui peuvent lui être appliqués. Ici, c'est particulièrement la variabilité inter-locuteur qui est importante, c'est-à-dire le fait que la parole produite possède des caractéristiques différentes suivant la personne qui la prononce. Cette variabilité impose une adaptation automatique des algorithmes. Par ailleurs, il est difficile de prévoir l'étendue de cette variabilité, c'est pourquoi il est souhaitable de ne pas essayer de construire des modèles *a priori*. Ainsi, l'apprentissage non-supervisé et particulièrement la quantification vectorielle se trouve bien adaptée. De plus, nous avons dû utiliser une méthode particulière permettant de poursuivre encore une fois la solution et pas de converger vers celle-ci. C'est le cas de la méthode GNG-T.

6.4 Conclusion

Après avoir fait un rappel théorique sur la quantification vectorielle, nous avons montré comment modéliser des problèmes pratiques pour qu'ils puissent se résoudre par ce biais. Nous avons montré, dans les Sections 6.2 et 6.3, que la quantification vectorielle répondait bien aux contraintes d'une certaine classe de problèmes que l'on peut se poser lorsque l'on travaille en relation avec l'humain. L'absence de vérité terrain est le problème principal. Ce manque pourrait être pallier grâce à l'intervention d'experts, mais la Section 6.2 a montré que les experts ne sont pas toujours totalement d'accord entre eux (bien que le désaccord ne porte normalement pas à conséquence sur l'analyse de résultats). De plus, le problème de la Section 6.3 ne pourrait pas être résolu par annotation manuelle sinon l'application serait obsolète. Ainsi, il est important de pouvoir traiter des données sans modèle *a priori* dans un grand nombre d'applications liées au traitement d'information produite par l'humain. En ce sens, l'apprentissage non-supervisé remplit bien sa mission. Aussi, le caractère non-stationnaire de la distribution des données, inhérent à la présence de l'humain, est bien pris en compte par l'algorithme GNG-T développé par notre collègue Hervé FREZZA-BUET et adapté pour nos besoins.

Notons que nous avons aussi apporté une contribution théorique à la quantification vectorielle en proposant une méthode permettant à tout algorithme de quantification vectorielle de profiter des méthodes à noyaux [96, 98]. Ces contributions sont assez originales et permettent de segmenter un espace en cellule de Voronoï de forme assez complexe suivant le noyau choisi.

Troisième partie

Projet de Recherche

Chapitre 7

Vers des systèmes interactifs situés

Dans ce manuscrit, nous avons rapporté les travaux réalisés durant les six dernières années dans les domaines du contrôle optimal et du traitement du signal. Particulièrement, nous nous sommes intéressé à répondre aux contraintes imposées par la présence de l'humain dans les environnements que nous cherchons à analyser ou à contrôler. De ce fait, certaines pistes de solutions ont été privilégiées et particulièrement le point de vue de l'apprentissage automatique qui permet de s'affranchir en grande partie du besoin de connaissances et de modèles *a priori* tout en laissant la possibilité d'introduire ce type de connaissance (au travers des bruits pour les méthodes de filtrage, des structures d'approximation pour l'estimation de paramètres ou du nombre de prototypes dans les application de quantification vectorielle). Nous avons aussi privilégié les pistes permettant de poursuivre les solutions plutôt que de converger vers des solutions globales de manière à tenir compte de la nature non-stationnaire des signaux et environnements considérés.

Un des grands enjeux de l'informatique de demain est, à notre sens, de faire sortir les systèmes intelligents des laboratoires pour les ancrer dans la réalité, se confrontant ainsi avec l'humain dans son quotidien. Pour ce faire, plusieurs verrous sont à lever parmi lesquels l'autonomie, l'adaptativité et le passage à l'échelle sont les plus importants. Ce sont des verrous que nous espérons contribuer à lever en adoptant une approche située dans laquelle la perception, la décision et l'action sont pensées ensemble relativement à la tâche et à l'interaction avec le monde. Dans les travaux que nous souhaitons mener à l'avenir, nous proposons de nous attaquer à ces verrous de manière concrète et traitant les points suivants :

Systèmes récompensés Nous souhaitons explorer de manière différente les systèmes récompensés que nous avons abordés sous l'angle des Processus Décisionnel de Markov (PDM) jusqu'à présent. Notamment, nous allons étudier le transfert de tâche par le biais de l'apprentissage par renforcement inverse et travailler sur la relaxe de la contrainte de Markov.

Adaptation Poursuivant nos études actuelles sur le filtrage bayésien notamment, nous allons nous concentrer sur les méthodes permettant de gérer la non-stationnarité des environnements en traquant les solutions et non en convergeant vers elles ainsi que de traiter la stochasticité en gérant l'incertitude pour explorer de manière active l'environnement et les perceptions.

Problèmes de grande taille Pour traiter des problèmes de grande taille, il est nécessaire de faire une représentation compacte et évolutive des informations (il faut apprendre mais aussi oublier). Ainsi, nous comptons travailler sur la recherche en ligne de structures d'approximation parcimonieuses, pilotée par la tâche via la récompense. Ceci nécessite en général de traiter de fortes non-linéarités.

7.1 Travail en cours et perspectives à court terme

Nos travaux en cours poursuivent les travaux présentés dans ce manuscrit et conservent leur caractère multidisciplinaire. Ainsi, nous poursuivons nos recherches dans les domaines du contrôle optimal ainsi que du traitement du signal tout en leur applicant le point de vue l'apprentissage automatique. Particulièrement, nous nous focalisons sur des problèmes non-stationnaires et non-linéaires. Etant donné notre

implication dans la communauté du dialogue homme-machine, cette thématique sera aussi poursuivie mais étendue à d'autres domaines que ceux traités jusqu'à présent, en particulier, les systèmes éducatifs.

7.1.1 Apprentissage par renforcement

Dans la Partie I, nous avons exposé une solution originale au problème de l'Apprentissage par Renforcement (AR) pour des environnements non-stationnaires avec les différences temporelles de Kalman ou *Kalman Temporal Differences* (KTD) développées dans le cadre de la thèse de Matthieu GEIST [71]. Nous avons la chance d'avoir pu embaucher Matthieu GEIST dans l'équipe IMS et nous espérons ainsi pouvoir continuer avec lui à développer la thématique de l'apprentissage par renforcement. Particulièrement, nous voulons améliorer encore le caractère générique et adaptatif des algorithmes proposés.

KTD permet un apprentissage *off policy* et donc peut utiliser une équation non-linéaire incluant un opérateur $\max$ (équation d'optimalité de Bellman) qui est de plus non-dérivable. C'est en utilisant une transformation statistique particulière (transformation non parfumée) que nous avons pu résoudre ce problème dans le cadre du filtrage de Kalman. Nous avons opté pour une représentation paramétrique de la fonction de valeur, ce qui nécessite de se fixer *a priori* la structure d'approximation et les variables d'état pertinentes pour résoudre le problème. Pourtant, ces choix *a priori*, bien que nous les ayons choisi avec soins, peuvent avoir une influence sur la qualité de la politique de contrôle apprise. De plus, si nous n'y prenons pas garde, le nombre de paramètres nécessaires à la représentation des fonctions de valeur peut devenir prohibitif même pour des applications simples (par exemple, plus de 300 paramètres pour un systèmes dont l'espace d'état est composé de 3 dimensions continues seulement). Nous travaillons donc actuellement sur des extensions de notre cadre de travail permettant toujours d'utiliser des représentations paramétriques non-linéaires et de traiter l'opérateur $\max$ mais en plus d'apprendre la structure optimale d'approximation. Nous nous orientons vers l'utilisation de méthodes de régularisation en norme $L1$ ou $L0$ ou de méthodes venant du *compressed sensing* pour ce faire. Des résultats préliminaires encourageants ont déjà été obtenus mais sont en cours de validation.

Nous cherchons aussi à généraliser le cadre KTD en observant que la méthode développée est en fait un cas particulier de linéarisation statistique autour de l'estimation en cours des paramètres que nous cherchons. Ce point de vue, adopté dans [82, 80, 81], nous a permis d'étendre l'algorithme *Least Square Temporal Differences* (LSTD) [20], connu pour être particulièrement efficace pour évaluer une politique à partir d'un jeu de données fixe, à l'utilisation de structures d'approximation compactes comme les réseaux de neurones mais aussi au cas de l'apprentissage *off-policy*. Ce résultat laisse présager d'améliorations significatives de l'état de l'art dans le domaine puisque le passage à l'échelle de ces algorithmes est limité par le caractère linéaire des structures d'approximation aujourd'hui obligatoire.

Aussi, nous cherchons à utiliser des schémas d'apprentissage acteur-critique [242] avec approximation de la fonction de valeur [78, 79]. Ces schémas permettent de s'affranchir du calcul de l'argmax sur la Q-fonction pour obtenir l'action optimale. En effet, réaliser un argmax peut devenir un problème lorsque nous traitons des problèmes avec actions continues (cela nécessite des descentes de gradients complexes suivant la structure d'approximation choisie pour la Q-fonction). Ces schémas nécessitent souvent des structures d'approximation linéaire du fait de la condition de compatibilité entre la représentation de la Q-fonction et de la politique [167]. C'est pourquoi nos travaux s'orientent pour l'instant vers la recherche d'algorithmes apprenant automatiquement des structures localement linéaires (en relation avec les LWPR (*Locally Weighted Projection Regression*) [251]).

7.1.2 Interaction homme-machine

Après le projet européen CLASSiC sur le dialogue homme-machine, qui est pour nous une application importante des méthodes efficaces d'AR que nous développons, nous participons à un nouveau projet sur un thème assez proche. Il s'agit du projet ALLEGRO, en collaboration avec l'Université de Saarbruck, le DFKI et l'INRIA Nancy-Grand Est, dont l'objectif est de créer un système de dialogue interactif pour l'apprentissage des langues. Dans ce projet, nous sommes en charge d'optimiser le choix des exercices permettant à l'apprenant d'acquérir des compétences linguistiques et à la machine d'évaluer l'acquisition de ces compétences.

Cette application est d'un type particulier puisqu'il ne s'agit pas d'un système de dialogue tel que nous les avons traités jusqu'à présent. En effet, il ne s'agit pas de fournir un service ou une information précise mais de constituer un parcours pédagogique adapté aux compétences de l'utilisateur. Ainsi, la modélisation de la tâche, mais aussi de l'utilisateur, seront totalement différentes des applications où l'objet est d'obtenir une information touristique. L'interaction sera beaucoup plus longue et les performances (ou récompenses) mesurées tout au long de l'interaction et pas uniquement à la fin. L'espace d'états risque d'être d'une taille très importante avec des composantes continues et discrète (système hybride). Les actions possibles seront elles aussi en très grand nombre. Il existe des exemples académique de systèmes automatiques dédiés à la pédagogie en ligne [5], néanmoins leur réalisme assez faible est loin de permettre une utilisation en ligne réelle.

Dans le cadre de ce projet, plusieurs pistes intéressantes seront investiguées, outre l'utilisation de méthodes efficaces d'AR. Il est par exemple connu qu'un système éducatif doit présenter des exercices de difficulté croissante à l'apprenant sans passer des paliers trop importants à chaque étape, sous peine de décourager les élèves. Nous entendons utiliser l'incertitude que nous pouvons calculer sur les différentes fonctions de valeur estimées pour explorer l'espace d'état-action de manière sécurisée pour éviter de passer ces paliers trop important justement. Aussi, nous voulons tirer parti du fait que l'application qui est visée est un service Internet. Il est donc très probable que plusieurs utilisateurs seront connectés en même temps. De ce fait, il doit être possible de ne pas faire supporter à un même élève tout l'aléa dû à l'exploration de l'espace d'état-action. Ainsi, des méthodes efficaces de répartition de cette exploration devront être conçues.

Aussi, nous participerons au projet FET OPEN ILHAIRE dont le but est de concevoir des interfaces homme-machine incorporant le rire pour rendre l'interaction plus naturelle. Ici, nous chercherons non seulement à optimiser les interfaces mais aussi à imiter l'humain.

7.1.3 Traitement de signaux physiologiques

La suite de la thèse de Julien OSTER [154] va s'organiser dans le cadre d'un post-doctorat effectué à l'Université d'Oxford (Royaume-Uni). Julien OSTER a en effet obtenu une bourse *Newton International* de la *Royal Society* pour poursuivre ses travaux à partir de mars 2011. Un programme de travail dans lequel nous sommes impliqués a été établi et nous permettra de poursuivre nos travaux sur le débruitage de signaux électrocardiogramme (ECG) dans l'environnement Imagerie par Résonance Magnétique (IRM) par des méthodes de filtrage bayésien, dans la ligne droite des travaux présentés dans le Chapitre 5.

Particulièrement, les modèles de signaux ECG que nous avons développés ne tiennent pas compte de tous les effets indésirables induits par l'imageur IRM dans l'acquisition du signal ECG. Par exemple, l'effet MagnetoHydroDynamique (MHD) n'est pas pris en compte ce qui empêche un traitement correct de l'ECG pour des patients arythmiques dans l'IRM. Aussi, nous aimerions investiguer plus avant les méthodes de détection du complexe QRS dans l'IRM pour des patients atteints de pathologies cardiaques, ce qui concerne évidemment la plupart des patients pour lesquels un examen IRM est requis. En effet, nos modèles supposent des cycles cardiaques plus ou moins réguliers et des complexes QRS de physionomie standard. Rien n'empêche de les étendre à des signaux de nature plus atypique mais cela n'a pas été tenté jusqu'à présent. Ce post-doctorat nous donnera l'occasion de continuer à nous intéresser à ces problèmes de modélisation et d'estimation de paramètres dans un cadre international en collaboration avec l'Université d'Oxford mais aussi avec la *Harvard Medical School*.

7.2 Perspectives à moyen terme

A moyen terme, nous prévoyons de nous intéresser plus précisément aux systèmes récompensés qui nous paraissent être une piste essentielles vers la conception de système situé. Nous conserverons entre autres l'application aux interactions homme-machine (dialogue, intelligence ambiante et robotique). Bien que nous ayons proposé un certain nombre d'algorithmes pour l'AR et envisageons de les améliorer dans un proche avenir, il est important de se souvenir que ce type d'apprentissage pose un problème plus large que celui que nous avons tenté de résoudre. Ce problème est celui de la recherche d'un comportement optimal pour les systèmes récompensés, sans autre contrainte que celle de trouver la séquences d'actions

optimales pour un agent sachant qu'il pourra observer plus ou moins directement les effets de ses actions. Nous nous sommes jusqu'ici placés dans le cadre mathématique des PDM pour résoudre ce problème mais ceci revient à imposer des contraintes le rendant moins ardu. Nous chercherons à dépasser le cadre des PDM dans deux directions qui sont la recherche de la récompense qui n'est pas toujours simple à définir et la relaxation de la contrainte imposée par la propriété de Markov.

Nous souhaitons par ce biais nous impliquer d'avantage encore dans la modélisation de l'humain et dans la prise en compte de sa présence pour la conception de systèmes interactifs au sens large (*i.e.* tout système devant prendre en compte les réactions de l'humain à ses actions pour aider ou collaborer avec l'humain).

7.2.1 Systèmes récompensés

Bien que nous ayons proposé un certain nombre d'algorithmes et envisageons de les améliorer dans l'avenir, il est important de se souvenir que l'AR pose un problème plus large que celui que nous avons tenté de résoudre. Ce problème est celui de la recherche d'un comportement optimal pour les systèmes récompensés, sans autre contrainte que celle de trouver la séquence d'actions optimales pour un agent sachant qu'il pourra observer les effets de ses actions et recevoir une récompense immédiate associée à ces effets. Nous nous sommes placés dans le cadre mathématique des PDM pour résoudre ce problème mais ceci revient à imposer des contraintes rendant le problème moins ardu. Nous aimerions dépasser ce cadre de travail pour nous intéresser à des problèmes plus réalistes et surtout génériques.

Apprentissage par Renforcement Inverse (ARI)

Dans le paradigme de l'AR, un agent apprend une stratégie optimale d'interaction avec un environnement en maximisant un cumul de récompenses immédiates que l'environnement lui fournit après chaque interaction. Dans la pratique, c'est le concepteur du système qui définit la récompense, souvent de manière subjective, ce qui biaise les solutions. Il serait plus justifié d'observer un autre agent (un humain par exemple) essayant avec ses moyens d'accomplir une tâche et d'en déduire ce qui est récompensé. On peut ainsi espérer transférer la tâche à un autre agent (lui apprendre cette tâche), via la fonction de récompense qui en serait la représentation la plus compacte et porteuse de sens. C'est le paradigme de l'Apprentissage par Renforcement Inverse (ARI). Une fois la fonction de récompense découverte, l'agent peut lui aussi optimiser son comportement relativement à cette fonction en utilisant ses propres moyens (qui peuvent être différents de ceux de l'expert comme c'est le cas pour un robot observant un humain). Jusqu'à présent, la littérature assez maigre sur le sujet se focalise sur l'imitation pure alors que le problème de base est d'agir de manière optimale pour atteindre un but (potentiellement mieux ou avec d'autres moyens que l'expert). Il s'agit en fait d'un problème mal posé et des contraintes doivent être ajoutées pour trouver une solution unique. Ces contraintes mènent souvent à des solutions de type programmation linéaire ou quadratique et impose l'imitation. Nous pensons travailler sur des méthodes différentes, introduisant des contraintes de parcimonie de la représentation de la récompense (du type régularisation $L1$ ou $L0$) et n'impliquant pas l'imitation pure. De plus, les algorithmes courant reposent sur l'hypothèse qu'un simulateur ou un modèle précis de la dynamique de l'environnement sont disponibles ce qui est rarement le cas. Nous avons donc commencé à travailler sur des méthodes hors ligne et sur la recherche de bornes d'erreurs impliquées par ces méthodes à échantillons finis. Les applications sont nombreuses notamment pour la robotique de service où une communauté pourrait s'échanger des comportements appris à des robots.

Propriété de Markov

Les algorithmes et les perspectives à court terme envisagés dans le cadre de l'AR partent du postulat que les états du système à contrôler sont totalement observables. Ainsi nous n'avons pas traité les Processus Décisionnel de Markov Partiellement Observable (PDMPO). C'est un des cas important où le système ne peut pas être considéré comme Markov d'ordre 1 en les observations (même s'il l'est en des états cachés dont dépendent ces observations). Ces problèmes sont particulièrement répandus dans la pratique. C'est par exemple le cas d'un robot qui ne peut se diriger que grâce à des caméras. S'il ne

connaît pas sa position exacte (avec un système de géolocalisation GPS par exemple), il pourra éviter des obstacles mais ne pourra pas se diriger vers un but précis dans un bâtiment puisqu'il sera incapable de planifier une trajectoire si le but n'est pas directement visible. Néanmoins, si le robot connaît tout l'historique de sa trajectoire, il peut calculer sa position actuelle. Ainsi, le problème n'est pas Markov en les observations (besoin de l'historique des observations) mais peut le rester en les états (les positions absolues du robot).

Une des perspectives que nous voulons creuser est l'utilisation de méthodes issues du *reservoir computing* [106]. Ces structures particulières de réseaux de neurones sont capables d'encoder de manière automatique des trajectoires ou des dynamiques à partir d'observations séquentielles. Cet encodage se lit au niveau des connexions qui se créent dans un réservoir comprenant un grand nombre de neurones. Il est donc naturel de penser que ce type de réseau permettrait de rendre Markov d'ordre 1 un problème qui serait naturellement Markov d'ordre N. Ceci a déjà été étudié [105] mais la méthode résulte tout de même en une explosion combinatoire des paramètres et nous cherchons à rendre parcimonieuse cette représentation paramétrique. Ceci donne lieu à des contacts ponctuels avec le LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur).

7.2.2 Modélisation de l'humain pour la simulation

La modélisation de l'humain pour en reproduire le comportement est au cœur de nos recherches depuis la thèse. En effet, dans le cadre des systèmes de dialogue hommme-machine, nous nous sommes intéressé à l'extension de bases de données pour l'apprentissage de stratégies de dialogue, du fait de l'inefficacité des algorithmes d'AR communément utilisés pour cette tâche. Dans ce type d'application, un corpus de données, annotées ou non, est disponible. Un modèle statistique de l'utilisateur humain est conçu et entraîné sur ces données. Ainsi, le modèle statistique peut reproduire un comportement similaire à celui des utilisateurs présent dans la base de données. Comme nous l'indiquions dans la Section 4.1, nous pensons que, pour ce type d'applications, des méthodes d'AR plus efficaces en termes d'échantillons gagneraient à être utilisées pour optimiser directement sur les données la stratégie du système. Nous en avons même fait la démonstration sur une application standard. Ceci tendrait donc à prouver que la modélisation de l'utilisateur humain dans le but de simuler son comportement n'est plus nécessaire pour apprendre des stratégies d'interaction dans le cadre spécifique du dialogue homme-machine.

En fait, c'est plutôt la conception que nous avions jusqu'ici de la modélisation de l'utilisateur humain (ou plus précisément de la simulation d'utilisateur) qui est probablement obsolète. En effet, jusqu'à présent, comme tous les acteurs de ce domaine, nous avons considéré que les simulateurs d'utilisateur avaient pour but de reproduire des données statistiquement cohérentes avec celles contenues dans des corpora annotés de dialogue homme-machine ou même humain-humain. Ainsi, il n'est pas étonnant que les informations contenues dans les données soient suffisantes pour apprendre directement une stratégie correcte puisque la simulation n'a pour but que d'augmenter artificiellement le volume des données sans apporter vraiment de nouvelle information.

Néanmoins, nous pensons que la simulation d'utilisateur (pour le dialogue homme-machine ou pour d'autres applications d'ailleurs) reste un domaine de recherche à creuser. En effet, un aspect important est oublié dans la recherche actuelle sur ce domaine : la faculté d'adaptation de l'humain pour atteindre son but. En effet, lorsqu'un simulateur d'utilisateur est utilisé pour optimiser un système interactif comme un système de dialogue, il reproduit des comportements qui étaient ceux de l'utilisateur lorsqu'il interagissait avec le système utilisé pour collecter les données. Lors de l'optimisation de la stratégie d'interaction, le système évolue bien entendu au fur et à mesure de l'optimisation. Rien ne dit pourtant que l'utilisateur continuerait à réagir de la même manière que lorsqu'il était face au système initial. C'est pourquoi nous voulons prolonger nos recherches dans ce domaine en adoptant un nouveau point de vue unifié de l'apprentissage de stratégie et de la simulation d'utilisateur. En effet, si l'optimisation de la gestion de l'interaction peut être interprétée en termes d'AR, nous pensons que le problème de la simulation d'utilisateur peut se poser en terme d'ARI [219].

Dans le paradigme de l'AR, un agent apprend une stratégie optimale d'interaction avec un environnement en maximisant un cumul de récompenses immédiates que l'environnement lui fournit après chaque interaction. Dans le paradigme de l'ARI, l'agent observe un expert (ou mentor) qui agit de manière optimale dans l'environnement et doit en déduire la fonction de récompense qui est maximisée par l'expert

pour agir. Une fois cette fonction de récompense découverte, l'agent peut lui aussi optimiser son comportement relativement à cette fonction et agir de manière optimale comme l'expert.

L'utilisation de l'ARI dans le cadre des systèmes de dialogue homme-machine a été proposé par [165]. Néanmoins, les auteurs proposent de partir de données collectées entre un utilisateur humain et un opérateur humain lui aussi et d'ensuite utiliser l'ARI pour apprendre la stratégie utilisée par l'opérateur humain et l'inclure dans le système de gestion de dialogue homme-machine en considérant que l'opérateur humain utilise une stratégie optimale. En fait, nous pensons qu'il y a plusieurs inconvénients à cette proposition, qui par ailleurs n'a jamais été testée. Le premier problème est que rien ne dit que l'opérateur humain utilise une stratégie optimale. Dans la plupart des applications réelles, l'opérateur humain ne fait qu'utiliser un arbre de décision qui a été conçu pour lui par des experts de la tâche à réaliser. Un système d'ARI ne ferait donc qu'apprendre cet arbre de décision. Quoi qu'il en soit, même si l'opérateur humain pouvait agir librement, rien ne dit que l'optimalité en terme de satisfaction de l'utilisateur serait atteinte. De plus, il est connu que les utilisateurs humains réagissent de manière totalement différentes lorsqu'ils interagissent avec un opérateur humain ou un système automatique [252].

Nous proposons donc une vision alternative du problème. Si l'opérateur humain peut ne pas agir de manière optimale, nous pensons que l'utilisateur humain essaye en revanche d'optimiser inconsciemment sa satisfaction personnelle durant l'interaction ou en tout il essaye d'une manière qui lui parait optimale d'atteindre son but. Il n'y a pas de raison que l'utilisateur ne poursuive pas cet objectif. L'ARI pourrait donc être utilisé pour inférer la fonction de récompense interne (non-observable) qu'essaye de maximiser l'utilisateur. Il y a plusieurs avantages à cette approche. Tout d'abord, les algorithmes d'ARI pourraient être entraînés sur des données de dialogues homme-machine et pas sur des données de dialogue humain-humain. Ces derniers sont plus difficiles à annoter puisque dans le premier cas, toutes les variables internes du système de dialogue peuvent être automatiquement *loggées*. Deuxièmement, la fonction de récompense apprise pourrait être utilisée comme métrique pour comparer les systèmes de simulation ou les systèmes de dialogue. En effet, un système de simulation d'utilisateur qui agit sous-optimalement par rapport à cette fonction de valeur pourrait être considéré comme peu représentatif du comportement des utilisateurs qu'on cherche à simuler.

De cette manière, nous pouvons espérer concevoir un simulateur d'utilisateur qui s'adapterait aux changements du système de dialogue durant l'optimisation entraînant une amélioration incrémentale du système de dialogue vers un véritable optimum et pas vers un optimum relatif à un système de simulation imparfait. C'est l'objet d'une thèse à venir réalisée par Senthil CHANDRAMOHAN.

Imitation de l'humain et de groupes

L'apprentissage par renforcement inverse, dont le principe a été exposé précédemment, pose un certain nombre de problèmes. Tout d'abord, le problème est en fait mal posé et il existe une infinité de fonctions de récompense qui, étant donné un espace d'état et d'action, peuvent déboucher sur la même politique optimale observée [147]. Plusieurs solutions ont été proposées pour pallier cette limitation comme l'introduction de contraintes dans un système de programmation mathématique [147] ou l'utilisation de programmation quadratique sous contraintes (comme pour les machine à support vecteurs) [4]. Néanmoins, ce problème reste assez peu exploré. Nous nous proposons d'investiguer les possibilités que peuvent offrir la théorie de la régularisation et le *compressed sensing* [48] pour imposer de nouvelles contraintes à la recherche de la fonction de récompense. Dans ces paradigmes, la contrainte imposée est de trouver la représentation la plus parcimonieuse de la fonction que nous recherchons. Les théories dans ces domaines de recherche sont assez bien fondées et pourraient donner des résultats théoriques et pratiques intéressants pour l'ARI.

Un autre problème est qu'il peut être utile d'apprendre de données qui proviennent de l'observation de plusieurs humains (comme c'est le cas pour l'application de dialogue). Dans ce cas, il est possible que différents groupes de personnes agissent de manière différente ou, en d'autres termes, optimisent différentes fonctions de récompense sans qu'il soit possible de s'en apercevoir simplement. Nous aimerions donc étudier comment la quantification vectorielle peut s'introduire dans le paradigme de l'ARI afin de segmenter une base de données en terme de groupes d'utilisateurs. Ceci est aussi particulièrement intéressant dans le domaine du dialogue homme-machine notamment puisqu'il est fréquent que des groupes tels que les novices (les personnes n'ayant jamais utilisé le système avant) et des experts (les personnes

connaissant bien le fonctionnement du système et allant donc droit au but) se distinguent de manière naturelle. Aujourd'hui, un utilisateur moyen est appris sur les données mais celui-ci ne représente finalement pas correctement le comportement des utilisateurs puisque cet utilisateur moyen est rarement présent dans les données.

Enfin, l'humain n'agit pas toujours de manière optimale vis-à-vis de la fonction de récompense qu'il cherche à maximiser. En effet, l'erreur étant humaine, il en résulte des comportement incohérents avec la fonction recherchée et il est nécessaire de trouver des algorithmes tolérants à ces erreurs.

L'ARI implique d'apprendre la fonction de récompense d'un PDM en observant un expert agir de manière optimale. En fait, cette tâche implique un grand nombre de problèmes que nous avons déjà rencontrés dans le cadre de l'AR, à savoir l'approximation (ici de la fonction de récompense), l'estimation de paramètres à partir d'observations bruitées, le caractère éventuellement non-linéaire de la paramétrisation etc. Aussi, la littérature actuelle sur l'ARI suppose que le système à contrôler est entièrement disponible ou simulable et qu'il est possible de générer autant de trajectoires qu'on le souhaite avec n'importe quelle politique (pour la tester). En fait, ceci est rarement vrai pour ne pas dire jamais. Une piste que nous envisageons est de réaliser la tâche de l'ARI sans ce type de simulateur, à partir de données fixes en étendant les méthodes qui ont fait le succès de l'AR hors ligne et que nous avons déjà modifiées pour permettre un apprentissage *off-policy* [80]. Ainsi, nous nous inscrivons dans la continuité des travaux déjà réalisés et nous efforcerons d'unifier nos travaux.

7.2.3 Environnements intelligents et robotique cognitive

Comme indiqué dans la Section **??**, nous avons largement contribué à la création d'un laboratoire d'étude de la la robotique cognitive et des environnements intelligents qui est aussi le cœur de la thématique de recherche de l'UMI 2958 dont nous sommes membre permanent depuis juin 2010. Cette thématique applicative pose tous les problèmes que nous avons cherché à résoudre dans notre travail à savoir des problèmes de contrôle optimal et de traitement de signal dans des conditions où la présence de l'humain impose ses contraintes. C'est en effet le cas ici plus que jamais puisqu'il s'agit d'optimiser des interactions physiques homme-robot-environnement. Le caractère matériel de l'application amène une réflexion supplémentaire sur une approche située du traitement de l'information, c'est-à-dire sur la nécessité de tenir compte de l'environnement et de le savoir disponible pour qu'il fournisse l'information nécessaire à la résolution de la tâche. C'est la thématique que nous avons cherché à développer dans le cadre de l'équipe IMS, que nous avons contribué à fonder et dont nous avons parlé dans la Section **??**.

Toutes les recherches menées dans le cadre du traitement de la parole ou de l'interaction homme-machine trouveront naturellement leur place dans ce cadre. En effet, l'interaction la plus naturelle entre l'homme et ce type d'environnement reste la parole. Néanmoins, le résultat de l'interaction pourra résulter en des modifications d'ordre matériel dans l'environnement (commande de l'éclairage, commande de robots *etc.*), ce qui renforce le caractère situé et même incarné de l'interaction. Par ailleurs, les travaux réalisés dans le domaine plus général du traitement de signal (comme l'identification de locuteurs) pourront aussi être appliqués.

De même, considérant la présence de robots ou considérant l'environnement lui-même comme un robot englobant, les travaux en apprentissage par renforcement s'intégreront aisément. Là encore, le principe d'interaction avec un environnement physique sera de première importance. Par exemple, la nécessité de travailler sur des algorithmes efficaces en termes d'échantillons se fera particulièrement sentir. En effet, il ne sera pas possible d'utiliser l'environnement aussi longtemps qu'on le souhaitera ou dans des conditions arbitrairement favorables. Les contraintes temporelles et matérielles seront prépondérantes et nous permettront d'ancrer nos algorithmes dans la réalité de manière encore plus affirmée que précédemment.

La robotique est appelée à s'immiscer de plus en plus dans la vie quotidienne du grand public. Cela impliquera la cohabitation de l'humain et du robot dans un même environnement, le robot ayant pour objectif de réaliser des tâches pour ou en collaboration avec l'humain. L'imitation par les robots de tâches réalisées par l'humain est une alternative intéressante aux méthodes de planification de tâches pré-établies et qui connaît actuellement un essor important. Les recherches en ARI iront dans ce sens. La robotique cognitive permet aussi d'envisager la production de solutions à large échelle et ainsi, à terme, de réduire les coûts associés au développement. En effet, les méthodes pouvant être "clonées" sur un grand

nombre de robots, elles profiteront au plus grand nombre. C'est pourquoi un paradigme d'apprentissage automatique de tâches, équivalent à un dressage, est préférable à la planification de tâches pré-établies limitant le spectre d'applications potentielles de la robotique. Par ailleurs, des communautés pourront se partager les comportements appris.

7.3 Perspective à long terme

7.3.1 Co-adaptation

Nous pensons que l'étude conjointe de l'AR direct et inverse permettrait une modélisation intéressante des phénomènes de co-adaptation qui interviennent lorsqu'un utilisateur humain s'adapte à un système optimisant son interaction. Par exemple, l'ARI peut permettre la simulation de l'utilisateur d'un système de dialogue alors que l'AR direct peut permettre d'optimiser la stratégie du système. Ces phénomènes donnent lieu une fois encore à des problèmes de non-stationnarité notamment. Cette co-adaptation nous semble être un enjeu majeur de l'intégration de l'informatique ou de la robotique dans le quotidien de tout un chacun.

7.3.2 Bio-inspiration

Nous ne nous sommes pas ou peu intéressés jusqu'à présent au caractère bio-inspiré des méthodes que nous avons développées au cours de notre travail. Pourtant, l'AR prend sa source dans la psychologie animale avec les travaux pionniers d'Edward Thorndike [248] et l'apprentissage bayésien gagne énormément en popularité dans le milieu des sciences cognitives [150]. Bien que nous nous soyons un peu intéressé à cet aspect dans le cadre d'un projet NeuroInformatique du CNRS en collaboration avec l'INRIA Nancy-Grand Est et l'Université de Bordeaux, il n'a pas été profondément étudié. Pourtant, nous pensons qu'il est intéressant d'aborder notre travail sous ce point de vue. D'autant plus que c'est une préoccupation partagée par certains de nos collègues proches dont Hervé FREZZA-BUET. Particulièrement, nous aimerions nous focaliser sur la gestion de la récompense dans les organismes vivants pour améliorer les modèles des systèmes récompensés qui sont pour l'instant restreints aux PDM. Dans ce paradigme, rappelons que la récompense est modélisée comme un simple scalaire. Pire encore, la notion de "punition" est gérée comme une récompense négative. La gestion de la récompense chez les êtres vivants est pourtant beaucoup plus complexe. La frustration ou la motivation intrinsèque sont autant de notions qui ne peuvent pas être bien modélisée dans le cadre défini jusqu'ici. Notons que nous commençons aussi à nous intéresser à l'utilisation de méthodes bayésiennes pour l'estimation des paramètres de champs neuronaux dynamiques.

7.4 Conclusion

Nous concluons ce document en espérant avoir montré notre volonté de transposer nos recherches dans le monde réel en proposant des applications concrètes de nos travaux, inspirées et ensuite utilisées par des acteurs externes à nos recherches et dans le cadre de projets collaboratifs. Ceci est d'autant plus important que dans chacune des applications discutées, l'être humain est présent soit en tant que partie d'un système complexe à contrôler, soit en tant que générateur d'informations à traiter, soit en tant qu'utilisateur critique des méthodes proposées. Il est en effet important pour nous de proposer des solutions viables et de ne pas laisser nos travaux à l'état d'expérience artificielle en laboratoire, le défi ultime étant d'interagir efficacement avec l'humain. Néanmoins, nous espérons aussi avoir montré que cela ne dispense nullement de s'intéresser aux aspects plus fondamentaux de chacun des problèmes théoriques posés par les applications. Par exemple, le cadre des différences temporelles de Kalman ou *Kalman Temporal Differences* (KTD) propose une solution innovante et un point de vue théorique alternatif au problème de l'AR. Les travaux sur le dialogue homme-machine que nous poursuivons sont au meilleur niveau de la recherche sur ce domaine. Les perspectives décrites dans les sections précédentes font aussi état de notre détermination à nous plonger dans ces problèmes fondamentaux.

Les aspects applicatifs de notre travail nécessitent une phase de compréhension puis de modélisation des problèmes que nous apprécions particulièrement. La généricité des solutions apportées nous permet, en effet, de nous intéresser à un grand nombre de problèmes applicatifs différents. Les domaines que nous appelons “applicatifs” de nos centres d’intérêt plus fondamentaux sont parfois eux-mêmes des domaines de recherche à part entière. En effet, l’imagerie médicale ou la robotique sont de tels domaines encore en pleine exploration scientifique. Ainsi, nous pouvons donner l’impression de toucher à beaucoup de choses. Pourtant le point de vue adopté est souvent assez similaire, mettant les contraintes du monde réel au cœur de nos préoccupations.

Cette diversité d’applications nous permet de garder l’ouverture d’esprit nécessaire à tout chercheur. Cela donne lieu aussi à des rencontres et des discussions avec des acteurs de différents domaines, ayant tous des préoccupations différentes. Là aussi, l’aspect humain est particulièrement important et la boucle est ainsi bouclée. C’est en effet à travers nos travaux sur les services que peuvent rendre les machines que nous avons pu avoir les interactions avec les humains les plus enrichissantes d’un point de vue professionnel mais aussi, parfois, d’un point de vue personnel.

Bibliographie

[1] R. Abächerli. *Restauration et analyse de l'électrocardiogramme acquis pendant les examens d'imagerie par résonance magnétique*. PhD thesis, Ecole Doctorale IAEM Lorraine, 2005.

[2] R. Abächerli, S. Hornaff, R. Leber, H.-J. Schmid, and J. Felblinger. Improving automatic analysis of the electrocardiogram acquired during magnetic resonance imaging using magnetic field gradient artefact suppression. *J. Electrocardiology*, 39 :134–139, 2006.

[3] R. Abächerli, C. Pasquier, F. Odille, M. Kraemer, J.-J. Schmid, and J. Felblinger. Suppression of MR gradient artefacts on electrophysiological signals based on an adaptive real-time filter with LMS coefficient updates. *Magma*, 18 :41–50, 2005.

[4] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *In Proceedings of the Twenty-first International Conference on Machine Learning*. ACM Press, 2004.

[5] H. Ai. Comparing user simulation models for dialog strategy learning. In *In Proc. of NAACL-HLT*, 2007.

[6] X. Anguera. *Robust Speaker Diarization for Meetings*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, October 2006.

[7] ANSI/AAMI :EC57. Testing and reporting performance results of cardiac rhythm and ST-segment measurement algorithms, 1998.

[8] A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1) :89–129, April 2008.

[9] R. Aras and O. Pietquin. Optimal Average Reward Controllers For POMDPs. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, Besançon (France), June 2010. 6 pages.

[10] M. Bagein, O. Pietquin, C. Ris, and G. Wilfart. An Architecture for Voice-Enabled Interfaces over Local Wireless Networks. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, Orlando, (USA, FL), July 2003. 6 pages.

[11] M. Bagein, O. Pietquin, C. Ris, and G. Wilfart. Enabling Speech Based Access to Information Management Systems over Wireless Network. In *Proceedings of the 3rd workshop on Applications and Services in Wireless Networks (ASWN 2003)*, Berne (Switzerland), July 2003. 6 pages.

[12] L. C. Baird. Residual Algorithms : Reinforcement Learning with Function Approximation. In *Proceedings of the International Conference on Machine Learning*, pages 30–37, 1995.

[13] M. Basseville. Distance measures for signal processing and pattern recognition. *Signal Processing*, 18(4) :349–369, 1999.

[14] R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 1957.

[15] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee. Incremental Natural Actor-Critic Algorithms. In *Proceedings of the Twenty-First Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, 2008.

[16] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, USA, 1995.

[17] J. Boger, J. Hoey, P. Poupart, C. Boutilier, G. Fernie, and A. Mihailidis. A planning system based on markov decision processes to guide people with dementia through activities of daily living. *IEEE Transactions on Information Technology and Biomedicine*, 10(2) :323–333, April 2006.

[18] L. Bottou and Y. Bengio. Convergence properties of the k-means algorithms. In *Proceedings of the 9th Conference on Neural Information Processing Systems (NIPS)*, pages 585–92, Denver, CO, USA, 1995.

[19] J. A. Boyan. Technical Update : Least-Squares Temporal Difference Learning. *Machine Learning*, 49(2-3) :233–246, 1999.

[20] S. J. Bradtke and A. G. Barto. Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3) :33–57, 1996.

[21] M. Catala, J. Andre, G. Katsanis, and J. Poirier. Histologie : organes, systèmes et appareils. http ://www.chups.jussieu.fr/polys/histo/histoP2/index.html, 2007. Faculté de médecine Pierre et Marie Curie, Site de la Pitié-Salpêtrière.

[22] S. Chague, J. D'Hose, J.-F. Goudou, B. Dorizzi, L. Giulieri, Q.-C. Pham, F. Sedes, M. Brut, D. Nicholson, and O. Pietquin. METHODEO : Méthodologie d'évaluation des algorithmes d'exploitation des enregistrements de la vidéoprotection. In *Workshop Interdisciplinaire sur la Sécurité Globale (WISG 2011)*, Troyes (France), January 2011. 6 pages.

[23] S. Chandramohan, M. Geist, and O. Pietquin. Optimizing spoken dialogue management with fitted value iteration. In *Proceedings of the International Conference on Speech Communication and Technologies (Interspeech 2010)*, pages 86–89, Makuhari (Japan), September 2010. ISCA.

[24] S. Chandramohan, M. Geist, and O. Pietquin. Sparse approximate dynamic programming for dialog management. In *Proceedings of the 11th SIGDial Conference on Discourse and Dialogue*, pages 107–115, Tokyo (Japan), September 2010. ACL.

[25] S. Chandramohan and O. Pietquin. User and noise adaptive dialogue management using hybrid system actions. In G. G. Lee, J. Mariani, W. Minker, and S. Nakamura, editors, *Spoken Dialogue Systems for Ambient Environments*, volume 6392 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 13–24, Gotemba, Shizuoka (Japan), October 2010. Springer Verlag, Heidelberg - Berlin. Proceedings of the International Workshop on Spoken Dialogue Systems (IWSDS 2010).

[26] G. Chartrand, G. Kubicki, and M. Schultz. Graph similarity and distance in graphs. *Aequationes Mathematicae*, 55(1-2) :129–145, 1998.

[27] Z. Chen. Bayesian Filtering : From Kalman Filters to Particle Filters, and Beyond. Technical report, Adaptive Systems Lab, McMaster University, 2003.

[28] B. Chevaillier. *Analyse de données d'IRM fonctionnelle rénale par quantification vectorielle*. Thèse en mathématiques, Université Paul Verlaine de Metz, mars 2010.

[29] B. Chevaillier, J.-L. Collette, D. Mandry, M. Claudon, and O. Pietquin. Objective assessment of renal DCE-MRI image segmentation. In *Proceedings of the European Signal Processing Conference (EUSIPCO 2010)*, Aalborg (Danmark), August 2010. Eurasip. 1214-1218.

[30] B. Chevaillier, J.-L. Collette, D. Mandry, M. Claudon, and O. Pietquin. Segmentation fonctionnelle de séquences d'IRM rénales à rehaussement de contraste par quantification vectorielle. In *Colloque Recherche en Imagerie et Technologies pour la Santé (RITS 2011)*, Rennes (France), April 2011. 3 pages.

[31] B. Chevaillier, D. Mandry, J.-L. Collette, M. Claudon, M.-A. Galloy, and O. Pietquin. Functional segmentation of renal DCE-MRI sequences using unsupervised learning algorithms. *Neural Processing Letters*, page 14 pages, 2011. accepted for publication - Impact Factor 1.47.

[32] B. Chevaillier, D. Mandry, J.-L. Collette, M. Claudon, and O. Pietquin. Functional semi-automated segmentation of renal DCE-MRI sequences using a Growing Neural Gas algorithm. In A. A. Zaher, editor, *Recent Advances in Signal Processing*, chapter 5, pages 69–80. Nov 2009.

[33] B. Chevaillier, D. Mandry, Y. Ponvianne, J.-L. Collette, M. Claudon, and O. Pietquin. Functional semi-automated segmentation of renal DCE-MRI sequences using a Growing Neural Gas algorithm. In *Proceedings of the 16th European Signal Processing Conference (EUSIPCO'08)*, page 5 pages (Proceedings on CDROM), Lausanne (Switzerland), August 2008.

[34] B. Chevaillier, Y. Ponvianne, J.-L. Collette, D. Mandry, M. Claudon, and O. Pietquin. Functional Semi-Automated Segmentation of Renal DCE-MRI Sequences. In *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 525–528, Las Vegas (NV, USA), April 2008.

[35] D. Choi and B. Van Roy. A Generalized Kalman Filter for Fixed Point Approximation and Efficient Temporal-Difference Learning. *Discrete Event Dynamic Systems*, 16 :207–239, 2006.

[36] H. Clark and E. Schaefer. Contributing to discourse. *Cognitive Science*, 13 :259–294, 1989.

[37] M. Claudon, D. Mandry, C. Pasquier, B. Chevaillier, J.-L. Collette, and O. Pietquin. Functional semi-automated segmentation of renal DCE-MRI sequences : preliminary results. In *Proceedings of the 15th Symposium of the European Society of Urogenital Radiology (ESUR 2008)*, Munich (Germany), September 2008.

[38] G. D. Clifford. A Novel Framework for Signal Representation and Source Separation : Applications to Filtering and Segmentation of Biosignals. *Journal of Biological Systems*, 14(2) :169–183, 2006.

[39] G. D. Clifford, F. Azuaje, and P. E. McSharry. *Advanced Tools and methods for ECG Data Analysis*. Artech House, 2007.

[40] G. D. Clifford, A. Shoeb, P. E. McSharry, and B. A. Janz. Model-based Filtering Compression and Classification of the ECG. *International Journal of Bioelectromagnetism*, 7(1) :158–161, 2005.

[41] M. Clynes. Computer Analysis of Reflex Control and Organization : Respiratory Sinus Arrhythmia . *Science*, 131(3396) :300 – 302, 1960.

[42] J.-L. Collette and O. Pietquin. Localisation de Source Sonore par Goniometrie Acoustique pour la Détection de Chute. In *2ème colloque PARAChute*, Troyes (France), November 2009.

[43] L. Daubigney, M. Geist, and O. Pietquin. Apprentissage par renforcement pour la personnalisation d'un logiciel d'enseignement des langues. In *Conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH 2011)*, Mons (Belgium), May 2011. 4 pages.

[44] L. Daubigney and O. Pietquin. Single-pass P300 detection with Kalman filtering and SVMs. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2011)*, Bruges (Belgium), April 2011. 6 pages.

[45] R. Dearden, N. Friedman, and S. J. Russell. Bayesian Q-Learning. In *AAAI/IAAI*, pages 761–768, 1998.

[46] P. Delacourt and C. Wellekens. DISTBIC : A speaker-based segmentation for audio data indexing. *Speech Communication*, 32(1-2) :111–126, September 2000.

[47] S. Denslow and D. S. Buckles. Pulse Oximetry-Gated Acquisition of Cardiac MR images in Patients with Congenital Cardiac Abnormalities. *Am. J. Roent.*, 160 :831–833, 1993.

[48] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4) :1289 –1306, 2006.

[49] Y. Engel. *Algorithms and Representations for Reinforcement Learning*. PhD thesis, Hebrew University, April 2005.

[50] Y. Engel, S. Mannor, and R. Meir. Reinforcement Learning with Gaussian Processes. In *Proceedings of International Conference on Machine Learning (ICML-05)*, 2005.

[51] J. Felblinger and C. Boesch. Amplitude Demodulation of the Electrocardiogram signal (ECG) for respiration monitoring and compensation during MR examinations. *Magn. Res. Med.*, 38 :129–136, 1997.

[52] J. Felblinger, C. Lehmann, and C. Boesch. Electrocardiogram acquisition during MR examinations for Patient monitoring and sequence triggering. *Magn. Res. Med.*, 32 :523–529, 1994.

[53] J. Felblinger, J. Slotboom, R. Kreis, B. Jung, and C. Boesch. Restoration of Electrophysiological Signals by Inductive Effects of Magnetic Field Gradients during MR sequences. *Magn. Res. Med.*, 41 :715–721, 1999.

[54] B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Synchronisation Adaptative Utilisant un Modèle Prédictif : Applications à lŠImagerie Cardiaque par Résonance Magnétique en Sang Noir et en Systole. Journée Claude Huriet, Nancy (France), December 2008.

[55] B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Adaptive Trigger Delay Using a Predictive Model Applied to Black Blood Fast Spin Echo Cardiac Imaging in Systole. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009.

[56] B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Amélioration de lŠimagerie du ventricule droit en IRM cardiaque en ń sang noir ż par une méthode adaptative. In *Journée Claude Huriet*, Nancy (France), December 2009.

[57] B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Synchronisation adaptative utilisant un modèle prédictif : Applications à l'imagerie cardiaque par résonance magnétique en sang noir et en systole. In *Actes des Journées de Recherche en Imagerie et Technologies de la Santé (RITS 2009)*, Lille (France), mars 2009.

[58] B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Adaptive Black Blood Fast Spin Echo for End-Systolic Rest Cardiac Imaging. *Magnetic Resonance in Medicine*, 64(6) :1760–1771, December 2010. - Impact Factor 3.131.

[59] B. Fernandez, J. Oster, M. Lohezic, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Beat to Beat Management of Heart Cycle Changes for Black Blood Imaging in End-Systolic Rest. In *Proceedings of the ISMRM workshop on Current Concepts of Motion Correction for MRI and MRS*, Kitzbuhel (Austria), February 2010.

[60] S. E. Fischer, S. A. Wickline, and C. H. Lorenz. Novel Real-Time R-wave detection Algortihm Based on Vectocardiogram for Accurate Gated Magnetic Resonance Acquisitions. *Magn. Res. Med.*, 42 :361–370, 1999.

[61] J. Fix, M. Geist, O. Pietquin, and H. Frezza-Buet. Dynamic Neural Field Optimization using the Unscented Kalman Filter. In *Proceedings of the IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB 2011)*, Paris (France), April 2011. 8 pages.

[62] J. L. Fleckenstein, B. T. Archer, B. A. Barker, J. T. Vaughan, R. W. Parkey, and R. M. Peshock. Fast short-tau inversion recovery MR imaging. *Radiology*, 179 :499–504, 1991.

[63] T. Frauenrath, S. Kozerke, P. Boesiger, and T. Niendorf. Cardiac Gating Free of Interference with Electro-Magnetic Fields at 1.5T, 3.0T and 7.0T. In *Proceedings of the annual meeting of the Int. Soc. for Magn. Res. in Med.*, page 207, 2008.

[64] T. Frauenrath, T. Niendorf, and M. Kob. Acoustic Method for Synchronization of Magnetic Resonance Imaging (MRI). *Acta Acustica united with Acustica*, 94 :148–155, 2008.

[65] H. Frezza-Buet. Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network. *Neurocomputing*, 71(7-9) :1191–1202, March 2008.

[66] H. Frezza-Buet. Following non-stationary distributions by controlling the vector quantization accuracy of a growing neural gas network. *Neurocomputing*, 71(7-9), 2008.

[67] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.

[68] B. Fritzke. A self-organizing network that can follow non-stationary distributions. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN'97)*, pages 613–618. Springer, 1997.

[69] V. Galtier, O. Pietquin, and S. Vialle. AdaBoost Parallelization on PC Clusters with Virtual Shared Memory for Fast Feature Selection. In *Proceedings of the 1st IEEE International Conference on Signal Processing and Communication*, pages 165–168, Dubai (United Arab Emirates), November 2007.

[70] F. Gaspard, O. Pietquin, F. Sèdes, J.-F. Seignole, and J.-F. Sulzer. Architecture Générique de Stockage Multimedia Réparti avec Recherche et Indexation distribuées (Projet ITEA2 LINDO). In *Journée d'étude sur l'Analyse Vidéo pour le Renseignement et la Sécurité (AViRS 2008)*, Paris (France), April 2008. SEE.

[71] M. Geist. *Optimisation des chaînes de production dans l'industrie sidérurgique : une approche statistique de l'apprentissage par renforcement*. Thèse en mathématiques, Université Paul Verlaine de Metz, Novembre 2009.

[72] M. Geist and O. Pietquin. A Brief Survey of Parametric Value Function Approximation. Technical report, September 2010.

[73] M. Geist and O. Pietquin. Eligibility Traces through Colored Noises. In *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia), October 2010. 8 pages (best paper award).

[74] M. Geist and O. Pietquin. Gestion de l'incertitude dans le cadre de l'approximation de la fonction de valeur pour l'apprentissage par renforcement. In *actes de la conférence francophone sur l'apprentissage automatique (CAP 2010)*, pages 101–112, Clermont-Ferrand (France), May 2010. PUG.

[75] M. Geist and O. Pietquin. Kalman temporal differences. *Journal of Artificial Intelligence Research (JAIR)*, 39 :489–532, October 2010. - Impact Factor (2008) 3.241.

[76] M. Geist and O. Pietquin. Managing Uncertainty within Value Function Approximation in Reinforcement Learning. In *Active Learning and Experimental Design workshop (collocated with AISTATS 2010)*, Sardinia, Italy, 2010. 8 pages, oral presentation.

[77] M. Geist and O. Pietquin. Revisiting natural actor-critics with value function approximation. In *Proceedings of the 22nd Benelux Conference on Artificial Intelligence (BNAIC 2010)*, Luxembourg (Luxembourg), October 2010. to appear.

[78] M. Geist and O. Pietquin. Revisiting natural actor-critics with value function approximation. In V. Torra, Y. Narukawa, and M. Daumas, editors, *Proceedings of 7th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2010)*, volume 6408 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 207–218, Perpinya (France), October 2010. Springer Verlag - Heidelberg Berlin.

[79] M. Geist and O. Pietquin. Revisiting natural actor-critics with value function approximation. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, Besançon (France), June 2010. 6 pages.

[80] M. Geist and O. Pietquin. Statistically Linearized Least-Squares Temporal Differences. In *Proceedings of the IEEE International Conference on Ultra Modern Control systems (ICUMT 2010)*, Moscow (Russia), October 2010. IEEE. 8 pages.

[81] M. Geist and O. Pietquin. Statistically Linearized Least-Squares Temporal Differences. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2010)*, Besançon (France), June 2010. 8 pages.

[82] M. Geist and O. Pietquin. Statistically Linearized Recursive Least Squares. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010)*, Kittilä (Finland), August-September 2010. 5 pages, to appear.

[83] M. Geist and O. Pietquin. Managing Uncertainty within the KTD Framework. In *Proceedings of the Workshop on Active Learning and Experimental Design (AL&E collocated with AISTAT 2010)*, Journal of Machine Learning Research Conference and Workshop Proceedings, Sardinia (Italy), 2011. 12 pages - to appear.

[84] M. Geist and O. Pietquin. Parametric Value Function Approximation : a Unified View. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2011)*, Paris (France), April 2011. 8 pages.

[85] M. Geist, O. Pietquin, and G. Fricout. A Sparse Nonlinear Bayesian Online Kernel Regression. In *Proceedings of the 2nd IEEE International Conference on Advanced Engineering Computing and*

Applications in Sciences (AdvComp 2008), volume I, pages 199–204, Valencia (Spain), October 2008. (best paper award).

[86] M. Geist, O. Pietquin, and G. Fricout. Bayesian reward filtering. In S. G. et al., editor, *Recent Advances in Reinforcement Learning*, volume 5323 of *Lecture Notes in Computer Science (LNCS)*, pages 96–109. Springer Verlag, June 2008. Revised and selected papers of EWRL 2008.

[87] M. Geist, O. Pietquin, and G. Fricout. Bayesian Reward Filtering. In *8th European Workshop on Reinforcement Learning (EWRL 2008)*, Lille (France), June 2008. 14 pages.

[88] M. Geist, O. Pietquin, and G. Fricout. Filtrage bayésien de la récompense. In *actes des Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2008)*, pages 113–122, Metz (France), June 2008.

[89] M. Geist, O. Pietquin, and G. Fricout. Filtrage Bayésien de la Récompense. In *Journée NeuroInfo - Loria*, Nancy (France), July 2008.

[90] M. Geist, O. Pietquin, and G. Fricout. Kalman Temporal Differences : Uncertainty and Value Function Approximation. In *NIPS Workshop on Model Uncertainty and Risk in Reinforcement Learning*, Vancouver (Canada), December 2008.

[91] M. Geist, O. Pietquin, and G. Fricout. Online Bayesian Kernel Regression from Nonlinear Mapping of Observations. In *Proceedings of the 18th IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2008)*, number a53, pages 309–314, Cancun (Mexico), October 2008.

[92] M. Geist, O. Pietquin, and G. Fricout. Différences Temporelles de Kalman. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, Paris (France), June 2009. 20 pages.

[93] M. Geist, O. Pietquin, and G. Fricout. Différences Temporelles de Kalman : le cas stochastique. In *Journées Francophones de Planification, Décision et Apprentissage pour la conduite de systèmes (JFPDA 2009)*, Paris (France), June 2009. 13 pages.

[94] M. Geist, O. Pietquin, and G. Fricout. From supervised to reinforcement learning : a kernel-based Bayesian filtering framework. *International Journal On Advances in Software*, 2(1) :101–116, 2009.

[95] M. Geist, O. Pietquin, and G. Fricout. Kalman Temporal Differences : the deterministic case . In *IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL 2009)*, pages 185–192, Nashville (TN, USA), April 2009.

[96] M. Geist, O. Pietquin, and G. Fricout. Kernelizing Vector Quantization Algorithms. In M. Verleysen, editor, *Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN 09)*, pages 541–546, Bruges (Belgium), April 2009.

[97] M. Geist, O. Pietquin, and G. Fricout. Tracking in Reinforcement Learning. In *Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009)*, volume 5863, Part I, pages 502–511, Bangkok (Thailand), December 2009. Springer LNCS. ENNS best student paper award.

[98] M. Geist, O. Pietquin, and G. Fricout. Astuce du Noyau & Quantification Vectorielle. In *Actes du 17ème colloque sur la Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'10)*, Caen (France), January 2010. 8 pages.

[99] M. Geist, O. Pietquin, and G. Fricout. Différences temporelles de Kalman : cas déterministe. *Revue d'Intelligence Artificielle*, 24(2) :423–442, September 2010.

[100] A. Gersho and R. Gray. *Vector quantization and signal compression*. Dordrecht, Netherlands, 1992.

[101] G. Gordon. Stable Function Approximation in Dynamic Programming. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1995.

[102] H. Gray. *Anatomy of the Human Body*. Philadelphia : Lea and Febiger, 1918 ; New York : Bartleby.com, http ://www.bartleby.com/107/, 20th edition, 2000.

[103] J.-L. Gutzwiller, H. Frezza-Buet, and O. Pietquin. Online Speaker Diarization with a Size-Monitored Growing Neural Gas Algorithm. In M. Verleysen, editor, *Proceedings of the 18 th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 505–510, Bruges (Belgium), April 2010.

[104] J. Henderson, O. Lemon, and K. Georgila. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Comput. Linguist.*, 34(4) :487–511, 2008.

[105] e. A. L. Istvan Szita, Viktor Gyenes. Reinforcement learning with echo state networks. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 2006.

[106] H. Jaeger. The"echo state" approach to analyzing and training recurrent neural networks. GMD Report 148, Fraunhofer Institute for Autonomous Intelligent Systems, 2001.

[107] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3) :401–422, 2004.

[108] L. P. Kaelbling. *Learning in embedded systems*. MIT Press, 1993.

[109] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D) :35–45, 1960.

[110] S. Keizer, O. Pietquin, S. Rossignol, and S. Young. Agenda-based user simulations and EM training tools for Appointment Scheduling and TownInfo domains. CLASSiC Project Deliverable 3.4, October 2010.

[111] J. Kelley. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems*, 2(1) :26–41, 1984.

[112] T. Kinnunen and H. Li. An overview of text-independent speaker recognition : From features to supervectors. *Speech Communication*, 52(1) :12 – 40, 2010.

[113] J. Kleinberg. An impossibility theorem for clustering. In *Proceedings of the 16th Conference on Neural Information Processing Systems (NIPS 2002)*, Vancouver, B.C., Canada, 2002. 8 pages.

[114] A. Kolin. Electromagnetic blood flow meters. *Science*, 130 :1088–1097, 1959.

[115] J. Z. Kolter and A. Y. Ng. Near-Bayesian Exploration in Polynomial Time. In *Proceedings of the 26th international conference on Machine learning (ICML 09)*, New York, NY, USA, 2009. ACM.

[116] V. R. Konda and J. N. Tsitsiklis. On actor-critic algorithms. *SIAM J. Control Optim.*, 42(4) :1143–1166, 2003.

[117] K.-D. Kuhnert and M. Krödel. Autonomous vehicle steering based on evaluative feedback by reinforcement learning. In P. Perner and A. Imiya, editors, *Machine Learning and Data Mining in Pattern Recognition*, volume 3587 of *Lecture Notes in Computer Science*, pages 405–414. Springer Berlin / Heidelberg, 2005.

[118] S. Laborie, F. Sedes, J.-F. Sulzer, O. Pietquin, N. Allezard, R. Pinchuk, J.-P. Guignard, P. Moreira, C. Fernandez, S. Moens, J.-F. Seignole, and D. Milan. Proposed Content Indexation Agent Software. LINDO Deliverable D3.1, 2008.

[119] M. G. Lagoudakis and R. Parr. Least-Squares Policy Iteration. *Journal of Machine Learning Research*, 4 :1107–1149, 2003.

[120] L. F. Lamel, J.-L. Gauvain, and M. Eskénazi. BREF, a large vocabulary spoken corpus for French. In *Proceedings of the European Conference on Speech Technologies (Eurospeech'91)*, pages 505–508, 1991.

[121] S. Larsson and D. Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural language engineering*, 6(3&4) :323–340, 2001.

[122] O. Lemon, K. Georgila, J. Henderson, and M. Stuttle. An ISU dialogue system exhibiting reinforcement learning of dialogue policies : generic slot-filling in the TALK in-car system. In *Proceedings of the meeting of the European chapter of the Associaton for Computational Linguistics (EACL'06)*, Morristown, NJ, USA, 2006.

[123] O. Lemon, K. Georgila, J. Henderson, and M. Stuttle. An ISU dialogue system exhibiting reinforcement learning of dialogue policies : generic slot-filling in the TALK in-car system. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 119–122. Association for Computational Linguistics, 2006.

[124] O. Lemon and O. Pietquin. Machine Learning for Spoken Dialogue Systems. In *Proceedings of the 10th European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, Anvers (Belgium), August 2007.

[125] O. Lemon, O. Pietquin, H. Frezza-Buet, V. Rieser, X. Liu, P. Bretier, S. Young, and J. Henderson. Shared Context Model (XML Schema). CLASSiC Project Deliverable 3.1, February 2009.

[126] E. Levin and R. Pieraccini. Using markov decision process for learning dialogue strategies. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'98), Seattle, Washington*, 1998.

[127] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 2000.

[128] L. Li, S. Balakrishnan, and J. Williams. Reinforcement Learning for Dialog Management using Least-Squares Policy Iteration and Fast Feature Selection. In *Proceedings of the International Conference on Speech Communication and Technologies (InterSpeech'09)*, Brighton (UK), 2009.

[129] F. Liu, H. Zhao, and S. Crozier. On the Induced Electric Field Gradients in the Human Body for Magnetic Resonance Stimulation by Gradient Coils in MRI. *IEEE Trans. Biomed. Eng.*, 50(7) :804–815, 2003.

[130] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2) :129–137, 1982.

[131] M. Lohezic, B. Fernandez, J. Oster, D. Mandry, O. Pietquin, P.-A. Vuissoz, and J. Felblinger. Free breathing black-blood systolic imaging using heart rate prediction and motion compensated reconstruction. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009.

[132] U. V. Luxburg and S. Ben-david. Towards a statistical theory of clustering. In *In PASCAL workshop on Statistics and Optimization of Clustering*, 2005.

[133] D. Mandry, B. Chevaillier, J.-L. Collette, M.-A. Galloy, Y. Ponvianne, J. Felblinger, O. Pietquin, and M. Claudon. Functional semi-automated segmentation of renal DCE-MRI sequences using vector quantization. In *Proceedings of the European Society for Magnetic Resonance in Medicine and Biology congress (ESMRMB 08)*, Valencia (Spain), October 2008.

[134] D. Mandry, B. Chevaillier, Y. Ponvianne, J.-L. Collette, M.-A. Galloy, O. Pietquin, and M. Claudon. Segmentation fonctionnelle rénale semi-automatique par quantification vectorielle de séries dynamiques en IRM. In *Journées Françaises de Radiologie (JFR 2008)*, Paris (France), October 2008.

[135] C. A. K. Marilyn A. Walker, Diane J. Litman and A. Abella. PARADISE : A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL'97)*, pages 271–280, Madrid (Spain), 1997.

[136] T. Martinetz, S. Berkovich, and K. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4) :558–569, 1993.

[137] T. Martinetz and K. Schulten. Topology representing networks. *Neural Networks*, 7(3) :507–522, 1994.

[138] T. M. Martinez and K. J. Schulten. Topology representing networks. *Neural Networks*, 7(3) :507–522, 1994.

[139] T. Martinez-Marin. A reinforcement learning algorithm for optimal motion of car-like vehicles. In *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pages 47 – 51, oct. 2004.

[140] D. Mattes, D. Haynor, H. Vesselle, T. Lewellen, and W. Eubank. PET-CT image registration in the chest using free-form deformations. *IEEE Transactions on Medical Imaging*, 22(1) :120–128, 2003.

[141] D. McRobbie and M. A. Foster. Cardiac response to pulsed magnetic fields with regard to safety in NMR imaging. *Phys. Med. Biol.*, 30(7) :695–702, 1985.

[142] P. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith. A Dynamical Model for Generating Synthetic Electrocardiogram Signals. *IEEE Trans. Biomed. Eng.*, 50 :289–294, 2003.

[143] J.-P. Morard, O. Pietquin, and S. Vialle. Method for the segmentation encoding of an image, July 2010. Patent n° WO/2010/072983.

[144] J.-P. Morard, S. Vialle, O. Pietquin, and V. Galtier. Method for broadcasting video data sequences by a server to a client terminal, March 2009. Patent n° WO/2009/034275.

[145] J.-P. Morard, S. Vialle, O. Pietquin, and V. Galtier. Procédé de diffusion de séquences de données vidéo par un serveur vers un terminal client, March 2009. Brevet n° FR2920933 (A1).

[146] F. H. Netter. *Ciba Collection of Medical Illustrations*, volume 5. Ciba Pharmaceutical Company, 1992.

[147] A. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of 17th International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann, 2000.

[148] T. Niendorf, M. Kob, and T. Frauenrath. ACT-MR : ACoustically Triggered Cardiovascular Magnetic Resonance Imaging. In *Proceedings of the annual meeting of the Int. Soc. for Magn. Res. in Med.*, page 765, 2007.

[149] T. Niendorf, M. Kob, and T. Frauenrath. Acoustic Triggering of a Magnetic Resonance Imaging Device. WO 2008/077495, 2008.

[150] M. Oaksford and N. Chater. *Bayesian Rationality*. Oxford University Press, Oxford, 2006.

[151] F. Odille. *Imagerie Adaptative en IRM : Utilisation des Informations de Mouvements Physiologiques pour l'Optimisation des Processus d'Acquisition et de Reconstruction*. PhD thesis, Université Henri Poincaré, Nancy 1, 2007.

[152] F. Odille, C. Pasquier, R. Abächerli, P.-A. Vuissoz, G. P. Zientara, and J. Felblinger. Noise Cancellation Signal Processing Method and Computer System for Improved Real-time Electrocardiogram Artifact Correction during MRI Data Acquisition. *IEEE Trans. Biomed. Eng.*, 54 :630–640, 2007.

[153] S.-Y. Oh, J.-H. Lee, and D.-H. Choi. A new reinforcement learning vehicle control architecture for vision-based road following. *IEEE Transactions on Vehicular Technology*, 49(3) :997 –1005, may. 2000.

[154] J. Oster. *Traitement temps-réel des signaux électrophysiologiques acquis dans un environnement d'Imagerie par Résonance Magnétique*. Thèse en automatique et traitement du signal, Nancy Université, Novembre 2009.

[155] J. Oster, B. Fernandez, M. Lohezic, D. Mandry, P.-A. Vuissoz, O. Pietquin, and J. Felblinger. Adaptive Heart Rate Prediction for Black-Blood Systolic Iimaging. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009.

[156] J. Oster, F. Odile, G. Bosser, O. Pietquin, C. Pasquier, P.-A. Vuissoz, and J. Felblinger. Adaptive Prediction of RR interval for online MR parameters changes. In *Proceedings of the Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM 2007)*, Berlin (Germany), May 2007.

[157] J. Oster, J. Pascal, O. Pietquin, M. Kraemer, J.-P. Blondé, and J. Felblinger. Real-Time Adaptive suppression of MR gradient Artifacts on Electorcardiograms using a new 3D Hall Probe. In *Proceedings of the 17th meeting of the International Society for Magnetic Resonance Medicine (ISMRM 2009)*, Honolulu (Hawaii, USA), April 2009.

[158] J. Oster, O. Pietquin, R. Abächerli, M. Kraemer, and J. Felblinger. A Specific QRS Detector for Electrocardiography during MRI : Using Waveltets and Local Regularity Characterization. In *Proceedings of the 34th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pages 341–344, Taipei (Taiwan), April 2009.

[159] J. Oster, O. Pietquin, R. Abächerli, M. Kraemer, and J. Felblinger. Independent component analysis based artefact reduction : application to electrocardiogram for improved magnetic resonance imaging triggering. *Physiological Measurement*, 30 :1381–1397, November 2009. Impact Factor 1.691.

[160] J. Oster, O. Pietquin, G. Bosser, and J. Felblinger. Adaptive RR Prediction for Cardiac MRI. In *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 513–516, Las Vegas (NV, USA), April 2008.

[161] J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Independent Component Analysis based Artifact Reduction Method for ECG in MR. In *Proceedings of the European Society for Magnetic Resonance in Medicine and Biology congress (ESMRMB 08)*, Valencia (Spain), October 2008.

[162] J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Méthode de réduction des artéfacts présents sur l'ECG basée sur l'analyse en composantes indépendantes. In *Actes du 12ème congrès du Groupe de Recherche sur les Applications du Magnétisme en Médecine (GRAMM'08)*, Lyon (France), mars 2008.

[163] J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Bayesian Framework for Artifact Reduction on ECG in MRI. In *Proceedings of the 35th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pages 489–492, Dallas (TX, USA), March 2010.

[164] J. Oster, O. Pietquin, M. Kraemer, and J. Felblinger. Nonlinear Bayesian Filtering for Denoising of Electrocardiograms acquired in a Magnetic Resonance Environment. *IEEE Transactions on Biomedical Engineering*, 57(7) :1628 – 1638, May 2010. Impact Factor 1.322.

[165] T. Paek and R. Pieraccini. Automating spoken dialogue management design using machine learning : An industry perspective. *Speech Communication*, 50 :716–729, 2008.

[166] J. Park and I. Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2) :246–257, 1991.

[167] J. Peters, S. Vijayakumar, and S. Schaal. Natural Actor-Critic. In J. G. et al., editor, *Proceedings of the European Conference on Machine Learning (ECML 2005)*, Lecture Notes in Artificial Intelligence. Springer Verlag.

[168] C. W. Phua and R. Fitch. Tracking Value Function Dynamics to Improve Reinforcement Learning with Piecewise Linear Function Approximation. In *Proceedings of the International Conference on Machine Learning (ICML 07)*, 2007.

[169] J. Picone. Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9) :1215–1247, September 1993.

[170] O. Pietquin. Algorithme de recouvrement à haute vitesse de données mémorisées sur support optique. Master's thesis, Faculté Polytechnique de Mons (FPMs), Belgium, 1999.

[171] O. Pietquin. Environnement virtuel pour la simulation et l'apprentissage de stratégies de dialogue. In *Actes de la 15ème Conférence Francophone sur l'Interaction Homme-Machine (IHM 2003)*, Caen (France), November 2003. 4 pages.

[172] O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. SIMILAR Collection. Presses Universitaires de Louvain, 2004. 246 pages.

[173] O. Pietquin. *A Framework for Unsupervised Learning of Dialogue Strategies*. PhD thesis, Faculté Polytechnique de Mons, TCTS Lab (Belgique), April 2004.

[174] O. Pietquin. Une description probabiliste de la communication parlée entre homme et machine. In *Actes de la 16ème Conférence Francophone sur l'Interaction Homme-Machine (IHM 2004)*, pages 247–250, Namur (Belgique), August-September 2004.

[175] O. Pietquin. A Probabilistic Description of Man-Machine Spoken Communication. In *Proceedings of the 5th IEEE International Conference on Multimedia and Expo (ICME 2005)*, pages 410–413, Amsterdam (The Netherlands), July 2005.

[176] O. Pietquin. Réseau bayésien pour un modèle d'utilisateur et un module de compréhension pour l'optimisation des systèmes de dialogues. In *Actes de la Conférence Francophone sur le Traitement du Langage Naturel (TALN 2005)*, volume I, pages 481–486, Dourdan (France), June 2005.

[177] O. Pietquin. Consistent Goal-Directed User Model for Realistic Man-Machine Task-Oriented Spoken Dialogue Simulation. In *Proceedings of the 7th IEEE International Conference on Multimedia and Expo*, pages 425–428, Toronto (Canada), July 2006.

[178] O. Pietquin. Machine learning for spoken dialogue management : an experiment with speech-based database querying. In J. E. . J. Domingue, editor, *Artificial Intelligence : Methodology, Systems & Applications*, volume 4183 of *Lecture Notes in Artificial Intelligence*, pages 172–180. Springer Verlag, 2006.

[179] O. Pietquin. Learning to Ground in Spoken Dialogue Systems. In *Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume IV, pages 165–168, Honolulu (Hawaii, USA), April 2007.

[180] O. Pietquin. Un cadre probabiliste pour l'optimisation des systèmes de dialogue. In *Proceedings of the 4rth International Conference : Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2007)*, Hammamet (Tunisia), March 2007. 8 pages.

[181] O. Pietquin. A method of recognizing a motion pattern of an object, April 2008. Patent n° EP1904951.

[182] O. Pietquin. Optimising spoken dialogue strategies within the reinforcement learning paradigm. In M. E. Cornelius Weber and N. M. Mayer, editors, *Reinforcement Learning, Theory and Applications*, pages 239–256. I-Tech Education and Publishing, Vienna, Austria, January 2008.

[183] O. Pietquin. Machine learning methods for spoken dialogue simulation and optimization. In A. Mellouk and A. Chebira, editors, *Machine Learning*, pages 167–184. IN-TECH, January 2009.

[184] O. Pietquin. Natural language and dialogue processing. In F. M. Jean-Philippe Thiran, Hervé Bourlard, editor, *Multi-modal signal processing : methods and techniques to build multimodal interactive systems*, chapter 4, pages 61–90. Elsevier Science & Technology Books, January 2010.

[185] O. Pietquin. Batch reinforcement learning for spoken dialogue systems with sparse value function approximation. In *NIPS Workshop on Learning and Planning from Batch Time Series Data*, Vancouver (Canada), 2011.

[186] O. Pietquin and R. Beaufort. Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning. In *Proceedings of the 9th European Conference on Speech Communication and Technologies (Interspeech/Eurospeech)*, pages 861–864, Lisbon (Portugal), September 2005. ISCA.

[187] O. Pietquin, L. Couvreur, and P. Couvreur. Applied clustering for automatic speaker-based segmentation of audio materials. *Journal of Operations Research, Statistics and Computer Science (JORBEL now 4OR)*, 41(1-2) :1–12, 2001. - Impact Factor 1.089.

[188] O. Pietquin and T. Dutoit. Modélisation d'un système de reconnaissance dans le cadre de l'evaluation et l'optimisation automatique des systèmes de dialogue. In *Actes des Journées d'Etude de la Parole, JEP 2002*, pages 281–284, Nancy (France), June 2002.

[189] O. Pietquin and T. Dutoit. Aided Design of Finite-State Dialogue Management Systems. In *Proceedings of the 4th IEEE International Conference on Multimedia and Expo (ICME 2003)*, volume III, pages 545–548, Baltimore (USA, MA), July 2003.

[190] O. Pietquin and T. Dutoit. Dynamic Bayesian Networks for NLU Simulation with Application to Dialog Optimal Strategy Learning. In *Proceedings of the 31st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, volume I, pages 49–52, Toulouse (France), May 2006.

[191] O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2) :589–599, March 2006. Impact Factor 1.848.

[192] O. Pietquin, H. Frezza-Buet, and P. Crook. New frameworks for generalization, with implementation in DIPPER ISU DM system. CLASSiC Project Deliverable 1.2, October 2009.

[193] O. Pietquin, H. Frezza-Buet, and J.-L. Gutzwiller. Online speaker diarization with a size-monitored growing neural gas algorithm. *Neural Processing Letters*, page 12 pages, 2010. under review.

[194] O. Pietquin, M. Geist, and S. Chandramohan. Sample Efficient On-line Learning of Optimal Dialogue Policies with Kalman Temporal Differences. In *International Joint Conference on Artificial Intelligence (IJCAI 2011)*, Barcelona, Spain, July 2011. to appear.

[195] O. Pietquin, M. Geist, S. Chandramohan, and H. Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 2011. 21 pages - accepted for publication - Impact Factor 2.214.

[196] O. Pietquin and H. Hastie. Metrics for the evaluation of user simulation. CLASSiC Project Deliverable 3.5, March 2010.

[197] O. Pietquin and H. Hastie. A survey on metrics for the evaluation of user simulations. *Knowledge Engineering Review*, 2011. 15 pages - accepted for publication - Impact Factor 1.611.

[198] O. Pietquin and O. Lemon, editors. *Data Driven Methods for Adaptive Spoken Dialogue Systems*. Springer, 2011. accepted for publication.

[199] O. Pietquin and V. Philomin. Method of determining motion-related features and method of performing motion classification, November 2008. Patent n° WO2008139399.

[200] O. Pietquin and S. Renals. ASR System Modeling For Automatic Evaluation And Optimization of Dialogue Systems. In *Proceedings of the 27th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, volume I, pages 45–48, Orlando, (USA, FL), May 2002.

[201] O. Pietquin, S. Rossignol, and M. Ianotto. Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proceedings of the 1rst International Workshop on Spoken Dialogue Systems Technology (IWSDS 2009)*, Irsee (Germany), December 2009. 4 pages.

[202] O. Pietquin, S. Rossignol, M. Ianotto, and P. Crook. Probabilistic user simulations for training in the (PO)MDP framework including the simulation of grounding in TownInfo dialogues. CLASSiC Project Deliverable 3.2, October 2009.

[203] O. Pietquin and F. Tango. Batch reinforcement learning for optimizing driving assistance strategies. In *NIPS Workshop on Learning and Planning from Batch Time Series Data*, Vancouver (Canada), 2011.

[204] O. Pietquin, F. Tango, and R. Aras. Batch Reinforcement Learning for Optimizing Longitudinal Driving Assistance Strategies. In *Proceedings of the IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS 2011)*, April 2011. 8 pages.

[205] M. L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming*. Wiley-Interscience, April 1994.

[206] L. Rabiner. *Digital Processing of Speech Signal*. Prentice Hall, 1978.

[207] A. J. Raper, D. W. Richardson, H. A. Kontos, and J. L. Patterson. Circulatory Responses to Breath Holding in Man. *Journal of Applied Physiology*, 22(2) :201–206, 1967.

[208] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[209] A. Raux, B. Langner, D. Bohus, A. Black, and M. Eskenazi. Let's go public ! taking a spoken dialog system to the real world. In *Proc. of Interspeech 2005*, 2005.

[210] V. Rieser. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data*. PhD thesis, Saarland University, Dpt of Computational Linguistics, July 2008.

[211] S. Rossignol, S. Janarthanam, X. Liu, O. Pietquin, and M. Ianotto. User simulations of different types of Appointment Scheduling and Self-Help user. CLASSiC Project Deliverable 3.6, January 2011.

[212] S. Rossignol and O. Pietquin. Precise Voicing Information Extraction in Speech Signals Using the Analytic Signal. In *Proceedings of the 8th IEEE Symposium on Signal Processing and Information Technology (ISSPIT 2008)*, pages 207–212, Sarajevo (Bosnia & Herzegovina), December 2008.

[213] S. Rossignol and O. Pietquin. Single-speaker/multi-speaker co-channel speech classification. In *Proceedings of the International Conference on Speech Communication and Technologies (Interspeech 2010)*, pages 2322–2325, Makuhari (Japan), September 2010. ISCA.

[214] S. Rossignol, O. Pietquin, and M. Ianotto. Grounding simulation in spoken dialog systems with bayesian networks. In G. G. Lee, J. Mariani, W. Minker, and S. Nakamura, editors, *Spoken Dialogue Systems for Ambient Environments*, volume 6392 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 110–121, Gotemba, Shizuoka (Japan), October 2010. Springer-Verlag, Heidelberg-Berlin. Proceedings of the 2nd International Workshop on Spoken Dialogue Systems (IWSDS 2010).

[215] S. Rossignol, O. Pietquin, and M. Ianotto. Simulation du processus de croyance mutuelle de la compréhension dans le dialogue (grounding process) à l'aide des réseaux bayésiens. In *Actes des Journées d'Etude de la Parole (JEP 2010)*, Mons (Belgium), May 2010. 5 pages.

[216] S. Rossignol, O. Pietquin, and M. Ianotto. Training a BN-based user model for dialogue simulation with missing data. In *Proceedings of the Intenational Conference on Acousctics, Speech and Signal Processing (ICASSP 2011)*, Prague (Tchec Republic), May 2011. submitted.

[217] J. Rougui, M. Rziza, D. Aboutajdine, M. Gelgon, and J. Martinez. Fast incremental clustering of gaussian mixture speaker models for scaling up retrieval in on-line broadcast. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, Toulouse, 2006.

[218] N. Roy, J. Pineau, and S. Thrun. Spoken dialogue management using probabilistic reasoning. In *Proceedings of the annual meetinf of the Association for Computational Linguistics (ACL'00)*, pages 93–100, Morristown, NJ, USA, 2000.

[219] S. Russell. Learning agents for uncertain environments (extended abstract). In *COLT' 98 : Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103, New York, NY, USA, 1998. ACM.

[220] M. Saidi, O. Pietquin, and R. André-Obrecht. EMD decomposition to discriminate nasal vs. oral vowels in French . In *Proceedings of the 13th International conference on Speech and Computer (SPECOM 2009)*, St Petersburg (Russia), June 2009. 5 pages.

[221] M. Saidi, O. Pietquin, and R. André-Obrecht. Application of the EMD decomposition to discriminate nasalized vs. vowels phones in French. In *Proceedings of the International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA 2010)*, pages 128–132, Innsbruck (Austria), February 2010. ACTA Press.

[222] R. Sameni, M. B. Shamsollahi, and C. Jutten. Multi-Channel Electrocardiogram Denoising using a Bayesian Filtering Framework. In *Comp. Cardio.*, volume 33, pages 185–188, September 2006.

[223] R. Sameni, M. B. Shamsollahi, and C. Jutten. Model-based Bayesian Filtering of Cardiac Contaminants from Biomedical Recordings. *Phys. Meas.*, 29 :595–613, 2008.

[224] R. Sameni, M. B. Shamsollahi, C. Jutten, and G. D. Clifford. A Nonlinear Bayesian Filtering Framework for ECG Denoising. *IEEE Trans. Biomed. Eng.*, 54 :2172–2185, 2007.

[225] O. Sayadi and M. B. Shamsollahi. ECG Denoising and Compression Using a Modified Extended Kalman Filter Structure. *IEEE Trans. Biomed. Eng.*, 55 :2240–2248, 2008.

[226] D. J. Schaefer, J. D. Bourland, and J. A. Nyenhuis. Review of Patient Safety in Time-Varying Gradient Fields. *J. of Magn. Res. Imag.*, 12 :20–29, 2000.

[227] J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young. Effects of the user model on simulation-based learning of dialogue strategies. In *Proceedings of workshop on Automatic Speech Recognition and Understanding (ASRU'05), San Juan, Puerto Rico*, December 2005.

[228] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review*, 2006.

[229] Schiller, 2009.

[230] D. Schneegaß, S. Udluft, and T. Martinetz. Improving optimality of neural rewards regression for data-efficient batch near-optimal policy identification. In J. M. de Sá, L. A. Alexandre, W. Duch, and D. P. Mandic, editors, *ICANN*, volume 4668 of *Lecture Notes in Computer Science*, pages 109–118. Springer, 2007.

[231] R. Schoknecht. Optimality of Reinforcement Learning Algorithms with Linear Function Approximation. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS 2002)*, 2002.

[232] A. D. Scott, J. Keegan, and D. N. Firmin. Motion in Cardiovascular MR Imaging. *Radiology*, 250 :331–351, 2009.

[233] F. G. Shellock. *Magnetic Resonance Procedures : Health effects and Safety*. CRC Press, 2001.

[234] O. Sigaud and O. Buffet. *Processus Décisionnels de Markov en Intelligence Artificielle - Tome 1 : Principes Généraux et Applications*. Lavoisier, 2008.

[235] D. Simon. *Optimal State Estimation : Kalman, H Infinity, and Nonlinear Approaches*. Wiley & Sons, 1. auflage edition, August 2006.

[236] O. P. Simonetti, J. P. Finn, R. D. White, G. Laub, and D. A. Henry. Black-blood T_2-weighted inversion-recovery MR imaging of the heart. *Radiology*, 199 :49–57, 1996.

[237] S. Singh, M. Kearns, D. Litman, and M. Walker. Reinforcement learning for spoken dialogue systems. In *Proceedings of the Annual meeting of the Neural Iniformation Processing Society (NIPS'99), Denver, USA*. Springer, 1999.

[238] S. Stevens, J. Volkman, and E. Newman. A scale for the measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8(3) :185–190, 1937.

[239] A. L. Strehl and M. L. Littman. An Analysis of Model-Based Interval Estimation for Markov Decision Processes. *Journal of Computer and System Sciences*, 2006.

[240] M. Strens. A Bayesian Framework for Reinforcement Learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 943–950. Morgan Kaufmann, San Francisco, CA, 2000.

[241] J.-F. Sulzer, J.-P. Guignard, J.-F. Seignole, F. Sedes, O. Pietquin, R. Pinchuk, S. Moens, D. Milan, N. Allezard, P. Moreira, C. Fernandez, and A.-M. Manzat. Preliminary Requirement Specification (generic and applicative). LINDO deliverable D2.1, 2008.

[242] R. Sutton and A. Barto. *Reinforcement Learning : An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 3rd edition, March 1998.

[243] T. Söderström and P. Stoica. Instrumental variable methods for system identification. *Circuits, Systems, and Signal Processing*, 21 :1–9, 2002.

[244] F. Tango, M. Alonso, M. H. Vega, R. Aras, and O. Pietquin. A Reinforcement Learning approach for designing and optimizing interaction strategies for a Human-Machine Interface of a Partially Autonomous Driver Assistance System. In *Proceedings of the Workshop on Human Modelling in Assisted Transportation (HMAT 2010)*, Belgirate (Italy), June 2010. Springer Verlag, Heidelberg - Berlin. to appear.

[245] F. Tango, R. Aras, and O. Pietquin. Learning Optimal Control Strategies from Interactions for a Partially Autonomous Driver Assistance System. In *Proceedings of the Workshop on Human Modelling in Assisted Transportation (HMAT 2010)*, Belgirate (Italy), June 2010. Springer Verlag, Heidelberg - Berlin. to appear.

[246] F. Tango, L. Minin, R. Aras, and O. Pietquin. Automation Effects on Driver's Behaviour when integrating a PADAS and a Distraction Classifier. In *Proceedings of the Intenational Conference on Human-Computer Interfaces (HCI 2011)*, Orlando (FL, USA), July 2011. 10 pages - Invited Paper.

[247] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of two samples. *Biometrika*, 25 :285–294, 1933.

[248] E. Thorndike. *Educational psychology : the psychology of learning*. Teachers College Press, New York, 1913.

[249] R. van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, Portland, OR, USA, April 2004.

[250] R. van der Merwe and E. Wan. Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. In *Proceedings of the Workshop on Advances in Machine Learning*, Montreal, Canada, June 2003.

[251] S. Vijayakumar and S. Schaal. Local adaptive subspace regression. *Neural Processing Letters*, 7 :139–149, 1998.

[252] M. Walker, D. Hindle, J. Fromer, G. D. Fabbrizio, and C. Mestel. Evaluating competing agent strategies for a voice email agent. In *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech'97)*, Rhodes (Greece), 1997.

[253] C. J. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, England, 1989.

[254] R. E. Wendt, R. Rokey, G. W. Vick, and D. L. Johnston. Electrocardiographic Gating and Monitoring in NMR Imaging. *Magn. Res. Imag.*, 6 :89–95, 1988.

[255] J. D. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech Language*, 2007.

[256] H. Yu and D. P. Bertsekas. Q-Learning Algorithms for Optimal Stopping Based on Least Squares. In *Proceedings of European Control Conference*, Kos, Greece, 2007.

[257] A. Zijdenbos, B. Dawant, R. Margolin, and A. Palmer. Morphometric analysis of white matter lesions in MR images : method and validation. *IEEE Transactions on Medical Imaging*, 13(4) :716–24, 1994.

[258] F. G. Zoellner, R. Sance, P. Rogelj, M. Ledesma-Carbayo, J. Roervik, A. Santos, and A. Lundervold. Assessment of 3D DCE-MRI of the kidneys using non-rigid image registration and segmentation of voxel time courses. *Computerized Medical Imaging and Graphics*, 33 :171–181, 2009.