



HAL
open science

Méthodes d'entraînement pour l'analyse de la stabilité d'un système complexe

Benjamin Bocquillon

► **To cite this version:**

Benjamin Bocquillon. Méthodes d'entraînement pour l'analyse de la stabilité d'un système complexe. Automatique. Université Paris-Saclay, 2024. Français. NNT : 2024UPASG014 . tel-04573933

HAL Id: tel-04573933

<https://theses.hal.science/tel-04573933v1>

Submitted on 13 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Méthodes d'entraînement pour l'analyse de la stabilité d'un système complexe

*Training methods for analyzing the stability of a
complex system*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la
Communication (STIC)
Spécialité de doctorat : Automatique
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche **Laboratoire des Signaux et Systèmes
(Université Paris-Saclay, CNRS, CentraleSupélec)**, sous la direction de **Guillaume
SANDOU**, professeur au sein du Laboratoire des Signaux et Systèmes, le co-encadrement
de **Pedro RODRIGUEZ-AYERBE**, professeur au sein du Laboratoire des Signaux et
Systèmes et la co-supervision de **Philippe FEYEL**, docteur au sein de Safran Electronics &
Defense

Thèse soutenue à Paris-Saclay, le 25 mars 2024, par

Benjamin BOCQUILLON

Composition du jury

Membres du jury avec voix délibérative

Cristina STOICA Professeur, Université Paris-Saclay, CNRS, CentraleSupélec, L2S	Présidente
Joseph HAGGÈGE Professeur, Ecole Nationale d'Ingénieurs de Tunis (ENIT)	Rapporteur
Mohamed BECHERIF Professeur Associé, Université de technologie de Belfort Montbéliard (UTBM)	Rapporteur
Spilios THEODOULIS Professeur Associé, Université de Technologie (TU Delft)	Examineur

Remerciements

La réalisation de cette thèse a été rendue possible grâce au soutien de plusieurs personnes à qui je témoigne toute ma gratitude pour leur présence et leur contribution tout au long de ce parcours.

J'exprime tout d'abord ma reconnaissance aux rapporteurs Joseph Haggège et Mohamed Becherif, pour le temps précieux qu'ils m'ont accordé, pour leurs critiques constructives et pour leur expertise qui ont contribué à l'avancement de mon travail.

Je souhaite exprimer ma gratitude à Cristina Stoica, qui a non seulement présidé mon jury de thèse avec distinction mais a également enrichi mon travail par ses retours en tant qu'examinatrice.

Un grand merci à Spilios Theodoulis pour avoir examiné ma thèse avec rigueur, ainsi que pour ses retours éclairés, essentiels pour améliorer la qualité de mon manuscrit.

Je remercie chaleureusement Guillaume Sandou, mon directeur de thèse, pour son encadrement, sa patience et sa perspicacité. Son accompagnement m'a permis d'avancer et de progresser avec confiance et détermination à travers les complexités de la recherche.

Je suis profondément reconnaissant envers Philippe Feyel et Pedro Rodriguez-Ayerbe, mes tuteurs, pour leur soutien inestimable et leurs conseils avisés qui m'ont servi de piliers tout au long de mon parcours doctoral.

L'Université Paris-Saclay, le Laboratoire des Signaux et Systèmes de CentraleSupélec, et Safran Electronics & Defense méritent également toute ma reconnaissance pour avoir créé un environnement de recherche stimulant. Un remerciement particulier à François Guillot pour son appui et sa bienveillance qui ont guidé mon travail.

Je souhaite également remercier chaleureusement tous mes collègues de Safran Electronics & Defense ainsi que ceux du Laboratoire des Signaux et Systèmes, en particulier Jérémy, Maxime, Nicolas et Miguel. Leur soutien, leur expertise et surtout leur esprit de collaboration ont été essentiels dans mon parcours. Chaque jour passé avec eux a été une occasion d'apprendre et de se motiver mutuellement.

J'adresse des remerciements à mes amis pour leur soutien indéfectible et les moments joyeux partagés, qui ont souvent apporté lumière et réconfort lors des périodes de doute. Un merci particulier à Baptiste qui a été à l'origine du lien avec mon tuteur Philippe.

Un grand merci à Capucine, ma compagne, pour sa présence constante, son soutien sans faille et ses encouragements qui ont été le socle de mes avancées.

Enfin, je souhaite remercier ma famille, les personnes les plus chères à mon cœur, qui ont joué un rôle crucial dans cette aventure. Leur accompagnement, leurs encouragements constants et leur grande source d'inspiration ont été déterminants. Chaque personne a été un refuge dans les moments de doute. Leur présence et leur amour ont été les fondations sur lesquels j'ai pu bâtir cette thèse.

Je dois une reconnaissance toute particulière à mes parents. La qualité de ce manuscrit est le résultat direct de leur aide précieuse. Ils ont été mes premiers et mes plus importants professeurs. Ils m'ont appris la valeur de la persévérance et ont cru en moi à chaque étape de ce doctorat.

Table des matières

Remerciements	I
Table des matières	II
Liste des figures	V
Liste des tableaux	VII
Liste des algorithmes	VIII
Acronymes	IX
Introduction générale	1
1 Notions de stabilité	6
1.1 Introduction	8
1.2 Rappels théoriques	9
1.2.1 Gradient	9
1.2.2 Jacobienne	9
1.2.3 Hessienne	9
1.2.4 Norme Euclidienne	9
1.2.5 Boule fermée	10
1.2.6 Propriétés de positivité et de négativité des fonctions	10
1.2.7 Fonctions de classe \mathcal{K} , \mathcal{K}^∞ , \mathcal{KL}	10
1.2.8 Modèles non linéaires	10
1.2.9 Systèmes à temps discret	11
1.2.10 Équilibre	11
1.2.11 Changement d'origine	12
1.3 Quelques outils usuels pour l'étude de la stabilité	13
1.3.1 Passivité	13
1.3.2 Systèmes de Lure	16
1.3.3 Critère de Popov	18
1.3.4 Critère du Cercle	18
1.4 Définitions de stabilité au sens de Lyapunov	21
1.4.1 Définition intuitive de la stabilité	21
1.4.2 Stabilité au sens de Lyapunov	22
1.4.3 Stabilité asymptotique	23
1.4.4 Stabilité exponentielle	23
1.4.5 Stabilité globale	24
1.5 Méthode directe de Lyapunov	25

1.5.1	Introduction	25
1.5.2	Cas des systèmes à temps continu	26
1.5.3	Stabilité entrée - état	29
1.5.4	Cas des systèmes à temps discret	31
1.5.5	Domaine d'attraction	31
1.5.6	Level set	32
1.5.7	Systèmes inter-connectés	33
1.5.8	Instabilité	35
1.6	Récapitulatif et perspectives	36
2	Littérature existante	38
2.1	Introduction	39
2.2	Méthodes théoriques	40
2.2.1	Méthode de linéarisation	40
2.2.2	Méthode de Krasovski	42
2.2.3	Méthode du principe d'invariance de LaSalle	43
2.3	Méthodes numériques	45
2.3.1	Méthode du gradient variable	46
2.3.2	Méthode de Zubov	47
2.3.3	Méthode de la somme des carrés	51
2.3.4	Méthode continue et affine par morceaux	54
2.3.5	Méthode de la fonction de base radiale	56
2.3.6	Synthèse des méthodes numériques	59
2.4	Méthodes automatiques	60
2.4.1	Réseaux de neurones	61
2.4.2	Entraînement par métaheuristiques	65
2.4.3	Domaine de stabilité garanti	71
2.4.4	Fonction de Lyapunov fondée sur les réseaux de neurones	72
2.5	Synthèse des méthodes	74
3	Recherche automatique d'une fonction de Lyapunov par optimisation	76
3.1	Introduction	78
3.2	Contexte	79
3.2.1	Expression de H^V	80
3.2.2	Expression de H^G	80
3.3	Cas typiques	82
3.3.1	Forme quadratique	82
3.3.2	Réseau de neurones	83
3.4	Mise en œuvre de l'optimisation	87
3.4.1	Cadre d'optimisation	87
3.4.2	Définition du maillage	87
3.4.3	Estimation d'un DOA	88
3.4.4	Fonction de coût	90
3.4.5	Robustesse au maillage	95
3.5	Différentes formes de stabilité	99
3.5.1	Stabilité exponentielle	99
3.5.2	Stabilité entrée - état	101
3.6	Extension à d'autres problèmes	103

3.6.1	Systèmes à temps discret	103
3.6.2	Instabilité	109
3.7	Méthode générique	111
3.7.1	Rappels théoriques	111
3.7.2	Expression de H^V	112
3.7.3	Expression de H^G	112
3.7.4	Application aux différents problèmes de stabilité	114
3.8	Récapitulatif et perspectives	115
4	Études détaillées et perspectives	117
4.1	Introduction	119
4.2	Validations et applications pratiques	120
4.2.1	Stabilité asymptotique	120
4.2.2	Instabilité	124
4.2.3	Systèmes à temps discret	125
4.2.4	Stabilité exponentielle	128
4.2.5	Stabilité entrée - état	129
4.3	Performance des approches	133
4.3.1	Performance de notre approche	133
4.3.2	Notre approche face à l'état de l'art	135
4.4	Enjeux d'une application industrielle concrète	138
4.4.1	Modélisation du système	138
4.4.2	Perspectives et collaborations	142
	Conclusion générale	145
	Bibliographie	147

Table des figures

Chapitre 1

1.1	Système de commande en boucle fermée	13
1.2	Configuration d'un système de Lure	16
1.3	Conditions locales et globales du secteur (G. CHEN, 2004)	17
1.4	Disque $D[-(1/\alpha), -(1/\beta)]$ (G. CHEN, 2004)	19
1.5	Système linéaire avec saturation en entrée	19
1.6	Critère du Cercle	20
1.7	Critère de Popov	20
1.8	Illustration de la stabilité au sens de Lyapunov	22
1.9	Système inter-connecté	34

Chapitre 2

2.1	Architecture d'un réseau de neurones artificiels	61
2.2	Organigramme d'un algorithme génétique typique	67
2.3	Organigramme de l'optimisation par essaim de particules	69
2.4	Agent sous forme d'arbre représentant $\left[\cos(\text{rand}) + \frac{ X }{2.3} \right]$	70

Chapitre 3

3.1	Réseau neuronal de la fonction de Lyapunov	84
3.2	Visualisation du domaine de stabilité estimé garanti	89
3.3	Variation de V en fonction du maillage	96
3.4	Variation de \dot{V} en fonction du maillage	96

Chapitre 4

4.1	Fonction de Lyapunov construite pour $x_e = [0; 0]$	122
4.2	Dérivée temporelle de la fonction de Lyapunov construite	122
4.3	Domaine de stabilité estimé garanti pour le point d'équilibre	123
4.4	Fonction neuronale de Lyapunov construite pour $x_e = [0; 0]$	123
4.5	Dérivée temporelle de la fonction de Lyapunov construite	123
4.6	Fonction de Chetaev construite pour $x_e = [\pi; 0]$	125
4.7	Dérivée temporelle de la fonction de Chetaev construite	125
4.8	Domaine de stabilité garanti estimé pour $x_e = [0; 0]$	127
4.9	Domaine de stabilité garanti associé à $\tau = 9$	129

4.10 Domaine de stabilité garanti associé à $\tau = 4$ 129
4.11 Vérification temporelle sur Simulink 130
4.12 Domaine de stabilité garanti estimé pour $\Gamma \in [0; 10]$ 132
4.13 Vérification temporelle sur Simulink 132
4.14 Système masse-ressort 138
4.15 Axe simplifié 139
4.16 Système industriel 141

Liste des tableaux

Chapitre 2

2.1 Avantages et inconvénients des méthodes numériques	59
--	----

Chapitre 4

4.1 Mesure de performance de l'algorithme pour différentes formes de la fonction de Lyapunov	135
4.2 Comparaison des méthodes sur le plus grand level set et le temps de convergence . .	136

Liste des algorithmes

Chapitre 1

1	Pseudo-code pour la détermination du plus grand c	33
---	---	----

Chapitre 3

2	Pseudo-code pour le calcul de la fonction de coût Q avec la méthode 1	92
3	Pseudo-code pour le calcul de la fonction de coût Q avec la méthode 2	93
4	Pseudo-code pour le calcul de la fonction de coût Q avec ajustement des contraintes .	95
5	Pseudo-code pour le calcul de la fonction de coût Q avec la méthode 1, incluant la robustesse au maillage	97
6	Pseudo-code pour le calcul de la fonction de coût Q pour la stabilité exponentielle avec la méthode 2	100
7	Pseudo-code pour la détermination d'un point d'équilibre instable	110

Acronymes

- ADAM** Adaptive Moment Estimation. [65](#)
- AI** Artificial Intelligence. [42](#)
- ANN** Artificial Neural Network. [63](#)
- BIBO** Bounded Input / Bounded Output. [18](#)
- CNN** Convolutional Neural Network. [64](#)
- CPA** Continuous Piecewise Affine. [57](#)
- DL** Deep Learning. [54](#)
- DNN** Deep Neural Network. [64](#)
- DOA** Domain Of Attraction. [30](#), [62](#), [78](#), [118](#)
- EA** Evolutionary Algorithm. [70](#)
- FFNN** Feed-Forward Neural Network. [64](#)
- GA** Genetic Algorithm. [68](#), [90](#), [115](#)
- GAN** Generative Adversarial Network. [65](#)
- GE** Grammatical Evolution. [73](#)
- GP** Genetic Programming. [72](#)
- IQC** Integral Quadratic Constraints. [139](#)
- ISS** Input-to-State Stable. [34](#), [62](#), [115](#), [119](#)
- LMI** Linear Matrix Inequality. [74](#)
- LP** Linear Programming. [58](#)
- MIMO** Multiple Inputs / Multiple Outputs. [21](#)
- NN** Neural Network. [63](#), [119](#)
- PDE** Partial Differential Equation. [59](#)
- PSO** Particle Swarm Optimization. [68](#), [70](#), [90](#), [115](#)
- RBF** Radial Basis Function. [59](#)
- RNN** Recurrent Neural Network. [64](#)
- SDP** Semidefinite programming. [55](#)
- SGD** Stochastic Gradient Descent. [65](#)
- SISO** Single Input / Single Output. [21](#)
- SOS** Sum Of Squares. [54](#)
- TA** Threshold Algorithm. [73](#)

Introduction générale

Introduction générale

La réalisation de cette thèse a été rendue possible grâce à une convention CIFRE, établie dans le cadre d'une collaboration associant l'Université Paris-Saclay, le Laboratoire de Signaux et Systèmes de CentraleSupélec et l'entreprise Safran Electronics & Defense.

Cette thèse prolonge une ligne de recherches amorcée il y a une douzaine d'années. Elle est centrée sur "l'optimisation de la commande robuste et prédictive à l'aide de techniques d'optimisation modernes et d'intelligence artificielle". Cette trajectoire académique a donné lieu à plusieurs travaux significatifs, dont cinq thèses déjà présentées, qui explorent diverses facettes de l'application industrielle de méthodes d'optimisation contemporaines et d'intelligence artificielle dans le développement de lois de commande. L'objectif de ces recherches est de développer des contrôleurs à l'aide de méthodologies utilisées en ligne et hors ligne, qui sont non seulement sophistiqués mais également capables de répondre à des cahiers des charges détaillés pour des systèmes complexes. Ces thèses, mentionnées précédemment, fournissent un cadre détaillé des premières avancées et explorations dans ce domaine ; les lecteurs sont encouragés à les examiner pour une compréhension plus approfondie (FEYEL, 2015 ; FRASNEDO, 2016 ; HIRWA, 2013 ; PINGUET, 2023 ; POUILLY-CATHELAÏN, 2020).

Dans le domaine de la recherche scientifique appliquée à l'automatique, ces dernières années ont été témoins de transformations technologiques substantielles, impulsées en grande partie par l'éclosion de technologies novatrices, l'intelligence artificielle en est un exemple prééminent. Cette dernière introduit dans l'écosystème de l'automatique un renouveau, ouvrant la voie à des avancées sans précédent en matière de "contrôle intelligent".

Le concept de "contrôle intelligent", ou "intelligent control" en anglais, représente un paradigme qui capitalise sur les technologies émergentes telles que l'apprentissage automatique ("Machine Learning" en anglais), les réseaux de neurones, les techniques d'optimisation à grande échelle et les méthodologies prédictives avancées. Grâce à ces outils, le contrôle intelligent élargit considérablement le spectre d'applications et les possibilités d'utilisation de l'automatique traditionnelle, en introduisant des mécanismes plus flexibles, adaptatifs et puissants.

Un enjeu crucial dans le développement et l'implémentation de contrôleurs intelligents réside dans la garantie de la stabilité en boucle fermée. Dans ce contexte, la stabilité n'est pas une notion abstraite mais une caractéristique fondamentale, indissociable de la fiabilité et de la performance des systèmes contrôlés. En termes simples, la stabilité d'un système en boucle fermée se réfère à sa capacité à maintenir ou à retrouver un état d'équilibre prévu ou désiré en réponse à différentes perturbations ou changements dans l'environnement opérationnel. La stabilité est fondamentale pour s'assurer que le contrôleur réagit de manière prévisible et contrôlée, sans dévier de manière imprévue ou non souhaitée, ce qui pourrait compromettre la fonctionnalité et la sécurité du système dans son ensemble.

En ce qui concerne la stabilité, il est impératif d'introduire les fonctions de Lyapunov, un concept-clé dans cette étude. Les fonctions de Lyapunov sont des outils mathématiques employés pour analyser la stabilité des systèmes dynamiques. En identifiant et en utilisant efficacement ces fonctions, les chercheurs et les ingénieurs peuvent démontrer et garantir la stabilité des boucles de contrôle

intelligentes. Ainsi, une direction de recherche particulièrement prometteuse et fructueuse consiste à élaborer des instruments et des méthodologies d'assistance pour identifier de manière automatique les fonctions de Lyapunov adaptées à chaque système. L'ambition est de fournir des garanties de stabilité sur des intervalles aussi larges et robustes que possible, offrant ainsi des bases solides pour le déploiement sécurisé et efficace de ces technologies de contrôle avancées.

La théorie de Lyapunov constitue un pilier fondamental dans l'examen de la stabilité des systèmes non linéaires. L'application directe de cette théorie, bien qu'efficace dans des cas simples tels que les systèmes linéaires, devient complexe et délicate lorsqu'il s'agit de systèmes non linéaires. Pour ces derniers, chaque cas est unique, nécessitant une approche sur mesure pour identifier une fonction de Lyapunov appropriée. Cette fonction doit couvrir un espace aussi large que possible, définissant ainsi le domaine de stabilité ou le domaine d'attraction. L'un des défis majeurs de ce processus est de trouver une fonction de Lyapunov qui soit la moins conservatrice possible, c'est-à-dire qui maximise la région d'attraction. Il s'agit d'une tâche souvent fastidieuse, avec une issue incertaine et un succès jamais garanti.

Dans le cadre de cette thèse, nous suggérons de recourir à des techniques modernes d'entraînement et d'optimisation comme leviers pour assurer la stabilité des boucles intelligentes. Ce travail implique la résolution de plusieurs problèmes techniques cruciaux afin de produire une analyse de stabilité qui soit la moins restrictive possible. Les défis techniques clés à surmonter dans la recherche d'une fonction de Lyapunov non conservatrice comprennent :

- La détermination de la structure optimale de la fonction de Lyapunov, qui est capitale pour son efficacité et sa précision.
- La maximisation du domaine de stabilité, qui est essentielle pour garantir la stabilité dans une gamme aussi large que possible de conditions et de situations.
- La minimisation du temps nécessaire pour la recherche, afin de rendre le processus d'analyse de stabilité à la fois rapide et efficace.

Il convient de souligner que cette thèse se focalise sur l'analyse de la stabilité d'un système donné, sans chercher à stabiliser le système. Cette thèse est structurée autour de plusieurs chapitres qui explorent progressivement les concepts, techniques et applications liés à la stabilité des systèmes dynamiques non linéaires.

Le [Chapitre 1](#), intitulé [Notions de stabilité](#), est dédié à une exploration approfondie de la notion de stabilité dans les systèmes dynamiques non linéaires. Il débute par des rappels théoriques essentiels pour établir une compréhension solide des concepts fondamentaux de stabilité, indispensable pour évaluer la fiabilité et la performance des systèmes dynamiques variés. Ce chapitre se poursuit par l'examen des méthodes traditionnelles d'analyse de stabilité. Bien que ces méthodes soient fondamentales, elles tendent à être conservatrices et ne suffisent pas toujours pour l'analyse de tous les systèmes dynamiques. Dans la continuité des méthodes d'analyse de stabilité, il est essentiel de reconnaître le travail de fond d'Alexandre Lyapunov. Ce dernier, en introduisant des définitions précises et des fonctions spécifiques, a fourni un cadre robuste, tout en étant flexible pour l'analyse de la stabilité. La présentation se poursuit par une exploration détaillée de la méthode directe de Lyapunov, un outil puissant pour évaluer la stabilité des systèmes non linéaires. La méthode, avec les fonctions de Lyapunov comme composants centraux, est présentée de manière à permettre une compréhension profonde de ses principes, applications et utilités dans divers contextes.

Le [Chapitre 2](#), dénommé [Littérature existante](#), débute par une exploration approfondie des différentes méthodes traitant de la stabilité des systèmes dynamiques, un domaine qui a connu de multiples révolutions au fil des années. Face à la complexité croissante des systèmes dynamiques,

particulièrement dans les applications industrielles, il devient essentiel de compléter les méthodologies traditionnelles en adoptant des approches qui tiennent compte des défis actuels. Notre étude approfondie dans l'état de l'art permet de retracer le parcours des méthodes d'analyse de stabilité, depuis les approches théoriques jusqu'aux techniques automatiques avancées. Cette étude n'est pas seulement informative mais aussi critique, identifiant les avantages et les limites de ces différentes méthodes. La suite de la réflexion s'oriente vers les méthodes automatiques, illustrant un changement significatif par rapport aux approches traditionnelles. S'appuyant sur les avancées de l'intelligence artificielle, ces techniques innovantes englobent des initiatives prometteuses, telles que la première "Lyapunov Machine". Cet examen approfondi permet de contextualiser et de positionner notre travail innovant dans le paysage de la recherche actuelle sur la stabilité des systèmes dynamiques.

Le [Chapitre 3, Recherche automatique d'une fonction de Lyapunov par optimisation](#), présente un schéma d'optimisation innovant afin de déterminer une fonction de Lyapunov automatiquement. Cette méthode est conçue pour identifier des fonctions de Lyapunov efficaces dans divers contextes. Un point central de notre méthode est de maximiser le domaine de stabilité garanti par la fonction. Cette optimisation caractérise notre approche, la rendant pertinente aux contraintes et spécificités du contexte industriel pour l'analyse de la stabilité. Au fur et à mesure du développement de notre méthode, sous la forme d'algorithme, il est devenu évident que notre approche d'optimisation pouvait être étendue et généralisée pour s'attaquer à une multitude de problèmes relatifs à la stabilité des systèmes dynamiques. Grâce au développement de diverses fonctionnalités, notre algorithme s'est transformé en une méthode flexible, applicable de manière générique dans plusieurs contextes. Cette généricité permet à notre approche de s'adapter à une variété de perspectives, rendant notre méthode applicable dans un large éventail de scénarios et de disciplines.

Le [Chapitre 4, intitulé Études détaillées et perspectives](#), procède à une évaluation approfondie des différentes méthodes de stabilité abordées dans ce manuscrit, mettant en lumière leurs avantages et leurs limites respectives à travers une série de tests et d'analyses. Cette évaluation critique vise à éclairer la pertinence et l'applicabilité de ces techniques dans différents contextes et à guider leur utilisation future. Ce chapitre débute par la présentation de divers tests de performance, couvrant une gamme étendue de situations, pour évaluer les différents critères de stabilité introduits. Cette démarche utilise différentes formes de fonctions de Lyapunov et d'algorithmes d'optimisation, fournissant ainsi un aperçu détaillé de leurs performances et applicabilités. Ensuite, une attention particulière est portée à l'examen de nos méthodes dans le contexte de la recherche actuelle. Des études comparatives sont conduites sur différents systèmes, offrant un aperçu sur les avantages et limites de notre méthode générique. Nous terminons par une analyse détaillée d'un système industriel typique, discutant des défis inhérents à l'étude de la stabilité de tels systèmes complexes et des stratégies envisagées pour les surmonter. Nous offrons ainsi une vision des perspectives d'avenir et des défis à relever pour l'application efficace de nos méthodes dans le monde industriel.

Liste des publications internationales

Les recherches menées durant le doctorat ont abouti aux publications suivantes, contribuant significativement au champ académique :

Publication 1 :

BOCQUILLON, B., FEYEL, P., SANDOU, G., & RODRIGUEZ-AYERBE, P. (2020a). Computation of neural networks Lyapunov functions for discrete and continuous time systems with domain of attraction maximization. *12th International Conference on Neural Computation Theory and Applications (NCTA)*

Publication 2 :

BOCQUILLON, B., FEYEL, P., SANDOU, G., & RODRIGUEZ-AYERBE, P. (2020b). Efficient construction of neural networks Lyapunov functions with domain of attraction maximization. *17th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*

Publication 3 :

BOCQUILLON, B., FEYEL, P., SANDOU, G., & RODRIGUEZ-AYERBE, P. (2022). A Comprehensive Framework to Determine Lyapunov Functions for a Set of Continuous Time Stability Problems. *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*, 1-6

Autres communications et activités

Outre les contributions aux publications internationales, cette période a également été marquée par une participation active à diverses manifestations scientifiques et académiques.

Journées des Doctorants de Safran Electronics & Defense

- Présentation à la Journée des Doctorants, Safran Electronics & Defense, 2019.
- Présentation à la Journée des Doctorants, Safran Electronics & Defense, 2020.
- Présentation à la Journée des Doctorants, Safran Electronics & Defense, 2021.
- Présentation à la Journée des Doctorants, Safran Electronics & Defense, 2022.

Journées des Doctorants de l'Université Paris-Saclay

- Intervenant à la Journée des Doctorants, Université Paris-Saclay, 2020.
- Intervenant à la Journée des Doctorants, Université Paris-Saclay, 2021.
- Intervenant à la Journée des Doctorants, Université Paris-Saclay, 2022.
- Intervenant à la Journée des Doctorants, Université Paris-Saclay, 2023.

Journée SYCOMORE du Laboratoire des Signaux et Systèmes de CentraleSupélec

- Intervenant à la Journée SYCOMORE, Laboratoire des Signaux et Systèmes, CentraleSupélec, 2022.

Notions de stabilité

Notions de stabilité

Sommaire

1.1	Introduction	8
1.2	Rappels théoriques	9
1.2.1	Gradient	9
1.2.2	Jacobienne	9
1.2.3	Hessienne	9
1.2.4	Norme Euclidienne	9
1.2.5	Boule fermée	10
1.2.6	Propriétés de positivité et de négativité des fonctions	10
1.2.7	Fonctions de classe \mathcal{K} , \mathcal{K}^∞ , \mathcal{KL}	10
1.2.8	Modèles non linéaires	10
1.2.9	Systèmes à temps discret	11
1.2.10	Équilibre	11
1.2.11	Changement d'origine	12
1.3	Quelques outils usuels pour l'étude de la stabilité	13
1.3.1	Passivité	13
1.3.1.1	Exemple d'illustration 1	15
1.3.2	Systèmes de Lure	16
1.3.3	Critère de Popov	18
1.3.4	Critère du Cercle	18
1.3.4.1	Exemple d'illustration 2	19
1.4	Définitions de stabilité au sens de Lyapunov	21
1.4.1	Définition intuitive de la stabilité	21
1.4.2	Stabilité au sens de Lyapunov	22
1.4.3	Stabilité asymptotique	23
1.4.4	Stabilité exponentielle	23
1.4.5	Stabilité globale	24
1.5	Méthode directe de Lyapunov	25
1.5.1	Introduction	25
1.5.2	Cas des systèmes à temps continu	26
1.5.2.1	Exemple d'illustration 3	27
1.5.2.2	Exemple d'illustration 4	28

1.5.2.3	Exemple d'illustration 5	29
1.5.3	Stabilité entrée - état	29
1.5.3.1	Exemple d'illustration 6	30
1.5.4	Cas des systèmes à temps discret	31
1.5.5	Domaine d'attraction	31
1.5.6	Level set	32
1.5.7	Systèmes inter-connectés	33
1.5.7.1	Exemple d'illustration 7	34
1.5.8	Instabilité	35
1.5.8.1	Exemple d'illustration 8	35
1.6	Récapitulatif et perspectives	36

1.1 Introduction

La stabilité est une notion essentielle pour comprendre et prévoir le comportement à long terme des systèmes dynamiques. Qu'il s'agisse de systèmes physiques, tels que les circuits électriques, les systèmes mécaniques ou les systèmes climatiques, de systèmes biologiques, comme les réseaux neuronaux ou les écosystèmes, ou de systèmes artificiels, comme les robots ou les réseaux informatiques, la stabilité est un critère clé pour évaluer la performance et la fiabilité de ces systèmes dynamiques.

Le présent chapitre se propose d'explorer en profondeur la notion de stabilité dans le contexte des systèmes dynamiques non linéaires. Une compréhension claire de cette thématique est fondamentale pour l'étude et le contrôle de nombreux systèmes physiques.

C'est dans ce contexte que le travail d'Alexandre Lyapunov prend toute son importance. Son approche a été fondamentale pour formaliser et comprendre la stabilité des systèmes dynamiques. En introduisant la notion de fonction de Lyapunov, il a fourni un outil puissant pour analyser la stabilité des systèmes non linéaires. Aujourd'hui, les fonctions de Lyapunov et la méthode directe de Lyapunov sont des pierres angulaires de l'analyse de stabilité.

Cependant, malgré les avancées considérables dans ce domaine, l'étude de la stabilité reste un sujet ouvert pour les automaticiens et les ingénieurs.

La nature non linéaire de nombreux systèmes dynamiques rend l'analyse de leur stabilité particulièrement complexe. En effet, il n'existe aucune méthode générale pour la recherche de fonctions de Lyapunov pour les systèmes de grande dimension ou les systèmes présentant des comportements complexes. Cette tâche est souvent chronophage et son succès n'est pas assuré.

En somme, l'étude de la stabilité offre un champ d'investigation riche et varié, qui continue de susciter l'intérêt des chercheurs en mathématiques, en ingénierie et dans de nombreux autres domaines. L'approche de Lyapunov, bien que vieille de plus d'un siècle, reste un pilier fondamental de ce domaine.

Dans ce chapitre, nous commençons par un rappel des concepts théoriques liés à la stabilité. Cette section sert de base pour comprendre les définitions plus avancées et les techniques d'analyse qui suivent.

Nous nous penchons ensuite sur l'examen de méthodes couramment utilisées pour étudier la stabilité des systèmes dynamiques. Nous abordons des méthodes permettant l'étude de stabilité qui fournissent une compréhension assez complète des comportements possibles des systèmes dynamiques. Nous étudions en profondeur les principes fondamentaux de ces méthodes, leurs applications concrètes et leurs utilités dans différents contextes. Cette partie nous permet d'appréhender également les limites de ces différentes notions.

C'est pourquoi nous nous tournons ensuite vers les définitions de la stabilité au sens de Lyapunov, un concept fondamental dans l'étude de la stabilité des systèmes dynamiques. La théorie de Lyapunov constitue le fondement sur lequel reposent de nombreuses méthodes d'analyse de la stabilité. Plutôt que de la considérer comme une alternative aux autres méthodes, nous reconnaissons ici son rôle central dans le domaine de la stabilité des systèmes dynamiques et les moyens de son application pour analyser avec précision le comportement de ces systèmes.

Enfin, la méthode directe de Lyapunov est le prochain sujet de discussion. Il s'agit d'un outil puissant et largement utilisé pour évaluer la stabilité des systèmes non linéaires. Nous explorons les fondamentaux de cette méthode, sa mise en application et son utilité dans divers contextes.

En résumé, ce chapitre fournit une fondation solide pour comprendre et analyser la stabilité des systèmes dynamiques non linéaires.

1.2 Rappels théoriques

Dans cette section, nous introduisons les notations et définitions que nous utilisons dans cette thèse.

1.2.1 Gradient

Pour une fonction à valeur réelle $f : \mathbb{R}^n \rightarrow \mathbb{R}$, nous désignons par :

$$\nabla f = (\nabla f)_x = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \quad (1.1)$$

le gradient de $f(x)$, qui est le vecteur colonne des dérivées partielles $\frac{\partial f}{\partial x_i}$ dans toutes les directions de $x = [x_1, \dots, x_n]^T$.

1.2.2 Jacobienne

Pour une fonction vectorielle $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, nous désignons par :

$$Jf = (Jf)_x = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (1.2)$$

la jacobienne de $f(x)$, qui est la matrice des dérivées partielles $\frac{\partial f_i}{\partial x_j}$.

1.2.3 Hessienne

Pour une fonction à valeur réelle $f : \mathbb{R}^n \rightarrow \mathbb{R}$, nous désignons par :

$$Hf = (Hf)_x = \frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (1.3)$$

la hessienne de $f(x)$, qui est la matrice des dérivées partielles secondes $\frac{\partial^2 f}{\partial x_i \partial x_j}$.

1.2.4 Norme Euclidienne

Soit $x = [x_1, \dots, x_n] \in \mathbb{R}^n$ un vecteur à valeur réelle. La norme :

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} \quad (1.4)$$

est la norme euclidienne de x .

1.2.5 Boule fermée

Soit un vecteur $x \in \mathbb{R}^n$, nous désignons par :

$$\mathcal{B}_\varepsilon(x) = \{y \in \mathbb{R}^n \mid \|x - y\| \leq \varepsilon\} \quad (1.5)$$

la boule fermée centrée sur x de rayon $\varepsilon \in \mathbb{R}^+$. Comme notation abrégée, nous utilisons \mathcal{B}_ε pour désigner la boule fermée $\mathcal{B}_\varepsilon(0)$ centrée sur 0 de rayon ε .

1.2.6 Propriétés de positivité et de négativité des fonctions

Soit $f : D \rightarrow \mathbb{R}$ une fonction, nous pouvons définir les relations suivantes :

- f est définie positive sur D si et seulement si $f(x) > 0$ et $(x = 0 \Rightarrow f(x) = 0)$ pour tout $x \in D$.
- f est semi-définie positive sur D si et seulement si $f(x) \geq 0$ pour tout $x \in D$.
- f est définie négative sur D si et seulement si $f(x) < 0$ et $(x = 0 \Rightarrow f(x) = 0)$ pour tout $x \in D$.
- f est semi-définie négative sur D si et seulement si $f(x) \leq 0$ pour tout $x \in D$.

1.2.7 Fonctions de classe \mathcal{K} , \mathcal{K}^∞ , \mathcal{KL}

Une fonction continue $\alpha : [0, a) \rightarrow [0, \infty)$ appartient à la classe \mathcal{K} si elle est strictement monotone croissante et $\alpha(0) = 0$. Elle appartient à la classe \mathcal{K}^∞ si $a = \infty$ et $\alpha(x) \rightarrow \infty$ lorsque $x \rightarrow \infty$.

Une fonction continue $\mathcal{B} : [0, \infty) \times [0, \infty) \rightarrow [0, \infty)$ appartient à la classe \mathcal{KL} si, pour chaque s fixé, l'application $\mathcal{B}(x, s)$ appartient à la classe \mathcal{K} par rapport à x , et pour chaque x fixé, l'application $\mathcal{B}(x, s)$ est décroissante par rapport à s et $\mathcal{B}(x, s) \rightarrow 0$ lorsque $s \rightarrow \infty$.

1.2.8 Modèles non linéaires

Nous traitons de systèmes dynamiques modélisés par un nombre fini d'équations différentielles ordinaires du premier ordre couplées :

$$\begin{cases} \dot{x}_1 = f_1(t, x_1, \dots, x_n, u_1, \dots, u_p) \\ \dot{x}_2 = f_2(t, x_1, \dots, x_n, u_1, \dots, u_p) \\ \vdots \\ \dot{x}_n = f_n(t, x_1, \dots, x_n, u_1, \dots, u_p). \end{cases} \quad (1.6)$$

Nous pouvons écrire l'équation différentielle vectorielle du premier ordre :

$$\dot{x} = f(t, x, u). \quad (1.7)$$

L'équation (1.7) est appelée équation d'état et réfère x comme *état* et u comme *entrée*.

L'équation d'état, sans entrée u , peut également être appelée équation d'état autonome à temps variant :

$$\dot{x} = f(t, x). \quad (1.8)$$

Le système est appelé système autonome à temps invariant lorsque la fonction f ne dépend pas explicitement de t , c'est-à-dire :

$$\dot{x} = f(x). \quad (1.9)$$

1.2.9 Systèmes à temps discret

Pour les systèmes à temps discret, la modélisation s'effectue de manière similaire, mais elle est représentée par des équations aux différences. Un système non linéaire à temps discret est généralement décrit par :

$$\begin{cases} x_{1_{k+1}} = g_1(k, x_{1_k}, \dots, x_{n_k}, u_{1_k}, \dots, u_{p_k}) \\ x_{2_{k+1}} = g_2(k, x_{1_k}, \dots, x_{n_k}, u_{1_k}, \dots, u_{p_k}) \\ \vdots \\ x_{n_{k+1}} = g_n(k, x_{1_k}, \dots, x_{n_k}, u_{1_k}, \dots, u_{p_k}) \end{cases} \quad (1.10)$$

où k est l'indice discret représentant des instants successifs.

De manière analogue, l'équation aux différences vectorielle est donnée par :

$$x_{k+1} = g(k, x_k, u_k) \quad (1.11)$$

où l'équation (1.11) est l'équation d'état discrète, avec x représentant l'état et u l'entrée à l'instant k .

Lorsqu'il n'y a pas d'entrée u , cela donne lieu à une équation d'état autonome discrète :

$$x_{k+1} = g(k, x_k). \quad (1.12)$$

Enfin, si la fonction g ne dépend pas explicitement de k , le système est dit autonome à temps discret invariant :

$$x_{k+1} = g(x_k). \quad (1.13)$$

1.2.10 Équilibre

Une notion primordiale dans l'étude de la stabilité est la notion de point d'équilibre. Commençons par rappeler sa définition pour le système autonome (1.9).

Définition 1 (Point d'équilibre) *Un point $x_e \in \mathbb{R}^n$ est dit point d'équilibre pour le système (1.9), si et seulement si (KHALIL, 2002) :*

$$f(x_e) = 0. \quad (1.14)$$

Pour un système dynamique donné, l'équilibre peut ne pas être unique.

Définition 2 (Point d'équilibre pour les systèmes discrets) *Un point $x_e \in \mathbb{R}^n$ est dit point d'équilibre pour le système à temps discret donné par (1.13), si et seulement si :*

$$f(x_e) = x_e. \quad (1.15)$$

Tout comme pour les systèmes continus, pour un système dynamique discret donné, le point d'équilibre peut ne pas être unique.

1.2.11 Changement d'origine

Dans la suite de la thèse, afin de simplifier les notations et sans perte de généralités, l'équilibre étudié est l'origine.

Pour justifier le propos, considérons le système autonome (1.9) et analysons un des points d'équilibre $x_e \in \mathbb{R}^n$. Notons $z = x - x_e$ le changement de variable. Nous pouvons alors écrire $\dot{z} = \dot{x} = f(x) = f(z + x_e)$. Si nous définissons $g(z) := f(z + x_e)$, le système devient, dans les nouvelles coordonnées :

$$\dot{z} = g(z). \tag{1.16}$$

Nous trouvons alors $g(0) = f(x_e) = 0$ (sachant que x_e est un point d'équilibre pour (1.9)).

1.3 Quelques outils usuels pour l'étude de la stabilité

Dans cette section, nous abordons divers méthodes et concepts essentiels pour l'étude de la stabilité des systèmes non linéaires. Ces méthodes sont d'une grande importance dans l'analyse des systèmes dynamiques. Il est pertinent de noter que la plupart des méthodes dont nous discutons ici découlent de l'étude de stabilité par Lyapunov, une approche fondamentale que nous présentons dans la section suivante. Cependant, ces méthodes ont leurs propres particularités et applications qui les rendent adaptées, et faciles d'emploi dans différents contextes particuliers.

Dans cette partie, nous définissons des méthodes usuelles pour l'étude de la stabilité. Il est important de noter que ces méthodes aboutissent à des critères spécifiques et dépendent de propriétés particulières du système. Toutefois, dans le contexte des systèmes dynamiques complexes, en particulier ceux rencontrés dans les applications industrielles, ces approches peuvent se révéler excessivement conservatrices, notamment lorsqu'il s'agit d'estimer des notions telles que le domaine d'attraction. Cette notion de domaine d'attraction est explorée plus en détail par la suite.

1.3.1 Passivité

Voici un système en boucle fermée, illustré par la [Figure 1.1](#) où les entrées, les sorties et les signaux internes sont liés par les équations suivantes :

$$\begin{cases} e_2 = u_2 + y_1 \\ e_1 = u_1 - y_2 \end{cases} \quad (1.17)$$

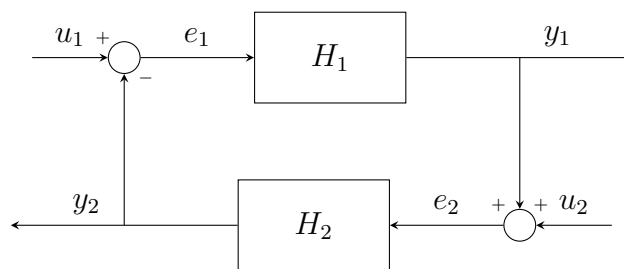


FIGURE 1.1 – Système de commande en boucle fermée

Dans certains cas, les systèmes non linéaires peuvent être représentés comme une interconnexion entre un système linéaire et une non-linéarité statique. Pour ce type d'interconnexion et lorsque la non-linéarité n'a pas de mémoire, la stabilité du système en boucle fermée peut être analysée en utilisant la réponse fréquentielle du système linéaire.

Il est important de noter que la stabilité entrée bornée / sortie bornée ([Bounded Input / Bounded Output \(BIBO\)](#)) individuelle de H_1 et H_2 n'est pas suffisante pour assurer la stabilité BIBO du système bouclé connecté.

Définition 3 (Système passif) Le système (KHALIL, 2002) :

$$\begin{cases} \dot{x} = f(x, u) \\ y = h(x, u) \end{cases}$$

où $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ est localement lipschitzienne, $h : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ est continue, $f(0, 0) = 0$ et $h(0, 0) = 0$, est dit passif si il existe une fonction continue, différentiable et semi-définie positive $V : \mathbb{R}^n \rightarrow \mathbb{R}^+$ (appelée fonction de stockage) telle que :

$$\dot{V} = \left(\frac{\partial V}{\partial x}(x) \right) f(x, u) \leq u^T y, \quad \forall (x, u) \in \mathbb{R}^n \times \mathbb{R}^m.$$

De plus, le système est :

- Sans perte si $\dot{V} = u^T y$.
- Strictement passif si $\dot{V} \leq u^T y - \psi(x)$ pour une fonction définie positive $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^+$.

Les théorèmes suivants fournissent des critères pour déterminer si un système ou une combinaison de systèmes est stable ou non, en se basant sur leurs propriétés de passivité. Il convient de noter que ces théorèmes s'appuient fortement sur les concepts de passivité et de stockage d'énergie (KHALIL, 2002).

Théorème 4 (Théorème de la passivité) La connexion en rétroaction de deux systèmes passifs est passive.

Théorème 5 (Théorème de la passivité) Considérons la connexion en rétroaction de deux systèmes dynamiques. Lorsque $u = 0$, l'origine du système en boucle fermée est asymptotiquement stable si chaque composant de rétroaction est :

- Soit strictement passif.
- Soit strictement passif en sortie et observable à l'état zéro.

De plus, si la fonction de stockage pour chaque composant est radialement non bornée, l'origine est globalement asymptotiquement stable.

Théorème 6 (Théorème de la passivité) Considérons la connexion en rétroaction d'un système dynamique strictement passif avec une fonction sans mémoire passive. Lorsque $u = 0$, l'origine du système en boucle fermée est uniformément asymptotiquement stable. Si la fonction de stockage pour le système dynamique est radialement non bornée, l'origine est globalement uniformément asymptotiquement stable.

1.3.1.1 Exemple d'illustration 1

Considérons les deux systèmes dynamiques décrits ci-dessous (KHALIL, 2002) :

$$H_1 : \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -ax_1^3 - kx_2 + e_1 \\ y_1 = x_2 \end{cases}$$

$$H_2 : \begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = -bx_3 - x_4^3 + e_2 \\ y_2 = x_4 \end{cases}$$

avec $a, b, k > 0$. Pour le système H_1 , nous pouvons définir une fonction V_1 et sa dérivée temporelle \dot{V}_1 comme suit :

$$V_1 = \frac{1}{4}ax_1^4 + \frac{1}{2}x_2^2$$

$$\dot{V}_1 = ax_1^3x_2 - ax_1^3x_2 - kx_2^2 + x_2e_1 = -ky_1^2 + y_1e_1$$

$$\text{avec } e_1 = 0, y_1(t) \equiv 0 \iff x_2(t) \equiv 0 \Rightarrow x_1(t) \equiv 0.$$

Alors H_1 est strictement passif en sortie et observable à l'état zéro. Pour le système H_2 , nous pouvons définir une fonction V_2 et sa dérivée temporelle \dot{V}_2 comme suit :

$$V_2 = \frac{1}{2}bx_3^2 + \frac{1}{2}x_4^2$$

$$\dot{V}_2 = bx_3x_4 - bx_3x_4 - x_4^4 + x_4e_2 = -y_2^4 + y_2e_2$$

$$\text{avec } e_2 = 0, y_2(t) \equiv 0 \iff x_4(t) \equiv 0 \Rightarrow x_3(t) \equiv 0.$$

H_2 est strictement passif en sortie et observable à l'état zéro. V_1 et V_2 sont radialement non bornées. L'origine est globalement asymptotiquement stable.

Considérons le système en boucle fermée suivant :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \\ u = -\psi(t, y) \end{cases} \quad (1.18)$$

où (A, B, C) est contrôlable et observable et ψ est une non-linéarité sans mémoire variant dans le temps satisfaisant certaines conditions de secteur. Nous souhaitons étudier la stabilité de l'origine pour toutes les non-linéarités dans le secteur donné. Si nous démontrons que l'origine du système en boucle fermée est globalement uniformément asymptotiquement stable pour toutes les non-linéarités dans le secteur donné, alors le système est dit être absolument stable. Ce problème est appelé le problème de Lure. Dans ce cas, la passivité permet d'obtenir un critère de stabilité, comme par exemple le critère de Popov ou le critère du Cercle que nous détaillons dans les prochaines sections.

1.3.2 Systèmes de Lure

Dans certains cas, les systèmes non linéaires peuvent être représentés comme une interconnexion entre un système linéaire et une non-linéarité statique. Considérons un système de rétroaction sous la forme de Lure :

$$\begin{cases} \dot{x} &= Ax + Bh(y) \\ y &= Cx \end{cases} \quad (1.19)$$

où A , B et C sont des matrices constantes, A une matrice non singulière, B et C non nécessairement carrées (bien que $B = C = I$ soit possible), et h est une fonction non linéaire à valeurs vectorielles. En effectuant la transformée de Laplace avec des conditions initiales nulles, et en désignant la transformée par $\hat{x} = \mathcal{L}\{x\}$, le vecteur d'état est obtenu comme suit :

$$\hat{x} = [sI - A]^{-1} B \mathcal{L}\{h(y)\} \quad (1.20)$$

et la sortie est donnée par :

$$\hat{y} = CG(s) \mathcal{L}\{h(y)\} \quad (1.21)$$

avec la matrice de transfert du système :

$$G(s) = [sI - A]^{-1} B. \quad (1.22)$$

Le système de Lure, illustré par la [Figure 1.2](#), est une configuration en boucle fermée, où la boucle de rétroaction est généralement considérée comme un "contrôleur". Ainsi, ce système est parfois écrit sous la forme équivalente suivante :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \\ u = h(y) \end{cases} \quad (1.23)$$

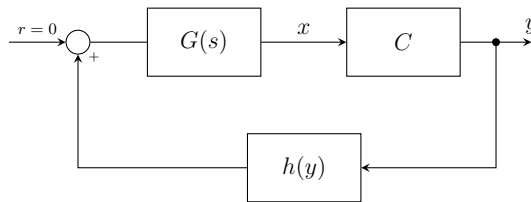


FIGURE 1.2 – Configuration d'un système de Lure

Dans ce mémoire, nous ne présentons que des résultats sur les systèmes possédant une seule entrée et une seule sortie (**Single Input / Single Output (SISO)**). Le lecteur peut avoir un aperçu des résultats pour des systèmes avec entrées et sorties multiples (**Multiple Inputs / Multiple Outputs (MIMO)**) dans ([G. CHEN, 2004](#)).

Les systèmes de Lure à une entrée et une sortie, où $u = h(y)$ et $y = c^T x$ sont des fonctions scalaires, sont représentés par :

$$\begin{cases} \dot{x} = Ax + bu \\ y = c^T x \\ u = h(y) \end{cases} \quad (1.24)$$

Supposons que $h(0) = 0$, de sorte que $x_e = 0$ soit un équilibre du système.

Définition 7 Si il existe deux constantes $\alpha < \beta$ telles que :

$$\alpha y(t)^2 \leq y(t)h(y(t)) \leq \beta y(t)^2 \quad (1.25)$$

pour tout $y(t)$ tel que $a \leq y(t) \leq b$ et $t \in [t_0; \infty]$, alors la condition de secteur locale sur la fonction non linéaire $h(\cdot)$ est satisfaite.

L'équation (1.25) exprime la condition de secteur locale, où la fonction non linéaire doit se situer entre les bornes $[\alpha; \beta]$ lorsque la valeur de y est comprise entre a et b et pour un temps t dans l'intervalle $[t_0; \infty]$. Cette condition locale est représentée sur la partie a) de la Figure 1.3.

Définition 8 Si il existe deux constantes $\alpha < \beta$ telles que :

$$\alpha y(t)^2 \leq y(t)h(y(t)) \leq \beta y(t)^2 \quad (1.26)$$

pour tout $y(t)$ tel que $-\infty \leq y(t) \leq \infty$ et $t \in [t_0; \infty]$, alors la condition de secteur globale sur la fonction non linéaire $h(\cdot)$ est satisfaite.

L'équation (1.26) quant à elle, représente la condition de secteur globale, où la fonction non linéaire doit se situer entre les bornes $[\alpha; \beta]$ pour toutes les valeurs de y et pour un temps t dans l'intervalle $[t_0; \infty]$. Cette condition globale est représentée sur la partie b) de la Figure 1.3.

Ces conditions de secteur sont d'une importance cruciale car elles caractérisent les comportements admissibles de la fonction non linéaire, influençant ainsi directement les propriétés de stabilité du système. Elles délimitent les régions dans lesquelles la fonction non linéaire peut opérer tout en garantissant que le système demeure stable. Lorsque la fonction non linéaire respecte ces conditions de secteur, il devient possible d'utiliser des techniques d'analyse spécifiques pour déduire la stabilité asymptotique du système autour de son équilibre $x_e = 0$. Les méthodes basées sur la passivité, les méthodes fréquentielles comme le critère de Popov ou le critère du Cercle et les méthodes basées sur les fonctions de Lyapunov sont quelques techniques d'analyse qui peuvent être employées.

Il est important de souligner que les méthodes basées sur les critères du Cercle et de Popov s'appuient sur la propriété affirmant que la mise en boucle de deux systèmes passifs aboutit à une boucle fermée stable. Dans cette optique, on utilise l'excès de passivité ou le défaut de passivité de la non-linéarité. Ceci permet de transformer la boucle en associant une non-linéarité passive à un système linéaire. Si ce système linéaire est passif, alors la boucle fermée reste stable.

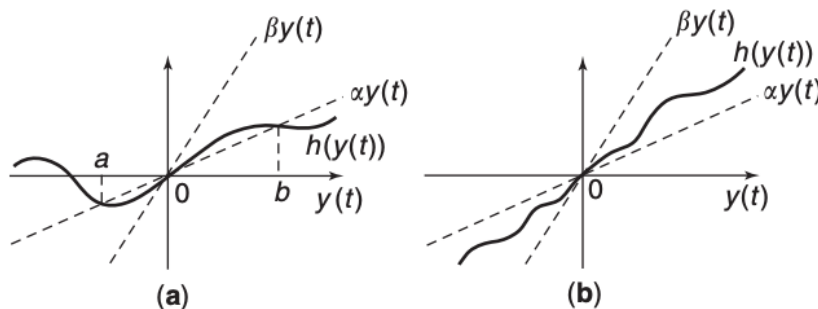


FIGURE 1.3 – Conditions locales et globales du secteur (G. CHEN, 2004)

1.3.3 Critère de Popov

Théorème 9 (Critère de Popov) Soit $G(s)$ la fonction de transfert définie par (1.22), et $\text{Re}\{\cdot\}$ désigne la partie réelle d'un nombre complexe (ou d'une fonction). Si le système de Lure (1.24) satisfait les conditions suivantes (KHALIL, 2002) :

- A est stable et $\{A, b\}$ est contrôlable.
- Le système satisfait la condition de secteur globale avec $\alpha = 0$.
- Pour tout $\epsilon > 0$, il existe une constante $\gamma > 0$ telle que

$$\text{Re} \{(1 + j\gamma\omega)G(j\omega)\} + \frac{1}{\beta} \geq \epsilon \quad \text{pour tout } \omega \geq 0. \quad (1.27)$$

Alors, le système est globalement asymptotiquement stable autour de son équilibre $x_e = 0$ dans le secteur.

1.3.4 Critère du Cercle

Le critère de Popov est naturellement lié au critère de Nyquist linéaire (KHALIL, 2002 ; MOHLER, 1991 ; NARENDRA, 2014 ; PARKS et HAHN, 1993). Une généralisation plus directe du critère de Nyquist aux systèmes non linéaires est la suivante.

Théorème 10 (Critère du Cercle) Si le système de Lure (1.24) satisfait les conditions suivantes (KHALIL, 2002) :

- A n'a pas de valeur propre purement imaginaire et possède k valeurs propres avec des parties réelles positives.
- Le système satisfait la condition de secteur globale.
- L'une des situations suivantes est vérifiée :
 - (a) $0 < \alpha < \beta \rightarrow$ le tracé de Nyquist de $G(j\omega)$ entoure le disque $D[-(1/\alpha), -(1/\beta)]$ dans le sens contraire des aiguilles d'une montre k fois sans y pénétrer.
 - (b) $0 = \alpha < \beta \rightarrow$ le tracé de Nyquist de $G(j\omega)$ reste dans le demi-plan ouvert $\text{Re}\{s\} > -(1/\beta)$.
 - (c) $\alpha < 0 < \beta \rightarrow$ le tracé de Nyquist de $G(j\omega)$ reste dans le disque ouvert $D[-(1/\beta), -(1/\alpha)]$.
 - (d) $\alpha < \beta < 0 \rightarrow$ le tracé de Nyquist de $G(j\omega)$ entoure le disque $D[(1/\alpha), (1/\beta)]$ dans le sens contraire des aiguilles d'une montre k fois sans y pénétrer.

Alors, le système est globalement asymptotiquement stable autour de son équilibre $x_e = 0$.

Dans le cas de $0 < \alpha < \beta$, le disque $D[-(1/\alpha), -(1/\beta)]$ est représenté dans la Figure 1.4.

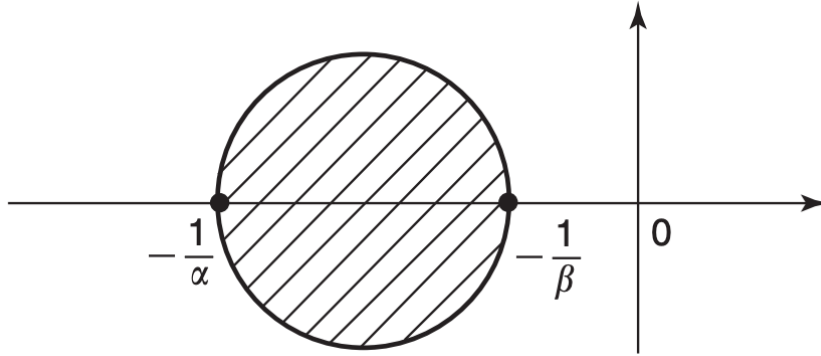


FIGURE 1.4 – Disque $D[-(1/\alpha), -(1/\beta)]$ (G. CHEN, 2004)

1.3.4.1 Exemple d'illustration 2

Prenons un exemple de systèmes linéaires avec une saturation. Nous considérons les deux systèmes :

$$G_1(s) = \frac{0,01}{s(1 + 10s)^2}$$

$$G_2(s) = \frac{0,02}{(1 + 0,01s)(1 + 2\xi\frac{s}{\omega_1} + (\frac{s}{\omega_1})^2)}$$

avec $\xi = 0,04$ et $\omega_1 = 200$. La boucle fermée correspond à la Figure 1.5 (où $G(s) = G_1(s)$ ou $G_2(s)$) avec une non-linéarité invariante dans le temps dans le secteur $[0; 1]$.

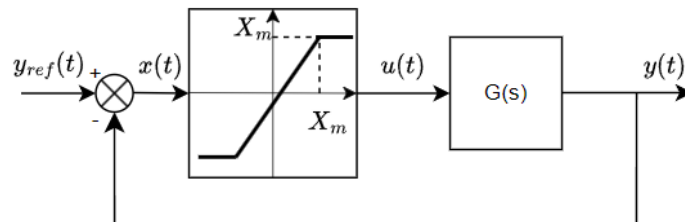


FIGURE 1.5 – Système linéaire avec saturation en entrée

Nous considérons l'analyse de la stabilité de ces systèmes en utilisant à la fois le critère du Cercle (Théorème 10) et le critère de Popov (Théorème 9).

En appliquant le cas b) du Théorème 10, la boucle fermée est asymptotiquement stable si le tracé de Nyquist de $G(s)$ se trouve à droite de la ligne verticale définie par $\text{Re}[s] = -1$. La Figure 1.6 montre le tracé de Nyquist de $G_1(s)$ et $G_2(s)$. Pour $G_1(s)$, le tracé de Nyquist est à droite de la ligne verticale -1 et la boucle fermée est asymptotiquement stable. Cependant, pour $G_2(s)$, le tracé de Nyquist croise la ligne -1 , nous ne pouvons donc pas conclure à la stabilité asymptotique.

Si nous considérons maintenant le critère de Popov, et que nous utilisons l'interprétation graphique du Théorème 9, la boucle fermée est asymptotiquement stable si le tracé de $\text{Re}[G(j\omega)]$ par rapport à $\omega\text{Im}[G(j\omega)]$ se trouve à droite de la ligne passant par -1 avec une pente de $\frac{1}{\gamma}$. Pour $\gamma = 0$, nous trouvons ici la même condition que celle du critère du Cercle. La Figure 1.7 montre une ligne passant par -1 pour une valeur de $\gamma = 100$ pour G_1 et $\gamma = 0,004$ pour G_2 . Comme nous pouvons le voir sur

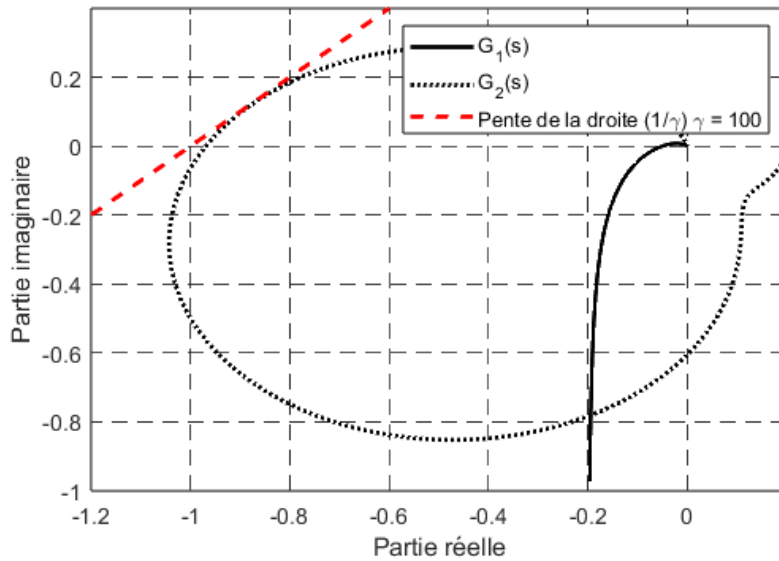


FIGURE 1.6 – Critère du Cercle

la figure, le tracé de Nyquist pour $G_1(s)$ et $G_2(s)$ se trouve à droite de cette ligne, nous pouvons donc conclure que pour les deux systèmes, la boucle fermée est asymptotiquement stable.

Les conditions du critère du Cercle pour la stabilité asymptotique sont plus conservatrices que celles du critère de Popov, mais il est également nécessaire de considérer que le critère du Cercle couvre une plus large classe de systèmes en permettant une non-linéarité variant dans le temps (ce qui n'est pas le cas dans l'exemple d'illustration).

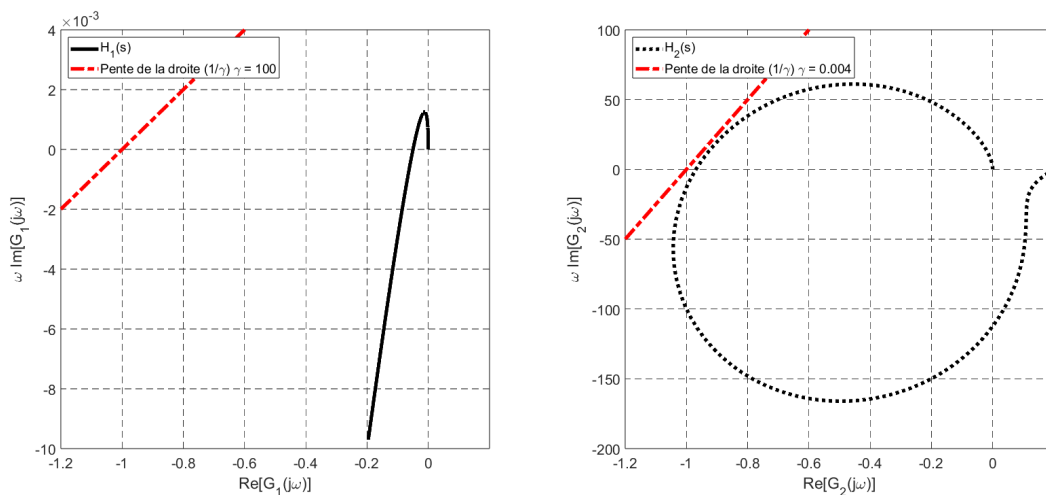


FIGURE 1.7 – Critère de Popov

1.4 Définitions de stabilité au sens de Lyapunov

Les notions de stabilité définies précédemment se concentrent sur des critères spécifiques ou des propriétés particulières du système. Pour être applicables, il est nécessaire que ces systèmes soient linéaires, sous forme de Lure, c'est-à-dire que la non-linéarité doit être extraite et à secteur borné. Cela rend leur approche assez restrictive.

De plus et de notre point de vue, ces méthodes sont également difficiles à mettre en œuvre, comme par exemple transformer un système en Lure. Cette complexité peut les rendre moins adaptées pour estimer une région de stabilité dans le contexte de systèmes industriels complexes, notion que nous définissons plus tard. C'est donc en raison de ces restrictions que nous nous intéressons à la théorie de Lyapunov, qui est à l'origine de ces critères et reste l'une des approches les plus fondamentales pour étudier la stabilité des systèmes dynamiques.

Dans notre cas d'utilisation et pour cette thèse, nous employons la théorie de Lyapunov. Celle-ci, introduite à la fin du dix-neuvième siècle (LYAPUNOV, 1892), est un moyen pratique d'étudier la stabilité d'un point d'équilibre pour un système dynamique sans hypothèse structurelle de ce dernier.

Dans cette section, nous abordons la stabilité, pour les systèmes autonomes à temps invariant définis par l'équation différentielle suivante :

$$\begin{cases} \dot{x} = f(x) \\ x(t_0) = x_0 \end{cases} \quad (1.28)$$

Note : La stabilité entrée - état est traitée dans la section 1.5.3.

1.4.1 Définition intuitive de la stabilité

Dans le cadre des systèmes dynamiques, l'idée de stabilité peut être interprétée au travers du concept d'énergie potentielle et d'énergie cinétique.

Prenons l'exemple d'une balle qui roule sur un paysage composé de collines et de vallées. Chaque position de la balle dans ce paysage a une certaine énergie potentielle en raison de sa hauteur par rapport au niveau de référence, et une énergie cinétique associée à sa vitesse.

Un point d'équilibre stable est comme un creux ou une vallée dans ce paysage. Supposons que la balle soit au repos dans un creux. Si nous donnons à la balle une petite poussée (c'est-à-dire si nous lui donnons un peu d'énergie cinétique), elle roulera un peu sur les pentes du creux, mais l'énergie potentielle qu'elle gagne en montant les pentes la ralentira et elle finira par s'arrêter et rouler en arrière vers le creux. En d'autres termes, toute petite perturbation est "absorbée" par le système et dissipée avec le temps, ce qui ramène le système à son point d'équilibre.

Un point d'équilibre instable est comme un sommet de colline. Si la balle est parfaitement au repos au sommet, elle pourrait y rester, mais la moindre perturbation (comme un coup de vent ou une petite poussée) lui donnerait de l'énergie cinétique et la ferait rouler vers le bas. Contrairement au cas stable, le système ne revient pas à son état initial après une perturbation, mais s'éloigne au contraire de plus en plus de son point d'équilibre initial.

Dans ces deux cas, l'énergie totale du système (énergie potentielle plus énergie cinétique) est conservée, si nous négligeons la dissipation d'énergie par frottement ou autres formes de dissipation. Cependant, la répartition de l'énergie entre l'énergie potentielle et l'énergie cinétique change avec le temps, et c'est cette redistribution de l'énergie qui sous-tend la dynamique du système et sa tendance à la stabilité ou à l'instabilité.

1.4.2 Stabilité au sens de Lyapunov

Définition 11 (Stabilité au sens de Lyapunov) *Le point d'équilibre x_e est dit stable au sens de Lyapunov si et seulement si $\forall \epsilon > 0$ il existe $\rho(\epsilon) > 0$ tel que (KHALIL, 2002) :*

$$\|x(t_0) - x_e\| \leq \rho(\epsilon) \Rightarrow \|x(t) - x_e\| \leq \epsilon, \quad \forall t \geq t_0. \quad (1.29)$$

Cela signifie que, quel que soit le rayon ϵ d'une boule centrée sur l'équilibre, il est possible de trouver une sous-boule de rayon $\rho(\epsilon)$, telle que, quelle que soit la condition initiale du système à l'intérieur de cette sous-boule de rayon ρ , la trajectoire qui en découlera ne quittera jamais la boule de rayon ϵ .

La [Figure 1.8](#) ci-dessous permet de mettre en image les définitions et illustre les stabilités au sens de Lyapunov que nous abordons dans cette section.

Ici, nous nous positionnons dans le cas *f* de la [Figure 1.8](#). Nous avons un point d'équilibre qui peut être envisagé comme un trait plat et horizontal. Si nous plaçons une balle sur ce trait, même si elle est légèrement poussée ou perturbée, en présence de frottements, elle se déplacera un peu mais s'arrêtera rapidement, restant toujours sur le trait. C'est ce que signifie la stabilité au sens de Lyapunov : lorsque le système est perturbé, il peut se déplacer temporairement, mais, grâce aux frottements ou à la forme d'une cuvette, il revient et reste à une position proche de son point d'équilibre, sans s'éloigner considérablement.

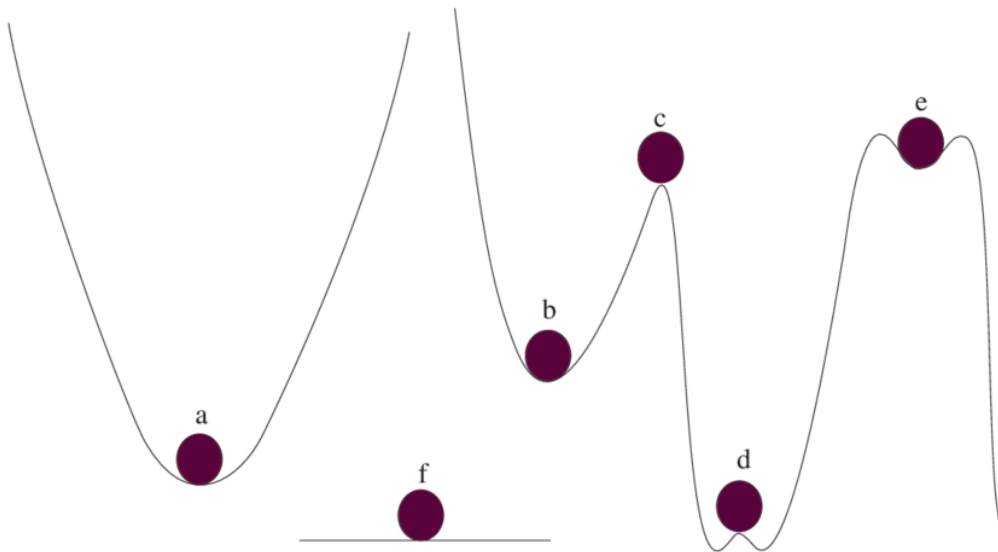


FIGURE 1.8 – Illustration de la stabilité au sens de Lyapunov

1.4.3 Stabilité asymptotique

Définition 12 (Stabilité asymptotique) *Le point d'équilibre x_e est dit stable asymptotiquement si deux conditions sont remplies (KHALIL, 2002) :*

- *Le point d'équilibre est stable selon la définition précédente.*
- *Il existe une fonction $\rho(\epsilon)$ telle que*

$$\|x(t_0) - x_e\| \leq \rho(\epsilon) \Rightarrow \lim_{t \rightarrow \infty} \|x(t) - x_e\| = 0. \quad (1.30)$$

Cela signifie que, non seulement toutes les trajectoires qui commencent dans une sous-boule de rayon ρ restent dans la boule de rayon ϵ , mais aussi qu'elles tendent vers le point d'équilibre x_e lorsque le temps tend vers l'infini.

La [Figure 1.8](#) permet de mettre en image cette définition et représente un système qui est asymptotiquement stable.

La balle, dans les cas a , b et e sur la figure, peut-être lancée sur le côté. Elle roulera de haut en bas, perdra de l'énergie à chaque oscillation, en supposant des frottements, et finalement s'installera au fond de la crevasse bol, qui est le point d'équilibre. Le système revient toujours à son point d'équilibre avec le temps.

1.4.4 Stabilité exponentielle

Définition 13 (Stabilité exponentielle) *Le point d'équilibre x_e est exponentiellement stable, si il est stable au sens de Lyapunov et si il existe $\alpha > 0$ tel que, quelque soit $\epsilon > 0$, il existe $\rho(\epsilon) > 0$ tel que (KHALIL, 2002) :*

$$\|x(t_0) - x_e\| < \rho(\epsilon) \Rightarrow \|x(t) - x_e\| < \epsilon \|x_e - x(t_0)\| e^{-\alpha(t-t_0)}, \quad \forall t \geq t_0. \quad (1.31)$$

Cette définition traduit le fait que toute trajectoire issue d'une condition initiale appartenant à la boule ouverte de rayon ρ converge vers le point d'équilibre x_e plus rapidement qu'une fonction exponentielle. α est appelé dans ce cas le taux de convergence. Nous notons que la stabilité exponentielle implique la stabilité asymptotique ainsi que la stabilité au sens de Lyapunov.

Pour visualiser la stabilité exponentielle, imaginons une vallée dont les parois sont abruptes. Dans les cas a et b de la [Figure 1.8](#), plus ces parois sont escarpées, plus la balle lancée contre celles-ci reviendra rapidement à son point de départ au fond de la vallée. Cette rapidité avec laquelle la balle retourne à sa position d'équilibre, poussée par l'inclinaison prononcée des parois, est analogue au taux de convergence α . Un α élevé signifierait des parois très abruptes, entraînant un retour rapide de la balle vers son point d'équilibre.

1.4.5 Stabilité globale

Dans chacune des définitions [Définition 11](#), [Définition 12](#) et [Définition 13](#), la stabilité est définie localement, les conditions initiales étant prises dans un voisinage $X(x_e)$ autour du point d'équilibre x_e . Si $X(x_e) = \mathbb{R}^n$, le point d'équilibre x_e est dit globalement asymptotiquement (exponentiellement) stable.

Définition 14 (Stabilité globale) *Si la condition de stabilité asymptotique (respectivement exponentielle) est vérifiée dans tout \mathbb{R}^n , le point d'équilibre est globalement asymptotiquement (respectivement exponentiellement) stable (KHALIL, 2002).*

En observant la [Figure 1.8](#) illustrant les balles, nous pouvons percevoir une distinction claire entre les stabilités locale et globale. Dans le cadre de la stabilité locale, visualisons un léger creux situé au sommet d'une colline, soit le cas e . Lorsque nous plaçons une balle à l'intérieur de cette cuvette, elle reste stable à l'intérieur de ce petit espace, mais si elle est poussée en dehors, elle pourrait rouler en bas de la colline. Dans le cas de la stabilité globale, imaginons une vaste et profonde vallée s'étendant sur tout le paysage, soit le cas a . Peu importe où nous plaçons la balle dans cette vallée, même si elle est déplacée ou perturbée, elle finira toujours par revenir et se stabiliser au fond de cette vallée, quelle que soit sa position initiale sur l'ensemble de la vallée.

Cependant, nous notons que ces définitions présentent des inconvénients dans la vérification des différentes propriétés définies précédemment pour un système :

- Il est difficile de calculer de manière explicite chaque solution correspondante à chacune des conditions initiales.
- Ces définitions sont exprimées sous la forme de propositions mathématiques qu'il n'est pas simple de vérifier (trouver ϵ , ρ , α , etc.).

C'est pourquoi il existe un cadre général d'étude fondé sur les méthodes de Lyapunov.

1.5 Méthode directe de Lyapunov

1.5.1 Introduction

La philosophie de base de la méthode directe de Lyapunov, ou appelée aussi seconde méthode de Lyapunov, est l'extension mathématique d'une observation physique fondamentale que nous avons expliquée lors de la section 1.4.1 : si l'énergie totale d'un système mécanique (ou électrique) est continuellement dissipée, alors le système, qu'il soit linéaire ou non linéaire, doit finalement converger vers un point d'équilibre. Ainsi, nous pouvons conclure sur la stabilité d'un système en examinant la variation d'une seule fonction scalaire.

Plus précisément, considérons le système non linéaire masse-amortisseur-ressort, dont l'équation dynamique est :

$$m\ddot{x} + b\dot{x}|\dot{x}| + k_0x + k_1x^3 = 0 \quad (1.32)$$

avec $b\dot{x}|\dot{x}|$ représentant une dissipation ou un amortissement non linéaire, et $(k_0x + k_1x^3)$ représentant un terme de ressort non linéaire. Supposons que la masse soit écartée de la longueur naturelle du ressort sur une grande distance, puis relâchée. Le mouvement qui en résulte est-il stable ? Il est très difficile de répondre à cette question en utilisant les définitions de la stabilité, car la solution générale de cette équation non linéaire est indisponible. Cependant, l'examen de l'énergie du système peut nous en dire beaucoup sur le modèle de mouvement.

L'énergie mécanique totale du système est la somme de son énergie cinétique et de son énergie potentielle :

$$V(x) = \frac{1}{2}m\dot{x}^2 + \int_0^x (k_0x + k_1x^3)dx = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}k_0x^2 + \frac{1}{2}k_1x^4. \quad (1.33)$$

En comparant les définitions de la stabilité et de l'énergie mécanique, nous pouvons facilement voir certaines relations entre l'énergie mécanique et les concepts de stabilité décrits précédemment :

- Une énergie minimale correspond au point d'équilibre.
- La stabilité asymptotique implique la convergence de l'énergie vers son minimum local.

Ces relations indiquent que la valeur d'une quantité scalaire, l'énergie mécanique, reflète indirectement l'amplitude du vecteur d'état ; de plus, que les propriétés de stabilité du système peuvent être caractérisées par la variation de l'énergie mécanique du système.

Le taux de variation de l'énergie pendant le mouvement du système est obtenu facilement en différenciant la première égalité (1.33) et en utilisant (1.32) :

$$\dot{V}(x) = m\dot{x}\ddot{x} + (k_0x + k_1x^3)\dot{x} = \dot{x}(-b\dot{x}|\dot{x}|) = -b|\dot{x}|^3. \quad (1.34)$$

L'équation (1.34) implique que l'énergie du système, partant d'une valeur initiale, est continuellement dissipée par l'amortisseur jusqu'à ce que la masse se stabilise, c'est-à-dire jusqu'à $x = 0$. Physiquement, il est facile de voir que la masse doit finalement se stabiliser à la longueur naturelle du ressort, car elle est soumise à une force de ressort non nulle à toute autre position.

La méthode directe de Lyapunov est basée sur une généralisation des concepts du système masse-ressort-amortisseur ci-dessus à des systèmes plus complexes. Face à un ensemble d'équations différentielles non linéaires, la procédure de base de la méthode directe de Lyapunov consiste à générer une fonction scalaire, image de l'énergie, pour le système dynamique et à examiner la variation temporelle

de cette fonction scalaire. De cette manière, des conclusions peuvent être tirées sur la stabilité de l'ensemble d'équations différentielles sans utiliser les définitions difficiles de la stabilité ou nécessiter une connaissance explicite des solutions.

La méthode directe de Lyapunov permet également de déterminer non seulement la stabilité d'un système, mais aussi d'identifier le domaine d'attraction associé. C'est l'un des avantages clés de cette approche par rapport à des critères plus simples et plus directs à mettre en œuvre comme dans la section 1.4.1. Lorsque nous utilisons les critères de stabilité, comme ceux présentés dans la [Définition 12](#) ou la [Définition 13](#) par exemple, nous pouvons déterminer si un système est stable ou non, mais ils ne nous donnent pas d'information sur la taille ou la forme du domaine d'attraction ([Domain Of Attraction \(DOA\)](#)). En revanche, la méthode directe de Lyapunov nous permet de trouver un domaine de stabilité précis dans l'espace des états du système.

En utilisant une fonction de Lyapunov appropriée, nous pouvons définir une région de l'espace des états où la stabilité du système est garantie. Cette région, appelée domaine ou région d'attraction, représente l'ensemble des conditions initiales pour lesquelles le système convergera vers un état d'équilibre appartenant à ce DOA. La forme et la taille de ce domaine dépendent de la fonction de Lyapunov choisie et des propriétés du système.

Cette capacité à déterminer un domaine de stabilité associé à un système est d'une importance fondamentale en ingénierie des systèmes de contrôle. Cela nous permet de garantir que l'état du système ne pénètre pas dans des zones où la stabilité n'est pas vérifiable. En connaissant ces régions, nous pouvons adapter notre stratégie de contrôle pour assurer un fonctionnement stable et éviter les comportements indésirables.

Dans la suite, nous considérons toujours le système autonome (1.9), défini précédemment.

1.5.2 Cas des systèmes à temps continu

Soit $V(x)$ une fonction continue différentiable définie dans un domaine $D \subset \mathbb{R}^n$; $0 \in D$. La dérivée de V le long des trajectoires de $\dot{x} = f(x)$ est ([KHALIL, 2002](#)) :

$$\begin{aligned}\dot{V}(x) &= \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i = \sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i(x) \\ &= \left(\frac{\partial V}{\partial x} \right)^T f(x).\end{aligned}$$

Nous pouvons noter que si 0 est un point d'équilibre, alors immédiatement $\dot{V}(0)$ est égale à 0.

Si $\phi(t, x_0)$ est la solution de $\dot{x} = f(x)$ qui commence à l'état initial x_0 au temps $t = 0$, alors :

$$\dot{V}(x) = \frac{d}{dt} V(\phi(t; x_0)).$$

Théorème 15 (Théorème de stabilité de Lyapunov) Soit $x = 0$ un point d'équilibre pour $\dot{x} = f(x)$ et un ensemble $D \subset \mathbb{R}^n$ contenant 0 dans son intérieur. Soit $V : D \rightarrow \mathbb{R}$ une fonction continue différentiable telle que (KHALIL, 2002) :

$$V(0) = 0 \text{ et } V(x) > 0 \quad \forall x \in D \setminus \{0\} \quad (1.35)$$

et

$$\dot{V}(x) \leq 0 \quad \forall x \in D \quad (1.36)$$

alors $x = 0$ est stable au sens de Lyapunov. De plus, si :

$$\dot{V}(x) < 0 \quad \forall x \in D \setminus \{0\} \quad (1.37)$$

alors $x = 0$ est asymptotiquement stable au sens de Lyapunov.

Suite à la présentation du théorème de stabilité de Lyapunov qui offre un cadre pour analyser la stabilité locale autour d'un point d'équilibre, nous introduisons maintenant une version plus générale qui permet d'examiner la stabilité globale dans l'ensemble de l'espace d'état.

Théorème 16 (Théorème de stabilité globale de Lyapunov) Soit $x = 0$ un point d'équilibre pour $\dot{x} = f(x)$. Soit $V : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue différentiable telle que (KHALIL, 2002) :

$$V(0) = 0 \text{ et } V(x) > 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}$$

$$\|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$$

$$\dot{V}(x) < 0 \quad \forall x \in \mathbb{R}^n \setminus \{0\}$$

alors $x = 0$ est globalement asymptotiquement stable au sens de Lyapunov.

1.5.2.1 Exemple d'illustration 3

Considérons le système suivant, l'équation du pendule sans frottement (KHALIL, 2002) :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a \sin(x_1) \end{cases}$$

avec $a > 0$. Étudions la stabilité du point d'équilibre à l'origine. Un candidat à la fonction de Lyapunov est la fonction d'énergie :

$$V(x) = a(1 - \cos(x_1)) + \frac{1}{2}x_2^2.$$

Clairement, $V(0) = 0$ et $V(x)$ est définie positive sur le domaine $-\pi < x_1 < \pi$ si $a > 0$. La dérivée de $V(x)$ le long des trajectoires du système est donnée par :

$$\dot{V}(x) = ax_1 \sin(x_1) + x_2 \dot{x}_2 = ax_2 \sin(x_1) - ax_2 \sin(x_1) = 0.$$

Ainsi, les conditions (1.35) et (1.36) sont satisfaites, et nous concluons que l'origine est stable.

Avec $\dot{V}(x) = 0$ et la fonction de Lyapunov actuellement choisie, cela ne nous permet pas d'affirmer que l'origine est asymptotiquement stable. Cependant, cette observation ne conteste pas la stabilité asymptotique de l'origine. Celle-ci pourrait être établie en choisissant une autre fonction de Lyapunov appropriée.

1.5.2.2 Exemple d'illustration 4

Considérons le système suivant (SLOTINE, LI et al., 1991) :

$$\begin{cases} \dot{x}_1 = x_2 - x_1(x_1^2 + x_2^2) \\ \dot{x}_2 = -x_1 - x_2(x_1^2 + x_2^2) \end{cases}$$

Nous considérons ici le point d'équilibre de ce système comme l'origine. Une fonction de Lyapunov candidate pourrait être :

$$V = x_1^2 + x_2^2.$$

La dérivée de V le long de toute trajectoire du système est :

$$\dot{V} = 2x_1\dot{x}_1 + 2x_2\dot{x}_2 = -2(x_1^2 + x_2^2)^2$$

qui est définie négative. Par conséquent, l'origine est un point d'équilibre globalement asymptotiquement stable. Notons que le caractère global de ce résultat de stabilité implique également que l'origine est le seul point d'équilibre du système.

Théorème 17 (Théorème de stabilité exponentielle de Lyapunov) *Si il existe une boule $B_\delta(0)$ et une fonction $V(x) \in C^1$ telles que, pour tout $x \in B_\delta(0)$ (KHALIL, 2002) :*

$$k_1\|x\|^a \leq V(x) \leq k_2\|x\|^a \tag{1.38}$$

$$\dot{V}(x(t)) \leq -k_3\|x\|^a \tag{1.39}$$

où k_1, k_2, k_3 et a sont des constantes positives. Alors l'origine est exponentiellement stable.

De plus, si (1.38) et (1.39) s'appliquent pour tout $x \in \mathbb{R}^n$, alors l'origine est globalement exponentiellement stable.

Note 1 : $\frac{k_2}{k_3}$ est une borne supérieure du taux de convergence exponentielle de $\|x(t)\|$.

Note 2 : Une fonction $f(x)$ appartient à la classe C^1 si elle est continue et si sa dérivée $f'(x)$ existe et est continue sur son domaine :

$$f \in C^1 \iff f \text{ est continue et } f' \text{ est continue.}$$

1.5.2.3 Exemple d'illustration 5

Considérons le système autonome donné par (PETTERSEN, 2017) :

$$\dot{x} = -x - x^3.$$

Voici une fonction de Lyapunov candidate :

$$V(x) = \frac{1}{2}x^2.$$

La dérivée temporelle de $V(x)$ le long des trajectoires du système est :

$$\dot{V}(x) = x\dot{x} = -x^2 - x^4.$$

Il est facilement démontrable que l'origine est globalement asymptotiquement stable. Nous voulons montrer qu'il existe des constantes $a > 0$, $k_1 > 0$, $k_2 > 0$ et $k_3 > 0$ telles que :

$$k_1\|x\|^2 \leq V(x) \leq k_2\|x\|^2$$

et

$$\dot{V}(x) \leq -k_3\|x\|^a.$$

Pour la première condition, nous choisissons $k_1 = k_2 = 1/2$ et $a = 2$. Alors il est vrai que :

$$\frac{1}{2}x^2 \leq V(x) \leq \frac{1}{2}x^2.$$

Pour la deuxième condition, nous devons montrer que $\dot{V}(x) \leq -k_3\|x\|^a$. Nous avons :

$$\begin{aligned} \dot{V}(x) &= -x^2 - x^4 \\ &= -\|x\|^2 - \underbrace{\|x\|^4}_{\leq 0} \\ &\leq -\|x\|^2. \end{aligned}$$

Nous choisissons $k_3 = 1$. Par conséquent, l'origine est globalement exponentiellement stable.

1.5.3 Stabilité entrée - état

Définition 18 (Stabilité entrée - état) Le système $\dot{x} = f(t, x, u)$ est stable entrée - état (*Input-to-State Stable (ISS)*) si il existe $\beta \in \mathcal{KL}$ et $\gamma \in \mathcal{K}$ tels que pour tout état initial $x(t_0)$ et pour toute entrée bornée $u(t)$ (KHALIL, 2002) :

$$\|x(t)\| \leq \max \left\{ \beta(\|x(t_0)\|, t - t_0), \gamma \left(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\| \right) \right\} \quad (1.40)$$

pour tout $t \geq t_0$.

La stabilité entrée - état nécessite que l'origine de $\dot{x} = f(t, x, 0)$ soit **globalement asymptotiquement stable**.

Théorème 19 (Théorème de stabilité ISS de Lyapunov) *Le système $\dot{x} = f(t, x, u)$ est ISS si il existe une fonction V de classe C^1 , des fonctions de classe \mathcal{K}^∞ α_1, α_2 , une fonction ρ de classe \mathcal{K} et une fonction continue positive α_3 telles que :*

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|) \quad (1.41)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x, u) \leq -\alpha_3(\|x\|), \quad \forall \|x\| \geq \rho(\|u\|) > 0. \quad (1.42)$$

Le théorème suivant aborde la stabilité ISS en retirant la considération du temps, se concentrant sur le système caractérisé par $\dot{x} = f(x, u)$.

Théorème 20 (Théorème de stabilité ISS de Lyapunov) *Le système $\dot{x} = f(x, u)$ est ISS si il existe une fonction V de classe C^1 , des fonctions de classe \mathcal{K}^∞ $\alpha_1, \alpha_2, \alpha_3$ et γ telles que (KHALIL, 2002) :*

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \quad (1.43)$$

$$\frac{\partial V}{\partial x} f(x, u) \leq -\alpha_3(\|x\|) + \gamma(\|u\|). \quad (1.44)$$

1.5.3.1 Exemple d'illustration 6

Considérons le système non-autonome suivant (KHALIL, 2002) :

$$\dot{x} = -x^3 + u.$$

L'origine du système non forcé est globalement asymptotiquement stable, ce qui peut être démontré en utilisant la fonction de Lyapunov candidate $V(x) = \frac{1}{2}x^2$. En utilisant la même fonction de Lyapunov, sa dérivée temporelle le long des trajectoires de l'ensemble du système est donnée par :

$$\dot{V}(x) = -x^4 + xu.$$

Sans aucun changement, introduisons un scalaire $\theta \in (0, 1)$. Ainsi, nous réécrivons la dérivée de \dot{V} comme suit :

$$\dot{V}(x) = -x^4 + xu + \theta x^4 - \theta x^4 = -(1 - \theta)x^4 + x(u - \theta x^3).$$

Nous obtenons $\dot{V}(x) \leq -(1 - \theta)x^4$ à condition que :

$$x(u - \theta x^3) < 0$$

ou, de manière équivalente :

$$|x| > \left(\frac{|u|}{\theta}\right)^3.$$

Ainsi, le système est ISS avec :

$$\rho(\|u\|) = \left(\frac{|u|}{\theta}\right)^3.$$

1.5.4 Cas des systèmes à temps discret

Les définitions précédentes ont des analogies pour les systèmes à temps discret. Considérons le système donné par (1.45) :

$$x_{k+1} = f(k, x_k, u_k) \quad (1.45)$$

avec l'état d'équilibre en 0, en remplaçant le temps continu $t \geq 0$ par le temps discret $k \in \{0, 1, 2, \dots\}$.

La dérivée temporelle $\dot{V}(t, x, u)$ de la fonction de Lyapunov le long des trajectoires est remplacée par :

$$\Delta V(k, x, u) = V(k+1, f(k, x, u)) - V(k, x, u) \quad (1.46)$$

dans les conditions définissant les fonctions de Lyapunov.

1.5.5 Domaine d'attraction

Le domaine d'attraction (également appelé bassin ou région de stabilité) d'un point d'équilibre d'un système est l'ensemble de tous les états initiaux qui génèrent des solutions du système qui, en cas de stabilité asymptotique, convergent vers ce point d'équilibre, ou qui, dans le cas de stabilité au sens de Lyapunov, restent dans un voisinage de ce point d'équilibre. Dans la majorité des cas (surtout dans l'industrie), il n'est pas suffisant de déterminer qu'un système donné a un point d'équilibre stable. Il est aussi crucial d'assurer une région de stabilité autour de ce point d'équilibre dans le but de maintenir le système dans ce voisinage.

En effet, le plus important est plutôt de déterminer la région d'attraction ou une approximation de cette région. De telles approximations peuvent être trouvées en utilisant la théorie de Lyapunov.

Mathématiquement, le DOA Rs s'écrit :

$$Rs = \{x(0) \in \mathbb{R}^n : \lim_{t \rightarrow \infty} x(t) = x_e\}$$

où $x(t)$ est l'état du système autonome au temps t et $x(0)$ est l'état initial.

Une grande partie de nos recherches se porte sur la définition de cette approximation du DOA.

La [Figure 1.8](#) illustre un système qui a plusieurs points d'équilibre, certains étant stables et d'autres instables. C'est une représentation d'un système dynamique non linéaire où différents points d'équilibre peuvent coexister. Par exemple, nous pourrions avoir un paysage avec plusieurs vallées (points d'équilibre stables) et collines (points d'équilibre instables). L'étude de ces systèmes nécessite une compréhension précise de la nature de chaque point d'équilibre pour prédire le comportement global du système. Cette figure montre à quel point il est important d'estimer le domaine attraction.

La détermination exacte du domaine d'attraction est généralement une tâche difficile, en particulier pour les systèmes non linéaires. Cependant, l'analyse des ensembles de niveau (level sets) de la fonction de Lyapunov peut fournir une estimation du domaine d'attraction plus ou moins conservative.

1.5.6 Level set

Une méthode analytique d'approximation du DOA est définie par la théorie de la stabilité de Lyapunov comme suit :

Théorème 21 (Estimation du DOA) *Un ensemble fermé $\mathcal{M} \subset \mathbb{R}^n$, incluant l'origine en tant qu'équilibre, peut approximer le DOA pour l'origine du système $\dot{x} = f(x)$ si les conditions suivantes sont satisfaites (CHESI, 2007a) :*

- \mathcal{M} est un ensemble invariant pour le système $\dot{x} = f(x)$.
- Une fonction définie positive $V(x)$ peut être trouvée telle que $\dot{V}(x)$ est définie négative à l'intérieur de \mathcal{M} .

Les conditions du [Théorème 21](#) garantissent que l'ensemble approximé \mathcal{M} est contenu dans le DOA.

L'ensemble de sous-niveaux, couramment désigné dans la littérature anglophone par "level set", est un concept clé dans notre étude. Pour assurer la clarté et la cohérence dans ce mémoire, nous utilisons le terme "level set" pour désigner cet ensemble. Plus formellement, le level set $\Omega(c)$ de la fonction de Lyapunov $V(x)$ est défini par la formule suivante :

$$\Omega(c) = \{x \in D : V(x) \leq c\}. \quad (1.47)$$

Selon le [Théorème 21](#), tout level set d'une fonction de Lyapunov candidate qui satisfait la stabilité asymptotique locale de l'équilibre peut être une estimation du DOA si la dérivée temporelle de la fonction de Lyapunov est négative partout à l'intérieur du level set.

Le problème d'approximation du DOA revient donc en un problème de recherche du plus grand level set d'une fonction de Lyapunov. Pour obtenir l'estimation la plus grande du DOA, il est nécessaire de trouver la valeur maximale $c \in \mathbb{R}$ pour laquelle $\Omega(c)$ satisfait les conditions du [Théorème 21](#).

Théorème 22 *Soit $V(x)$ une fonction de Lyapunov positive pour le système dynamique donné par $\dot{x} = f(x)$, et soit $\Omega(c)$ un ensemble invariant associé à un level set de la fonction $V(x)$. L'ensemble $\Omega(c_*)$ représente la plus grande estimation du DOA pour l'origine du système si c_* est donné par :*

$$\begin{cases} c_* = \max c \\ \text{tel que } \Omega(c) \subseteq \mathcal{H}(x) \\ \mathcal{H}(x) = \{0\} \cup \{x \in \mathbb{R}^n : \dot{V}(x) < 0\} \end{cases} \quad (1.48)$$

La détermination de la plus grande estimation du DOA via les level sets de la fonction de Lyapunov peut être abordée par plusieurs méthodes. Bien qu'il existe de nombreuses stratégies heuristiques, qui sont examinées dans le chapitre suivant consacré à la littérature existante, notre attention se porte sur une approche proposée dans ([NAJAFI et al., 2016](#)). L'[Algorithme 1](#), proposé dans cette publication, se distingue par son efficacité à déterminer la valeur optimale de c , soit c_* . Cette technique particulière est d'un intérêt central pour le développement de notre méthode. Dans cette algorithme, nous partons d'un ensemble de n_s points dans l'espace d'état pour évaluer la fonction de Lyapunov et sa dérivée.

Algorithme 1 : Pseudo-code pour la détermination du plus grand c

```
Input :  $V(x), \dot{V}(x), n_s$   
 $c = \infty$  ;  
for  $i = 1 : n_s$  do  
  | if  $\dot{V}(x_i) \geq 0$  et  $V(x_i) < c$  then  
  |   |  $c = V(x_i)$  ;  
  | end  
 $c_* = c$  ;  
end
```

1.5.7 Systèmes inter-connectés

Les systèmes inter-connectés sont des systèmes composés de plusieurs sous-systèmes qui interagissent entre eux. C'est une propriété importante dans l'analyse et la conception de systèmes de contrôle, car elle permet de diviser un système complexe en sous-systèmes plus simples qui peuvent être analysés séparément. De nombreux outils dans la littérature fournissent des propriétés de stabilité des sous-systèmes inter-connectés. Un des résultats le plus fondamental pour les systèmes inter-connectés est probablement le théorème du petit-gain pour la stabilité entrée - sortie, introduit à l'origine par Zames ([ZAMES, 1959](#); [ZAMES, 1966a, 1966b](#)), et ([SANDBERG, 1963](#)), qui étudie l'interconnexion de rétroaction de deux systèmes stables d'entrée à sortie.

Chaque sous-système peut être décrit par ses propres équations d'état et l'interconnexion entre les sous-systèmes est souvent modélisée par des équations algébriques supplémentaires. Le résultat suivant, présenté à l'origine dans ([JIANG et al., 1995](#)), exploite le formalisme ISS pour énoncer un théorème du petit-gain pour les systèmes non linéaires invariants dans le temps, de la forme :

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2, u_1) & (1.49a) \\ \dot{x}_2 = f_2(x_2, x_1, u_2) & (1.49b) \end{cases}$$

avec $f_1 : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{m_1} \rightarrow \mathbb{R}^{n_1}$ et $f_2 : \mathbb{R}^{n_2} \times \mathbb{R}^{n_1} \times \mathbb{R}^{m_2} \rightarrow \mathbb{R}^{n_2}$ qui sont localement Lipschitz et qui satisfont $f_1(0, 0, 0) = 0$ et $f_2(0, 0, 0) = 0$. La [Figure 1.9](#) représente l'interconnexion en rétroaction des deux systèmes (1.49a) et (1.49b).

Théorème 23 (Théorème du petit-gain) *Supposons que, pour chaque $i \in \{1, 2\}$, il existe une fonction candidate de Lyapunov $V_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^+$ et $\alpha_i, \chi_i, \gamma_i \in K_\infty$ telles que, pour tous $(x_1, x_2) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ et tous $(u_1, u_2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$, les implications suivantes sont vraies ([JIANG et al., 1995](#)) :*

$$V_1 \geq \max\{\chi_1(V_2); \gamma_1(|u_1|)\} \Rightarrow \frac{\partial V_1}{\partial x_1}(x_1) f_1(x_1, x_2, u_1) \leq -\alpha_1(V_1) \quad (1.50)$$

$$V_2 \geq \max\{\chi_2(V_1); \gamma_2(|u_2|)\} \Rightarrow \frac{\partial V_2}{\partial x_2}(x_2) f_2(x_2, x_1, u_2) \leq -\alpha_2(V_2). \quad (1.51)$$

Alors, sous la condition de petit-gain :

$$\chi_1 \circ \chi_2(s) < s, \quad \forall s > 0 \quad (1.52)$$

le système (1.49) est ISS.

La condition (1.50) est équivalente à exiger que le sous-système x_1 (1.49a) soit ISS par rapport à ses entrées x_2 et u_1 . De même, (1.51) impose que le sous-système x_2 (1.49b) soit ISS par rapport à x_1 et u_2 .

La condition de petit-gain (1.52) peut être énoncée de manière équivalente :

$$\chi_2 \circ \chi_1(s) < s, \quad s > 0.$$

Le théorème du petit-gain (DE FIGUEIREDO et CHEN, 1993 ; DESOER et VIDYASAGAR, 2009 ; KHALIL, 2002) est également un critère pratique pour vérifier la stabilité BIBO d'un système de contrôle en boucle fermée. Il s'applique à presque tous les types de systèmes (linéaires et non linéaires, en temps continu et en temps discret, avec retard, de dimensions quelconques), tant que la formulation mathématique est appropriée pour satisfaire les conditions du théorème.

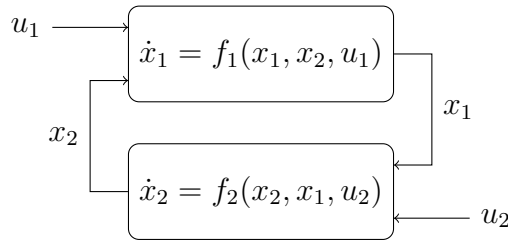


FIGURE 1.9 – Système inter-connecté

1.5.7.1 Exemple d'illustration 7

Considérons le système non linéaire suivant (JIANG et al., 1996) :

$$\begin{cases} \dot{z}_1 = -z_1 + z_1^3 z_2 \\ \dot{z}_2 = -z_2 - z_1^6 z_2 + z_3^2 \\ \dot{z}_3 = -z_3^3 + 0.5|z_3|^{3/2} \end{cases} \quad (1.53)$$

Soit $x_1 = (z_1, z_2)^T$ et $x_2 = z_3$. Une vérification directe montre que $V_1(x_1) = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2$ est une fonction de Lyapunov ISS pour le sous-système x_1 de (1.53) avec x_2 en entrée et :

$$\chi_1(s) = \frac{8}{2 - \epsilon_1} s^2$$

pour une valeur suffisamment petite de $\epsilon_1 > 0$. Aussi, $V_2(x_2) = \frac{1}{2}x_2^2$ est une fonction de Lyapunov ISS pour le sous-système x_2 de (1.53) avec x_1 en entrée et :

$$\chi_2(s) = \frac{2^{1/3}}{(4 - \epsilon_2)^{2/3}} s^{1/2}$$

pour une valeur suffisamment petite de $\epsilon_2 > 0$. Un calcul simple montre que pour des valeurs suffisamment petites de $\epsilon_1 > 0$ et $\epsilon_2 > 0$, alors $\chi_1 \circ \chi_2(s) < s$ pour tout $s > 0$. Par conséquent, selon le Théorème 23, le système (1.53) est globalement asymptotiquement stable au point d'équilibre $(z_1, z_2, z_3) = (0, 0, 0)$.

1.5.8 Instabilité

Le théorème de Chetaev permet d'étendre l'analyse de la stabilité des systèmes dynamiques non linéaires à des situations où les fonctions de Lyapunov ne fournissent pas de résultat concluant. En effet, bien que les fonctions de Lyapunov soient des outils puissants pour évaluer la stabilité des systèmes, leur absence ne signifie pas nécessairement que le système est instable.

En utilisant le théorème de Chetaev, nous pouvons examiner des cas où les fonctions de Lyapunov ne peuvent pas être trouvées. Ainsi, le théorème de Chetaev permet de fournir des conditions suffisantes pour l'instabilité, au sens de Lyapunov, d'un point d'équilibre.

Théorème 24 (Théorème de Chetaev) *Le point d'équilibre $x_e = 0$ est instable si il existe une fonction de classe C^1 $V : \mathbb{R}^+ \times \mathbb{R}^n \rightarrow \mathbb{R}$, une boule \mathcal{B}_r , un ensemble ouvert $\Omega \subseteq \mathcal{B}_r$, et une fonction γ de classe \mathcal{K} telles que (CHETAEV *et al.*, 1961) :*

- $V(t, x) > 0, \forall t \geq 0, \forall x \in \Omega.$
- $\sup_{t \geq 0} \sup_{x \in \Omega} V(t, x) < \infty.$
- $0 \in \partial\Omega$ (à la frontière de Ω).
- $V(t, x) = 0, \forall t \geq 0, \forall x \in \partial\Omega \cap \mathcal{B}_r.$
- $\dot{V}(t, x) > \gamma, \forall t \geq 0, \forall x \in \Omega.$

La Figure 1.8, dans les cas *c* et *d*, montre un point d'équilibre instable. Ici, nous avons un point d'équilibre précaire, comme une balle sur le sommet d'une colline. Même une petite brise peut déplacer la balle, et une fois qu'elle commence à bouger, elle roulera de plus en plus loin du sommet. Dans ce scénario, le système est susceptible de s'éloigner de plus en plus de son point d'équilibre suite à une perturbation, aussi petite soit-elle.

1.5.8.1 Exemple d'illustration 8

Considérons le système suivant (KHALIL, 2002) :

$$\begin{cases} \dot{x}_1 &= x_1(k^2 - x_1^2 - x_2^2) + x_2(x_1^2 + x_2^2 + k^2) \\ \dot{x}_2 &= -x_1(k^2 + x_1^2 + x_2^2) + x_2(k^2 - x_1^2 - x_2^2) \end{cases}$$

Considérons $V(x) = x_1^2 + x_2^2$ comme fonction de Lyapunov candidate. La dérivée de $V(x)$ le long des trajectoires du système est :

$$\begin{aligned} \dot{V}(x) &= 2x_1^2(k^2 - x_1^2 - x_2^2) + 2x_1x_2(x_1^2 + x_2^2 + k^2) \\ &\quad - 2x_1x_2(x_1^2 + x_2^2 + k^2) + 2x_2^2(k^2 - x_1^2 - x_2^2) \\ &= 2(x_1^2 + x_2^2)(k^2 - x_1^2 - x_2^2). \end{aligned}$$

En posant $k = 0$, nous obtenons $\dot{V}(x) = -2(x_1^2 + x_2^2)^2$. L'origine est donc globalement asymptotiquement stable.

En supposant $k \neq 0$, nous obtenons $\dot{V}(x) > 0$, pour $x_1^2 + x_2^2 < k^2$. Selon le théorème de Chetaev, l'origine est instable.

1.6 Récapitulatif et perspectives

Ce chapitre a exposé les bases fondamentales nécessaires à la compréhension des notions de stabilité au sens de Lyapunov qui nous sont utiles dans la suite. En commençant par les rappels théoriques, nous avons exploré les principaux outils mathématiques qui constituent le socle sur lequel reposent les études avancées de la stabilité. Par ailleurs, les modèles non linéaires et les systèmes à temps discret nous ont offert un éclairage sur la diversité et la complexité des systèmes étudiés.

Des méthodes classiques pour l'étude de la stabilité ont été mises en avant, notamment la passivité, les systèmes de Lure, le critère de Popov et le critère du Cercle. Ces méthodes, bien qu'essentielles, ont montré leurs limites dans certains contextes.

En approfondissant les notions de stabilité selon Lyapunov, nous avons découvert toute la richesse et la complexité de ce domaine. Au-delà du concept majeur de la stabilité asymptotique, d'autres formes et définitions de stabilité ont été abordées, reflétant la diversité des situations et systèmes que nous pouvons rencontrer. La méthode de Lyapunov se révèle être un outil essentiel, mais sa mise en œuvre nécessite une compréhension approfondie de ces différentes formes de stabilité. Enfin, la méthode directe de Lyapunov et ses applications ont mis en lumière la polyvalence et la puissance de cette technique, tout en révélant les défis associés à sa mise en œuvre.

Cette compréhension solide des notions de stabilité établies nous ouvre la porte aux discussions académiques actuelles sur les fonctions de Lyapunov. Le prochain chapitre est dédié à une analyse détaillée de la littérature existante, mettant en lumière diverses méthodes pour déterminer une fonction de Lyapunov.

Littérature existante

Littérature existante

Sommaire

2.1	Introduction	39
2.2	Méthodes théoriques	40
2.2.1	Méthode de linéarisation	40
2.2.1.1	Exemple d'illustration 9	41
2.2.2	Méthode de Krasovski	42
2.2.2.1	Exemple d'illustration 10	42
2.2.3	Méthode du principe d'invariance de LaSalle	43
2.3	Méthodes numériques	45
2.3.1	Méthode du gradient variable	46
2.3.1.1	Exemple d'illustration 11	47
2.3.2	Méthode de Zubov	47
2.3.2.1	Première approche	48
2.3.2.2	Deuxième approche	48
2.3.2.3	Troisième approche	50
2.3.2.4	Exemple d'illustration 12	50
2.3.3	Méthode de la somme des carrés	51
2.3.3.1	Exemple d'illustration 13	53
2.3.4	Méthode continue et affine par morceaux	54
2.3.5	Méthode de la fonction de base radiale	56
2.3.6	Synthèse des méthodes numériques	59
2.4	Méthodes automatiques	60
2.4.1	Réseaux de neurones	61
2.4.1.1	Principes fondamentaux des réseaux de neurones	62
2.4.1.2	État de l'art sur les fonctions de Lyapunov neuronales	64
2.4.2	Entraînement par métaheuristiques	65
2.4.2.1	Algorithmes génétiques	66
2.4.2.2	Optimisation par essaim particulaire	68
2.4.2.3	État de l'art de l'optimisation par métaheuristiques	70
2.4.3	Domaine de stabilité garanti	71
2.4.4	Fonction de Lyapunov fondée sur les réseaux de neurones	72
2.5	Synthèse des méthodes	74

2.1 Introduction

L'étude de la stabilité des systèmes dynamiques s'ancre dans une histoire riche d'innovations et de découvertes. Ce chapitre propose un panorama des méthodes existantes afin de déterminer une fonction de Lyapunov, en introduisant des perspectives novatrices, incarnées par les méthodes automatiques dans ce manuscrit. Alors que les systèmes dynamiques deviennent de plus en plus complexes, surtout dans le contexte industriel, les méthodes traditionnelles, bien qu'essentielles, montrent leurs limites, nécessitant ainsi une évolution des approches utilisées. Cet état de l'art révèle une évolution de la recherche depuis les premières méthodes théoriques jusqu'aux approches numériques et automatiques.

Nous débutons notre exploration par une étude de la littérature existante sur les méthodes théoriques, puis sur les méthodes numériques. Ces approches, ayant fait leurs preuves au fil du temps, permettent d'offrir une compréhension complète de l'analyse de la stabilité jusqu'à aujourd'hui. Nous mettons en évidence leurs forces, leurs contributions essentielles, mais aussi leurs limites face à des systèmes non linéaires de dimension élevée.

Après avoir établi ce solide contexte, nous plongeons dans le vif du sujet en explorant les méthodes automatiques. Inspirées des récentes avancées en intelligence artificielle ([Artificial Intelligence \(AI\)](#)), ces méthodes représentent une rupture potentielle avec les approches conventionnelles. Nous examinons en détail des innovations telles que la "Lyapunov Machine" et d'autres techniques basées sur les réseaux de neurones et l'entraînement par métaheuristiques. L'objectif est de déterminer la mesure dans laquelle ces nouvelles méthodes peuvent combler les lacunes des approches précédentes et offrir des outils adaptés aux défis actuels de la stabilité des systèmes dynamiques. En lien avec les méthodes automatiques explorées dans ce chapitre, une section est dédiée à une approche prometteuse et novatrice : la recherche d'une fonction de Lyapunov, modélisée par un réseau neuronal, par optimisation. Cette partie s'appuie sur un papier clé qui a guidé l'approche adoptée tout au long des travaux menés durant cette thèse.

Au travers de cette exploration, nous visons non seulement à fournir une compréhension exhaustive de l'analyse de la stabilité à travers ses développements significatifs, mais également à entrevoir les perspectives futures de ce domaine. Ce chapitre sert donc à la fois de bilan, de mini-guide et d'invitation à l'innovation dans l'étude de la stabilité des systèmes dynamiques.

Au cours des sections suivantes, une étude détaillée des méthodes théoriques et numériques est présentée. Cependant, il serait bénéfique pour les lecteurs de se référer également à la revue citée dans ([P. GIESL et HAFSTEIN, 2015b](#)). Cette revue, qui offre une analyse exhaustive sur la détermination numérique des fonctions de Lyapunov, constitue une ressource précieuse pour appréhender les avancées et les défis actuels dans ce domaine. Bien que cette revue aborde les méthodes automatiques, elle le fait de manière succincte. Ce chapitre aspire donc à combler ce manque en étudiant en profondeur l'état de l'art des méthodes de stabilité automatique.

2.2 Méthodes théoriques

La stabilité des systèmes non linéaires est un sujet fondamental de recherche en automatique et en théorie du contrôle. La détermination des fonctions de Lyapunov s'avère être une approche particulièrement puissante pour analyser la stabilité de tels systèmes. Dans cette optique, les méthodes théoriques jouent un rôle déterminant, offrant une base solide pour l'étude de la stabilité des systèmes non linéaires.

Parmi ces méthodes théoriques, la méthode de linéarisation, qui repose sur l'idée de simplifier un système non linéaire en une forme linéarisée autour d'un point d'équilibre, offre une première approche intuitive et accessible. Cependant, elle n'est valide qu'à proximité de l'équilibre et peut donc être limitée pour des systèmes présentant des comportements complexes ou chaotiques.

La méthode de Krasovski et le principe d'invariance de LaSalle apportent une contribution significative à l'analyse de la stabilité en offrant des outils plus sophistiqués et flexibles. La méthode de Krasovski fait appel à des formes fonctionnelles prédéfinies pour caractériser la stabilité, tandis que le principe d'invariance de LaSalle s'appuie sur l'idée d'ensembles invariants pour comprendre le comportement des solutions du système.

Ces méthodes théoriques sont essentielles pour poser les bases de la stabilité et fournir des premières approches pour la détermination des fonctions de Lyapunov. Elles sont complétées par des méthodes numériques et des méthodes automatiques qui permettent d'aborder des systèmes plus complexes ou d'obtenir des résultats plus précis et rapides.

Dans cette sous-section, nous explorons en détail certaines méthodes théoriques, leurs applications et les perspectives qu'elles offrent pour l'analyse de la stabilité des systèmes non linéaires.

2.2.1 Méthode de linéarisation

La méthode de linéarisation de Lyapunov concerne la stabilité locale d'un système non linéaire (HIRSCH et al., 2012). Considérons le système autonome (1.9), et supposons que $f(x)$ est continûment différentiable. Alors, la dynamique du système peut être écrite comme suit :

$$\dot{x} = \left. \frac{\partial f}{\partial x} \right|_{x=0} x + f_{t.o.s.}(x) \quad (2.1)$$

où *t.o.s.* signifie : termes d'ordre supérieur. Utilisons la matrice constante A pour désigner la matrice jacobienne de f par rapport à x en $x = 0$:

$$A = \left. \frac{\partial f}{\partial x} \right|_{x=0} \quad (2.2)$$

Il est important de noter que cette approche signifie que nous nous limitons uniquement aux systèmes invariants dans le temps. Alors, le système :

$$\dot{x} = Ax \quad (2.3)$$

est appelé linéarisation (ou approximation linéaire) du système original au point d'équilibre 0.

La stabilité d'un système linéaire décrit par une équation du type $\dot{x} = Ax$, où A est une matrice carrée, est déterminée par les valeurs propres de la matrice A :

- L'origine est stable au sens de Lyapunov si toutes les valeurs propres de A sont à partie réelle négative ou nulle, soit :

$$\forall \lambda \in \text{sp } A, \text{Re}(\lambda) \leq 0$$

et si les valeurs propres de A dont la partie réelle est nulle ont une multiplicité algébrique égale à la multiplicité géométrique (ANTSAKLIS et MICHEL, 1997).

- L'origine est asymptotiquement / exponentiellement stable si et seulement si la matrice A est Hurwitz, soit :

$$\forall \lambda \in \text{sp } A, \text{Re}(\lambda) < 0.$$

- Sinon l'origine est instable.

2.2.1.1 Exemple d'illustration 9

L'équation du pendule (KHALIL, 2002) :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a \sin x_1 - bx_2 \end{cases}$$

a deux points d'équilibre : $(x_1 = 0, x_2 = 0)$ et $(x_1 = \pi, x_2 = 0)$. Analysons la stabilité de ces deux points en utilisant la linéarisation. La matrice jacobienne est donnée par :

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -a \cos x_1 & -b \end{bmatrix}$$

Pour déterminer la stabilité de l'origine, nous évaluons la matrice jacobienne en $x = 0$:

$$A = \left. \frac{\partial f}{\partial x} \right|_{x=0} = \begin{bmatrix} 0 & 1 \\ -a & -b \end{bmatrix}$$

Les valeurs propres de A sont déterminées par le polynôme caractéristique $\lambda^2 + b\lambda + a$. En utilisant le critère de Routh, nous constatons que, pour toutes les valeurs positives de a et b , les parties réelles des valeurs propres sont négatives. Par conséquent, le point d'équilibre à l'origine est asymptotiquement stable.

Pour déterminer la stabilité du point d'équilibre en $(x_1 = \pi, x_2 = 0)$, nous évaluons la matrice jacobienne en ce point. Cela revient à effectuer un changement de variables $z_1 = x_1 - \pi, z_2 = x_2$ pour déplacer le point d'équilibre à l'origine, et évaluer la matrice jacobienne $[\partial f / \partial z]$ en $z = 0$:

$$\tilde{A} = \left. \frac{\partial f}{\partial x} \right|_{x_1=\pi, x_2=0} = \begin{bmatrix} 0 & 1 \\ a & -b \end{bmatrix}$$

Les valeurs propres de \tilde{A} sont :

$$\lambda_{1,2} = -\frac{1}{2}b \pm \frac{1}{2}\sqrt{b^2 + 4a}.$$

Pour toutes les valeurs de $a > 0$ et $b \geq 0$, il existe une valeur propre dans le demi-plan droit ouvert. Par conséquent, le point d'équilibre en $(x_1 = \pi, x_2 = 0)$ est instable.

2.2.2 Méthode de Krasovski

La méthode de Krasovski suggère une forme la plus simple de candidat à la fonction de Lyapunov pour les systèmes non linéaires autonomes (1.9), à savoir, $V = f^T f$. L'idée de base de la méthode consiste simplement à vérifier si ce choix particulier conduit effectivement à une fonction de Lyapunov.

Théorème 25 (Krasovski) *Considérons le système autonome défini par (1.9), avec le point d'équilibre d'intérêt étant l'origine. Soit $J(x)$ la matrice jacobienne du système (PERKO, 2013) :*

$$J(x) = \frac{\partial f}{\partial x}.$$

Si la matrice $K = J^T + J$ est définie négative dans un voisinage Ω de l'origine, alors le point d'équilibre à l'origine est asymptotiquement stable. Une fonction de Lyapunov pour ce système est :

$$V(x) = f^T(x)f(x).$$

Si Ω est l'espace d'état entier et, si de plus, $V(x) \rightarrow \infty$ lorsque $\|x\| \rightarrow \infty$, alors le point d'équilibre est globalement asymptotiquement stable.

2.2.2.1 Exemple d'illustration 10

Considérons le système non linéaire suivant (SLOTINE, LI et al., 1991) :

$$\begin{cases} \dot{x}_1 &= -6x_1 + 2x_2 \\ \dot{x}_2 &= 2x_1 - 6x_2 - 2x_2^3 \end{cases}$$

Nous avons :

$$J = \frac{\partial f}{\partial x} = \begin{bmatrix} -6 & 2 \\ 2 & -6 - 6x_2^2 \end{bmatrix}$$

$$K = J^T + J = \begin{bmatrix} -12 & 4 \\ 4 & -12 - 12x_2^2 \end{bmatrix}$$

La matrice K est facilement démontrée comme étant définie négative. Par conséquent, l'origine est asymptotiquement stable. Selon le théorème, un candidat à la fonction de Lyapunov est :

$$V(x) = (-6x_1 + 2x_2)^2 + (2x_1 - 6x_2 - 2x_2^3)^2.$$

Comme $V(x) \rightarrow \infty$ lorsque $\|x\| \rightarrow \infty$, l'état d'équilibre à l'origine est globalement asymptotiquement stable.

Bien que l'utilisation de ce résultat soit assez simple, son applicabilité est limitée en pratique car $K(x)$, pour de nombreux systèmes, ne satisfait pas la propriété de définition négative.

Théorème 26 (Théorème généralisé de Krasovski) *Considérons le système autonome défini par (1.9), avec le point d'équilibre d'intérêt étant l'origine, et $J(x)$ la matrice jacobienne du système. Si il existe deux matrices symétriques définies positives P et Q , telles que pour tout $\forall x \neq 0$, la matrice (PERKO, 2013) :*

$$K(x) = J^T P + P J + Q$$

est semi-définie négative dans un certain voisinage Ω de l'origine, alors l'origine est asymptotiquement stable.

La fonction $V(x) = f(x)^T f(x)$ est alors une fonction de Lyapunov pour ce système. Si la région Ω est l'ensemble de l'espace d'état, et si de plus, $V(x) \rightarrow \infty$ lorsque $\|x\| \rightarrow \infty$, alors le système est globalement asymptotiquement stable.

2.2.3 Méthode du principe d'invariance de LaSalle

Le théorème de LaSalle (LA SALLE, 1976; LASALLE, 1960) permet de conclure à la stabilité asymptotique d'un point d'équilibre même lorsque nous ne pouvons pas trouver une fonction $V(x)$ telle que $\dot{V}(x)$ soit localement définie négative. Cependant, il ne s'applique uniquement qu'aux systèmes invariants dans le temps ou périodiques.

Pour illustrer la méthode, prenons un exemple. Considérons l'équation du pendule avec frottements (KHALIL, 2002) :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\left(\frac{g}{l}\right) \sin x_1 - \left(\frac{k}{m}\right) x_2 \end{cases}$$

Prenons une fonction de Lyapunov candidate telle que :

$$V(x) = \left(\frac{g}{l}\right) (1 - \cos x_1) + \frac{1}{2} x_2^2.$$

La dérivée de $V(x)$ le long des trajectoires du système est donnée par :

$$\dot{V}(x) = \left(\frac{g}{l}\right) \dot{x}_1 \sin(x_1) + x_2 \dot{x}_2 = -\left(\frac{k}{m}\right) x_2^2.$$

La fonction de Lyapunov candidate basée sur l'énergie ne satisfait pas la condition de stabilité asymptotique du théorème de Lyapunov car $\dot{V}(x) = -\left(\frac{k}{m}\right) x_2^2$ est seulement semi-définie négative. Elle n'est pas définie négative car $\dot{V}(x) = 0$ pour $x_2 = 0$ quelle que soit la valeur de x_1 , c'est-à-dire le long de l'axe x_1 . Par conséquent, nous pouvons uniquement conclure que l'origine est stable. Le théorème de LaSalle nous permet d'arriver à une conclusion différente.

Notons que $\dot{V}(x)$ est négative partout, sauf sur la ligne $x_2 = 0$, où $\dot{V}(x) = 0$. Pour que le système maintienne la condition $\dot{V}(x) = 0$, la trajectoire du système doit être confinée à la ligne $x_2 = 0$. Sauf si $x_1 = 0$, en raison de l'équation du pendule :

$$x_2(t) \equiv 0 \Rightarrow \dot{x}_2(t) \equiv 0 \Rightarrow \sin x_1(t) \equiv 0.$$

Sur le segment $-\pi < x_1 < \pi$ de la ligne $x_2 = 0$, le système ne peut maintenir la condition $\dot{V}(x) = 0$ qu'à l'origine $x = 0$. Par conséquent, $V(x(t))$ doit diminuer vers 0, et $x(t) \rightarrow 0$ lorsque $t \rightarrow \infty$. Cela est cohérent avec la compréhension physique selon laquelle, en raison de la friction, l'énergie ne peut pas rester constante pendant que le système est en mouvement.

Si dans un domaine autour de l'origine, nous pouvons trouver une fonction de Lyapunov dont la dérivée le long des trajectoires du système est semi-définie négative, et si nous pouvons établir qu'aucune trajectoire ne peut rester identique aux points où $\dot{V}(x) = 0$ sauf à l'origine, alors l'argument ci-dessus montre que l'origine est asymptotiquement stable. Cette idée découle du principe d'invariance de LaSalle.

Théorème 27 (Théorème de LaSalle) Soit $\Omega \subset D$ un ensemble compact qui est positivement invariant par rapport au système autonome (1.9). Soit $V : D \rightarrow \mathbb{R}$ une fonction continue différentiable telle que $\dot{V}(x) \leq 0$ dans Ω . Soit E l'ensemble de tous les points de Ω où $\dot{V}(x) = 0$. Soit M le plus grand ensemble invariant dans E . Alors, chaque solution commençant dans Ω tend vers M lorsque $t \rightarrow \infty$ (STROGATZ, 2018).

Contrairement au théorème de Lyapunov, le théorème de LaSalle ne requiert pas que la fonction $V(x)$ soit définie positive. Notons également que la construction de l'ensemble Ω n'a pas besoin d'être liée à la construction de la fonction $V(x)$. Cependant, dans de nombreuses applications, la construction de $V(x)$ garantit elle-même l'existence d'un ensemble Ω . En particulier, si $\Omega_c = \{x \in \mathbb{R}^n \mid V(x) \leq c\}$ est borné et $\dot{V}(x) \leq 0$ dans Ω_c , alors nous pouvons prendre $\Omega = \Omega_c$.

Pour montrer que $x(t) \rightarrow 0$ lorsque $t \rightarrow \infty$, nous devons établir que le plus grand ensemble invariant dans E est l'origine. En effet, aucune solution ne peut rester identiquement dans E , autre que la solution triviale $x(t) = 0$. En spécialisant le théorème de LaSalle à ce cas et en prenant $V(x)$ comme étant définie positive, nous obtenons les deux corollaires suivants.

Corollaire 28 Soit $x = 0$ un point d'équilibre pour le système autonome (1.9). Soit $V : D \rightarrow \mathbb{R}$ une fonction continue différentiable et définie positive sur un domaine D contenant l'origine $x = 0$, telle que $\dot{V}(x) \leq 0$ dans D . Soit $S = \{x \in D \mid \dot{V}(x) = 0\}$ et supposons qu'aucune solution ne peut rester identiquement dans S , autre que la solution triviale. Alors, l'origine est asymptotiquement stable (STROGATZ, 2018).

Corollaire 29 Soit $x = 0$ un point d'équilibre pour (1.9). Soit $V : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction continue différentiable, radialement non bornée et définie positive telle que $\dot{V}(x) \leq 0$ pour tout $x \in \mathbb{R}^n$. Soit $S = \{x \in \mathbb{R}^n \mid \dot{V}(x) = 0\}$ et supposons qu'aucune solution ne peut rester identiquement dans S , autre que la solution triviale. Alors, l'origine est globalement asymptotiquement stable (STROGATZ, 2018).

Lorsque $\dot{V}(x)$ est définie négative, soit $S = \{0\}$, alors les corollaires précédents coïncident avec les théorèmes de Lyapunov.

2.3 Méthodes numériques

Tout en reconnaissant l'importance et la rigueur des méthodes théoriques dans l'analyse de la stabilité des systèmes dynamiques, il convient de noter que ces techniques peuvent présenter des limites dans leur application aux systèmes non linéaires. Souvent, la complexité de ces systèmes nécessite une approche plus flexible et adaptable, capable de gérer des dynamiques plus complexes. Les systèmes non linéaires, par leur nature même, ont souvent des dynamiques complexes qui ne peuvent pas être facilement simplifiées ou linéarisées.

C'est là qu'interviennent les méthodes numériques. Ces dernières, en tirant parti des avancées technologiques et de la puissance de calcul, offrent des techniques efficaces et pragmatiques pour l'analyse de la stabilité des systèmes non linéaires. En utilisant des méthodes algorithmiques pour approximer les fonctions de Lyapunov, ces techniques sont en mesure de surmonter certaines des contraintes associées aux méthodes théoriques, offrant ainsi un cadre plus flexible pour l'analyse de la stabilité. Nous étudions en détail plusieurs de ces méthodes numériques et leurs applications.

Les méthodes numériques permettent de déterminer des fonctions de Lyapunov pour une gamme plus large de systèmes non linéaires. Ces fonctions, qui sont essentielles pour prouver la stabilité d'un système, peuvent être très difficiles à obtenir par des moyens analytiques pour certains systèmes non linéaires. Les techniques numériques, en revanche, peuvent offrir une approche plus flexible et adaptable pour trouver ces fonctions.

Parmi les techniques numériques disponibles, la méthode du gradient variable, la méthode de Zubov, la méthode de la somme des carrés, la méthode continue et affine par morceaux, et la méthode de fonction de base radiale ont toutes prouvé leur efficacité. Chacune de ces méthodes a ses propres forces et limites.

La méthode du gradient variable est une technique générale qui utilise les informations locales sur la trajectoire du système pour évaluer la stabilité. Elle est largement utilisée grâce à sa flexibilité et sa capacité à s'adapter à une variété de systèmes.

La méthode de Zubov, quant à elle, offre une approche mathématiquement rigoureuse pour déterminer la fonction de Lyapunov et le domaine d'attraction d'un point fixe. Elle repose sur la résolution d'une équation différentielle partielle du premier ordre, appelée équation de Zubov.

La méthode de la somme des carrés est une technique puissante qui a été appliquée avec succès à une grande variété de problèmes d'optimisation. En stabilisation des systèmes, elle permet de trouver des fonctions de Lyapunov de manière algorithmique, rendant les analyses de stabilité plus accessibles.

La méthode continue et affine par morceaux est une approche intéressante qui vise à approximer la fonction de Lyapunov par une fonction affine par morceaux. Elle offre une flexibilité considérable et peut être particulièrement efficace pour des systèmes qui ont un grand nombre de variables d'état.

Enfin, la méthode de la fonction de base radiale est une autre technique d'approximation qui utilise des fonctions de forme particulière, appelées fonctions de base radiale, pour approximer la fonction de Lyapunov.

Dans la suite de cette section, nous détaillons chacune de ces méthodes et nous illustrons leurs applications et leurs performances dans l'analyse de la stabilité des systèmes dynamiques.

2.3.1 Méthode du gradient variable

La méthode du gradient variable est une approche formelle pour construire des fonctions de Lyapunov (MOHAMMEDALI et al., 2015 ; SCHULTZ et GIBSON, 1962).

Pour commencer, notons qu'une fonction scalaire $V(x)$ est liée à son gradient ∇V par la relation d'intégrale :

$$V(x) = \int_0^x \nabla V dx \quad \text{avec } V(0) = 0$$

où $\nabla V = \left\{ \frac{\partial V}{\partial x_1}, \dots, \frac{\partial V}{\partial x_n} \right\}^T$.

Pour retrouver une fonction scalaire unique V à partir du gradient ∇V , la fonction de gradient doit satisfaire ces conditions, que nous pouvons appeler équation de rotation ("Curl Condition" en anglais) :

$$\frac{\partial \nabla V_i}{\partial x_j} = \frac{\partial \nabla V_j}{\partial x_i} \quad \text{pour } i, j = 1, 2, \dots, n. \quad (2.4)$$

Le principe de la méthode du gradient variable est de supposer une forme spécifique pour le gradient ∇V , au lieu d'une forme spécifique pour une fonction de Lyapunov V elle-même. Une façon simple est de supposer que la fonction de gradient est de la forme :

$$\nabla V_i = \sum_{j=1}^n a_{ij} x_j \quad (2.5)$$

où a_{ij} sont des coefficients à déterminer.

Cela conduit à la procédure suivante pour rechercher une fonction de Lyapunov V :

- Supposer que ∇V est donné par (2.5) (ou une autre forme).
- Résoudre les coefficients a_{ij} pour satisfaire les équations de rotation (2.4).
- Restreindre les coefficients dans (2.5) de sorte que \dot{V} soit semi-définie négative (au moins localement).
- Calculer V à partir de ∇V par intégration.
- Vérifier si V est définie positive.

Il est généralement pratique d'obtenir V en intégrant le long d'une trajectoire qui est parallèle à chaque axe à son tour, c'est-à-dire :

$$V(x) = \int_0^{x_1} \nabla V_1(u, 0, \dots, 0) du + \int_0^{x_2} \nabla V_2(x_1, u, \dots, 0) du + \dots + \int_0^{x_n} \nabla V_n(x_1, x_2, \dots, x_{n-1}, u) du.$$

2.3.1.1 Exemple d'illustration 11

Dans cet exemple, nous utilisons la méthode du gradient variable pour trouver une fonction de Lyapunov pour le système suivant (W.-C. LIU et ZHANG, 2017) :

$$\begin{cases} \dot{x}_1 = \frac{-6x_1}{(1+x_1^2)^3} + 2x_2 \\ \dot{x}_2 = -2x_1 + \frac{2x_2}{1+x_1^2} \end{cases}$$

Nous pouvons déterminer que $x = 0$ est l'unique état d'équilibre. Sa fonction de gradient $\nabla V(x)$ peut être représentée comme :

$$\nabla V(x) = \begin{bmatrix} \nabla V_1(x) \\ \nabla V_2(x) \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 \\ a_{21}x_1 + a_{22}x_2 \end{bmatrix}$$

En posant $a_{22} = 2$, nous pouvons résoudre la fonction dérivée de la fonction de Lyapunov de la manière suivante :

$$\begin{aligned} \frac{dV(x)}{dt} = & - \left[\frac{6a_{11}}{(1+x_1^2)^3} + 2a_{21} \right] x_1^2 + \left[2a_{11} - \frac{6a_{12}}{(1+x_1^2)^3} - 4 - \frac{2a_{21}}{1+x_1^2} \right] x_1x_2 \\ & + \left(2a_{12} - \frac{4}{1+x_1^2} \right) x_2^2. \end{aligned}$$

Selon les équations (2.4) et (2.5), et en choisissant les paramètres $a_{11} = 2$, $a_{12} = a_{21} = 0$, nous obtenons :

$$\frac{dV(x)}{dt} = -\frac{12x_1^2}{(1+x_1^2)^3} - \frac{4x_2^2}{1+x_1^2}.$$

La fonction dérivée de la fonction de Lyapunov est définie négative. Ainsi, la fonction de Lyapunov $V(x)$ peut être résolue :

$$V(x) = \int_0^{x_1(x_2=0)} 2x_1 dx_1 \int_0^{x_2(x_1=x_1)} 2x_2 dx_2 = x_1^2 + x_2^2.$$

Il est clair que la fonction de Lyapunov résolue $V(x)$ est définie positive. Par conséquent, le système non linéaire est asymptotiquement stable.

2.3.2 Méthode de Zubov

Le domaine d'attraction d'un point fixe localement asymptotiquement stable est un objet important dans l'analyse du comportement des systèmes dynamiques. Un résultat essentiel a été obtenu par Vladimir Ivanovich (V.I.) Zubov (ZUBOV, 1964), qui a prouvé que pour les systèmes dynamiques induits par des équations différentielles ordinaires, le domaine d'attraction peut être représenté comme un level set d'une solution d'une équation différentielle partielle du premier ordre, l'équation de Zubov. Au-delà de la caractérisation du domaine d'attraction, la solution à l'équation de Zubov fournit également une fonction de Lyapunov.

V.I. Zubov a initialement proposé que si (ZUBOV, 1964) :

$$\nabla V f = -(1 - V)\phi \quad (2.6)$$

est vérifiée pour une fonction candidate de Lyapunov V et une fonction définie positive ϕ , alors $\{x \mid V(x) \leq 1\}$ est une caractérisation exacte de la région d'attraction pour la solution stationnaire $x^* = 0$ de f . Malheureusement, (2.6) n'admet presque jamais de solution explicite. Par conséquent, différentes méthodes approximatives ont été proposées afin d'estimer le domaine d'attraction. Plusieurs méthodes ont été développées pour résoudre ou approximer la solution de l'équation de Zubov. Dans ce qui suit, nous ne présentons que quelques approches en détail. Pour un examen complet des différentes méthodes, nous renvoyons le lecteur à (GENESIO et al., 1985 ; P. GIESL et HAFSTEIN, 2015b).

2.3.2.1 Première approche

Une première approche du problème est due à Zubov lui-même, qui suppose que le système autonome est écrit sous la forme :

$$\dot{x} = Ax + g(x) \quad (2.7)$$

avec les valeurs propres de A ayant des parties réelles négatives et $g(x)$ admettant une expansion de Taylor commençant par des termes de second degré. Dans ce cas, la solution de (2.6) peut être exprimée (au sens de Lyapunov) sous la forme de la série :

$$V(x) = V_2(x) + V_3(x) + \dots + V_m(x) + \dots \quad (2.8)$$

où $V_m(x)$ est une fonction homogène de degré m . Zubov a montré que pour chaque \hat{V} approximatif, dérivé de (2.8) tronqué au terme m , la frontière du domaine d'attraction se situe entre les surfaces (HAHN et al., 1967) :

$$\hat{V}(x) = \min_{\hat{V}=0} \hat{V}; \quad \hat{V}(x) = \max_{\hat{V}=0} \hat{V}. \quad (2.9)$$

Le calcul des formes successives $V_2, V_3, \dots, V_m, \dots$ est effectué en résolvant de façon séquentielle des équations linéaires en leurs coefficients obtenus par substitution dans (2.6). Un exemple d'illustration peut être trouvé dans (LEFFERTS, 1965).

Cette méthode a d'abord été appliquée par (MARGOLIS et VOGT, 1963) aux systèmes d'ordre deux. Le choix de la fonction g influence la forme des surfaces $\hat{V} = \text{constant}$ (HEWIT et STOREY, 1967, 1969). Quand m tend vers l'infini, ces surfaces convergent, non uniformément, vers la vraie frontière du domaine d'attraction (KERR et al., 1964). Le point essentiel de la méthode consiste à minimiser \hat{V} sous la contrainte $\hat{V} = 0$, comme dans d'autres procédures de Lyapunov. Une solution possible a été proposée par (RODDEN, 1964). Des applications de la méthode ci-dessus aux systèmes électriques se trouvent dans (DE SARKAR et RAO, 1971 ; YU et VONGSURIYA, 1967), tandis qu'une solution approximative similaire de l'équation de Zubov, en supposant la fonction V comme un rapport de polynômes, est proposée dans (VANNELLI et VIDYASAGAR, 1985).

2.3.2.2 Deuxième approche

Prenons l'équation du système autonome (1.9). Soit Z un domaine simplement connecté contenant un voisinage de l'origine. Le théorème suivant donne une condition suffisante pour que Z soit exactement le domaine d'attraction de l'origine.

Théorème 30 (Théorème de Zubov) Soit deux fonctions scalaires $V(x)$, $h(x)$. $V(x)$ est continue et définie positive dans $x \in Z$ et satisfait dans Z l'inégalité $0 < V(x) < 1$ ($x \neq 0$). $h(x)$ est définie pour tous les x finis, $h(0) = 0$, $h(x)$ est continue et positive pour $x \neq 0$. Pour $x \in Z$ nous avons (HAHN et al., 1967) :

$$\dot{V}(x) = -h(x)(1 - V(x))\sqrt{1 + \|f(x)\|^2}. \quad (2.10)$$

À mesure que $x \in Z$ s'approche d'un point de la frontière de Z ou dans le cas d'une région Z non bornée et que $|x| \rightarrow \infty$, $\lim V(x) = 1$, alors Z est exactement le domaine d'attraction de l'équilibre.

Une preuve peut être trouvée dans (HAHN et al., 1967). Le théorème suivant est une conséquence immédiate du Théorème 30.

Théorème 31 Soit la fonction $h(x)$ qui satisfait les hypothèses du Théorème 30. Supposons que la fonction $V(x)$ est définie positive dans Z et satisfait l'inégalité $0 \leq V(x) \leq 1$, $x \in Z$, ainsi que l'équation différentielle :

$$\sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i(x) = -h(x)(1 - V(x))\sqrt{1 + \|f(x)\|^2}.$$

Alors, la frontière du domaine d'attraction est définie par l'équation :

$$V(x) = 1. \quad (2.11)$$

Si le domaine d'attraction Z est tout \mathbb{R}^n , nous avons une stabilité asymptotique globale. Dans ce cas, la condition sur $V(x)$ est :

$$V(x) \rightarrow 1 \text{ quand } |x| \rightarrow \infty.$$

Il est également possible de travailler avec une fonction de Lyapunov différente. Par exemple, si nous définissons :

$$W(x) = -\ln(1 - V(x))$$

alors l'équation (2.10) devient :

$$\dot{W}(x) = -h(x)\sqrt{1 + \|f\|^2}$$

et la condition (2.11) définissant la frontière devient $W(x) \rightarrow \infty$.

Il est essentiel de noter que, bien que plusieurs fonctions $h(x)$ puissent respecter les conditions énoncées, chaque choix de $h(x)$ influence potentiellement la forme et les propriétés de $V(x)$. Cela signifie que différentes fonctions h peuvent conduire à différentes fonctions V qui peuvent ou non satisfaire aux critères du Théorème 30 et du Théorème 31. Le but principal du théorème est d'identifier un domaine d'attraction en utilisant une combinaison appropriée de $h(x)$ et $V(x)$. Étant donné que plusieurs fonctions h peuvent être utilisées, le choix est crucial. Il est possible que certains choix de h ne fournissent pas une V utile, tandis que d'autres le font. En pratique, cela peut nécessiter une exploration ou une justification pour le choix de h , surtout si nous cherchons une V qui offre une perspective particulière sur le système. Un exemple d'illustration de cette méthode peut être trouvé dans (HAHN et al., 1967).

2.3.2.3 Troisième approche

Cette nouvelle approche offre un cadre plus général pour analyser la stabilité, permettant de traiter des situations où la stabilité ne peut être réduite à un seul point, mais implique un ensemble d'états. Le résultat suivant apparaît pour la première fois dans (GRUJIC et al., 1994).

Nous considérons un flot continu et invariant dans le temps, $\phi : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. Le flot ϕ satisfait $\phi(0, x) = x$ et $\phi(t, \phi(s, x)) = \phi(t + s, x)$ pour tout $t, s \in \mathbb{R}$ et $x \in \mathbb{R}^n$. Un flot, dans ce contexte, est une fonction continue ϕ qui décrit l'évolution d'un point dans l'espace \mathbb{R}^n au fil du temps, respectant des propriétés spécifiques (BHATIA et HÁJEK, 2006 ; BHATIA et SZEGÖ, 2002). Un ensemble $\mathcal{K} \subset \mathbb{R}^n$ est dit positivement invariant si pour tout $x \in \mathcal{K}$, $\phi(t, x) \in \mathcal{K}$ pour tout $t \in \mathbb{R}^+$.

Théorème 32 (Théorème de Zubov) Soit $\mathcal{K} \subset \mathbb{R}^n$ un ensemble compact positif et invariant du système autonome (1.9) et $\mathcal{A} \subset \mathbb{R}^n$ un ensemble ouvert positif et invariant de (1.9) contenant un voisinage de \mathcal{K} . \mathcal{A} est le domaine d'attraction de l'ensemble stable asymptotiquement \mathcal{K} si et seulement si il existe une fonction lisse $V : \mathcal{A} \rightarrow \mathbb{R}$ et une fonction C^k $h : \mathcal{A} \rightarrow \mathbb{R}$ telles que :

- $V(\mathcal{K}) = h(\mathcal{K}) = 0$.
- $-1 < V(x) < 0$ et $h(x) > 0$ pour tout $x \in \mathcal{A} \setminus \mathcal{K}$.
- Si \mathcal{A} est borné alors $\lim_{x \rightarrow \partial \mathcal{A}} V(x) = -1$, sinon $\lim_{\|x\| \rightarrow +\infty} V(x) = -1$.
- L'équation de Zubov :

$$\nabla V(x) \cdot f(x) = h(x)(1 + V(x)) \quad (2.12)$$

est vérifiée pour tout $x \in \mathcal{A}$.

De plus, nous avons $V^{-1}([-1; 0]) = \mathcal{A}$ et $V^{-1}(0) = \mathcal{K}$.

2.3.2.4 Exemple d'illustration 12

Nous examinons un exemple en utilisant l'équation de Zubov (2.12). Considérons le système avec $\mu > 0$:

$$\begin{cases} \dot{x}_1 &= x_2 + \mu x_1(1 - x_1^2 - x_2^2) \\ \dot{x}_2 &= -x_1 + \mu x_2(1 - x_1^2 - x_2^2) \end{cases} \quad (2.13)$$

Choisissons la fonction h suivante :

$$h(x_1, x_2) = \frac{2\mu(1 - x_1^2 - x_2^2)^2(1 + x_1^2 + x_2^2)}{x_1^2 + x_2^2}.$$

L'équation de Zubov associée au système (2.13) est donc donnée par :

$$\frac{\partial V}{\partial x_1}(x_2 + \mu x_1(1 - x_1^2 - x_2^2)) + \frac{\partial V}{\partial x_2}(-x_1 + \mu x_2(1 - x_1^2 - x_2^2)) = h(x_1, x_2)(1 + V).$$

Une solution de l'équation de Zubov est donnée par :

$$V(x_1, x_2) = e^{-\frac{(1 - x_1^2 - x_2^2)^2}{x_1^2 + x_2^2}} - 1$$

où $\mathcal{K} = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 = 1\}$, et le domaine d'attraction est donné par $\mathcal{A} = \mathbb{R}^2 \setminus \{(0, 0)\}$. Ainsi, $\{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 = 1\}$ est un cycle limite asymptotiquement stable.

Dans (CAMILLI et al., 2001) et (MALISOFF, 2003), la méthode de Zubov a été étendue à des systèmes avec perturbation déterministe variant dans le temps et dans (CAMILLI et GRÜNE, 2002) à des systèmes avec perturbation stochastique. Une première application aux systèmes de contrôle a été donnée dans (SONTAG, 1983), où l'équation de Zubov (2.10) a été utilisée sous une forme intégrale. La version différentielle originale de (2.10) a été étudiée pour les systèmes avec perturbations dans (GRUNE et WURTH, 2000 ; GRÜNE, 2004), et (sous des hypothèses plus générales) dans (CAMILLI et al., 2008).

Une variante de la méthode de Zubov pour les systèmes avec des commandes et des perturbations stochastiques a été étudiée dans (CAMILLI et al., 2006). De plus, les auteurs dans (CAMILLI et al., 2007) ont introduit une méthode numérique basée sur l'équation de Zubov. Cette approche a été initialement développée pour les scénarios perturbés de manière déterministe, comme présenté dans (CAMILLI et al., 2001). La méthode a depuis été étendue et appliquée avec succès à d'autres cadres. Par exemple, le travail exposé dans (CAMILLI et al., 2004) explore l'application de cette méthode à un pendule inversé contrôlé. Dans un contexte différent, l'étude présentée dans (GRÜNE et al., 2007) applique cette méthode à un problème de contrôle stochastique en économie mathématique. Une autre approche numérique a été introduite dans (P. GIESL, 2007), mais elle n'est jusqu'à présent applicable qu'au cadre original de Zubov sans perturbation ni contrôle.

Les méthodes existantes de résolution de l'équation de Zubov ont certaines contraintes. Notamment, la plupart des méthodes font face à l'obstacle de la dimensionnalité croissante, c'est-à-dire que la complexité de la solution augmente exponentiellement avec la dimension de l'espace d'état.

Il convient de noter que des idées d'utilisation de l'apprentissage profond (Deep Learning (DL)) pour atténuer le problème de la dimensionnalité croissante, ont été appliquées avec succès à diverses équations aux dérivées partielles au cours des dernières années (HAN et al., 2016 ; NAKAMURA-ZIMMERER et al., 2021 ; RAISSI et al., 2019).

2.3.3 Méthode de la somme des carrés

La méthode de programmation de la somme des carrés (Sum Of Squares (SOS)) est une méthode numérique de recherche des fonctions de Lyapunov qui tire parti du fait qu'une fonction polynomiale est non négative si elle peut être exprimée sous la forme d'une somme des carrés de fonctions polynomiales. La méthode SOS consiste à représenter la fonction de Lyapunov comme une somme des carrés de fonctions polynomiales, puis à utiliser des techniques d'optimisation pour trouver les coefficients des fonctions polynomiales qui satisfont aux conditions de Lyapunov. Il est important de noter que cette méthode suppose que le système étudié soit polynomial.

Un polynôme multivarié $p(x_1, \dots, x_n) \triangleq p(x)$ est une somme des carrés, si il existe des polynômes $f_1(x), \dots, f_m(x)$ tels que :

$$p(x) = \sum_{i=1}^m f_i^2(x). \quad (2.14)$$

L'ensemble des polynômes somme des carrés en n variables est un cône convexe. L'existence d'une décomposition SOS (2.14) peut être montrée comme étant équivalente à l'existence d'une matrice Q semi-définie positive telle que :

$$p(x) = Z^T(x)QZ(x) \quad (2.15)$$

où $Z(x)$ est le vecteur de monômes de degré inférieur ou égal à $\deg(p)/2$, c'est-à-dire, ses entrées sont de la forme $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$, où les α sont des entiers non négatifs et $\alpha_1 + \dots + \alpha_n \leq \deg(p)/2$.

Exprimer un polynôme SOS comme une forme quadratique en (2.15) a également été désigné comme la méthode de la matrice de Gram (CHOI et al., 1995 ; POWERS et WÖRMANN, 1998). La décomposition (2.14) peut être facilement convertie en (2.15) et vice versa. Cette équivalence rend une décomposition SOS calculable en utilisant la programmation semi-définie positive (PARRILO, 2000), puisque trouver une matrice symétrique $Q \geq 0$ soumise à la contrainte affine (2.15) n'est rien d'autre qu'un problème de programmation semi-définie positive.

La programmation semi-définie positive (Semidefinite programming (SDP)) (FREUND, 2004) est une forme de programmation mathématique, une branche de l'optimisation convexe, où l'objectif est de minimiser une fonction objectif linéaire sur un ensemble de matrices semi-définies positives.

Pour être plus précis, un problème typique de SDP peut être formulé comme suit :

SDP

Minimiser $c^T x$ sous les contraintes :

$$A_0 + \sum_{i=1}^n x_i A_i \geq 0 \text{ et } x \in \mathbb{R}^n.$$

Ici, x est le vecteur de variables d'optimisation, c^T est la transposée d'un vecteur de coefficients donnés, et les A_i sont des matrices symétriques données. La notation " ≥ 0 " indique que la matrice est semi-définie positive, c'est-à-dire que toutes ses valeurs propres sont non négatives.

La puissance de la SDP réside dans le fait qu'elle peut modéliser une grande variété de problèmes d'optimisation et qu'elle est convexe.

Un programme de somme des carrés est un problème d'optimisation convexe de la forme suivante :

OPTIMISATION

Minimiser $\sum_{j=1}^J w_j c_j$ sous contrainte que : (2.16)

$$a_{i,0}(x) + \sum_{j=1}^J a_{i,j}(x) c_j \text{ est SOS, pour } i = 1, \dots, I. \quad (2.17)$$

Ici, c_j sont les variables réelles de décision scalaires, w_j sont des nombres réels donnés et $a_{i,j}(x)$ sont des polynômes donnés (avec des coefficients fixes). Il existe d'autres formes canoniques équivalentes de programmes SOS dans (PAPACHRISTODOULOU et al., 2013 ; PRAJNA et al., 2002). Alors que la conversion de programmes SOS en programmes semi-définis positifs peut être manuellement réalisée pour de petites instances ou adaptée pour des classes de problèmes spécifiques, une telle conversion peut être assez lourde à réaliser en général. Il est donc souhaitable d'avoir une aide algorithmique qui effectue automatiquement cette conversion pour des programmes SOS généraux.

L'un des outils disponible à cette fin est SOSTOOLS. Il automatise la conversion du programme SOS en SDP, appelle le solveur SDP, et convertit la solution SDP en la solution du programme SOS d'origine. La version actuelle de SOSTOOLS utilise SeDuMi (STURM, 1999), SDPT3 (TOH et al., 1999), CSDP (BORCHERS, 1999), SDPNAL (ZHAO et al., 2010), SDPNAL+ (YANG et al., 2015), SDPA (YAMASHITA et al., 2003) et MOSEK (APS, 2019) comme solveur SDP. L'interface utilisateur de SOSTOOLS a été conçue pour être aussi simple, facile à utiliser et transparente que possible, tout en conservant un haut degré de flexibilité.

SOSTOOLS trouve des applications dans plusieurs problèmes de la théorie du contrôle. Ces problèmes incluent l'analyse de stabilité de systèmes non linéaires, hybrides, et à retard (PAPACHRISTODOULOU, 2004; PAPACHRISTODOULOU et PRAJNA, 2002; PARRILO, 2000; PRAJNA et PAPACHRISTODOULOU, 2003), l'analyse de robustesse (PAPACHRISTODOULOU et PRAJNA, 2002; PARRILO, 2000; PRAJNA et PAPACHRISTODOULOU, 2003), l'estimation du domaine d'attraction (PARRILO, 2000; SEILER, 2003), l'analyse et la synthèse LPV (WU et PRAJNA, 2004), la synthèse non linéaire (JARVIS-WLOSZEK et al., 2003; PRAJNA et al., 2004; PRAJNA et al., 2004), la vérification de la sécurité (PRAJNA et JADBABAIE, 2004), et la validation de modèle (PRAJNA, 2006). D'autres domaines dans lesquels SOSTOOLS est applicable sont, par exemple, la preuve de théorèmes géométriques (PARRILO et PERETZ, 2004) et la théorie de l'information quantique (DOHERTY et al., 2002).

Dans (TAN et PACKARD, 2006), basée sur la décomposition SOS, une méthode est présentée pour rechercher des fonctions de Lyapunov polynomiales qui agrandissent un domaine d'attraction d'un système polynomial non linéaire.

2.3.3.1 Exemple d'illustration 13

Nous présentons maintenant de quelle manière la recherche d'une fonction de Lyapunov peut être formulée comme un programme de sommes des carrés. Les lecteurs sont renvoyés à (PAPACHRISTODOULOU et PRAJNA, 2002; PARRILO, 2000; PRAJNA et PAPACHRISTODOULOU, 2003) pour des discussions et des extensions plus détaillées. Pour notre exemple, considérons le système (HENRION et GARULLI, 2005) :

$$\begin{cases} \dot{x}_1 &= -x_1^3 - x_1x_3^2 \\ \dot{x}_2 &= -x_2 - x_1^2x_2 \\ \dot{x}_3 &= -x_3 - \frac{3x_3}{x_3^2+1} + 3x_1^2x_3 \end{cases} \quad (2.18)$$

qui a un point équilibre à l'origine. Notons que la linéarisation de (2.18) a une valeur propre nulle, et ne peut donc pas être utilisée pour analyser la stabilité locale de l'équilibre. Supposons maintenant que nous sommes intéressés par une fonction de Lyapunov quadratique $V(x)$ pour prouver la stabilité du système. Alors $V(x)$ doit satisfaire :

$$V - \epsilon(x_1^2 + x_2^2 + x_3^2) \geq 0 \quad (2.19)$$

$$-\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3 \geq 0. \quad (2.20)$$

La première inégalité, avec ϵ étant une constante quelconque supérieure à zéro (dans ce qui suit, nous choisissons $\epsilon = 1$), est nécessaire pour garantir la positivité de $V(x)$. Nous formulons un programme SOS qui calcule une fonction de Lyapunov pour ce système, en remplaçant les conditions de non-négativité ci-dessus par des conditions SOS. Cependant, notons que \dot{x}_3 est une fonction rationnelle, et donc (2.20) n'est pas une expression polynomiale. Mais comme $x_3^2 + 1 > 0$ pour tout x_3 , nous pouvons simplement reformuler (2.20) comme suit :

$$(x_3^2 + 1) \left(-\frac{\partial V}{\partial x_1}\dot{x}_1 - \frac{\partial V}{\partial x_2}\dot{x}_2 - \frac{\partial V}{\partial x_3}\dot{x}_3 \right) \geq 0. \quad (2.21)$$

Ensuite, nous paramétrons la fonction candidate de Lyapunov quadratique $V(x)$ par quelques coefficients réels inconnus c_1, \dots, c_6 :

$$V(x) = c_1x_1^2 + c_2x_1x_2 + c_3x_2^2 + c_4x_1x_3 + c_5x_2x_3 + c_6x_3^2. \quad (2.22)$$

Le programme SOS suivant peut être formulé :

PROGRAMME

Trouver un polynôme $V(x)$ (équivalent à trouver c_1, \dots, c_6), tel que :

$$V(x) - (x_1^2 + x_2^2 + x_3^2) \text{ est SOS,}$$

$$(x_3^2 + 1) \left(-\frac{\partial V}{\partial x_1} \dot{x}_1 - \frac{\partial V}{\partial x_2} \dot{x}_2 - \frac{\partial V}{\partial x_3} \dot{x}_3 \right) \text{ est SOS.}$$

Dans cet exemple, l'algorithme SOSTOOLS renvoie $V(x) = 5.5489x_1^2 + 4.1068x_2^2 + 1.7945x_3^2$ comme une fonction de Lyapunov qui prouve la stabilité du système.

Nous avons pris comme exemple SOSTOOLS, disponible sous forme de "toolbox" MATLAB (PRAJNA et al., 2004), mais il existe d'autres programmes tels que YALMIP (LOFBERG, 2004), PENBMI (KOCVARA et al., 2005) ou encore les techniques de récupération de représentation SOS exacte, présentées dans (E. KALTOFEN et al., 2008; E. L. KALTOFEN et al., 2012). Une "toolbox" dans le contexte de MATLAB fait référence à une collection de fonctions et d'algorithmes spécifiques, regroupés pour faciliter leur utilisation et leur application à des problèmes particuliers. Chaque toolbox est conçue pour un domaine ou une application spécifique, offrant des outils et des ressources qui permettent aux utilisateurs d'implémenter et de résoudre des problèmes spécifiques de manière plus efficace et simplifiée.

Pour un aperçu des méthodes SOS, nous renvoyons le lecteur à (ANDERSON et PAPACHRISTODOULOU, 2015), et pour un aperçu des autres méthodes polynomiales, nous renvoyons le lecteur à (KAMYAR et PEET, 2014).

2.3.4 Méthode continue et affine par morceaux

La méthode de la fonction de Lyapunov continue et affine par morceaux (Continuous Piecewise Affine (CPA)) offre une approche pour étudier la stabilité des systèmes dynamiques. En brisant la complexité des systèmes non-linéaires en des sous-régions plus simples et linéaires, elle offre une représentation plus flexible de la fonction de Lyapunov.

Nous définissons une grille sur $K \subset \mathbb{R}^n$ et calculons pour chaque point x dans la grille la valeur que V doit prendre en x . Les valeurs pour V sur la grille peuvent ensuite être utilisées pour étendre V à tout K de sorte que V est une fonction de Lyapunov CPA sur K . Dans cette section, nous fournissons les détails de cette construction présentée dans (S. F. HAFSTEIN, 2009).

Soit x_0, \dots, x_k une collection de points indépendants linéairement affines dans \mathbb{R}^n , telle que $\sum_{i=1}^k c_i(x_i - x_0) = 0$ implique que $c_i = 0$ pour $i = 1, \dots, k$. La combinaison convexe de x_0, \dots, x_k , c'est-à-dire l'ensemble $\text{co}\{x_0, \dots, x_k\} := \left\{ \sum_{i=0}^k c_i x_i \mid \text{tous les } c_i \geq 0 \text{ et } \sum_{i=0}^k c_i = 1 \right\}$, est un k -simplexe dans \mathbb{R}^n .

Soit $\mathcal{T} = \{T_1, \dots, T_m\}$ un ensemble de k -simplexes tels que $K = \bigcup_{i=1}^m T_i$. Supposons en outre que pour chaque i, j tels que $i \neq j$, nous avons $T_i \cap T_j = \emptyset$ ou que $T_i \cap T_j$ est un h -simplexe avec $h < k$. \mathcal{T} est alors une triangulation de K . Nous définissons en outre $V_{\mathcal{T}} := \{x \in \mathbb{R}^n \mid x \text{ est un sommet d'un simplexe dans } \mathcal{T}\}$ et appelons $V_{\mathcal{T}}$ l'ensemble de sommets pour la triangulation.

Pour une triangulation donnée $\mathcal{T} = \{T_1, \dots, T_m\}$ et une fonction $V_0 : V_{\mathcal{T}} \rightarrow \mathbb{R}$, nous pouvons maintenant étendre de manière unique V_0 à une fonction affine continue par morceaux $V : K \rightarrow \mathbb{R}$ qui est C^∞ à l'intérieur de chaque simplexe T_i . Plus précisément :

- Si $x \in V_{\mathcal{T}}$ alors $V(x) = V_0(x)$.
- Pour chaque $T_i \in \mathcal{T}$, il existe une application linéaire $A_i : \mathbb{R}^n \rightarrow \mathbb{R}$ et $b_i \in \mathbb{R}$ telle que $V(x) = A_i(x) + b_i$ pour tout $x \in T_i$ et telle que V est continue sur K .

Nous sommes intéressés par les systèmes avec un équilibre asymptotiquement stable à l'origine et donc par les fonctions de Lyapunov qui possèdent un minimum à l'origine. Étant donné qu'un minimum local pour une fonction CPA peut être atteint soit à un sommet soit identiquement sur une arête, nous privilégions des configurations où l'origine est un sommet de notre triangulation pour éviter le cas échéant de minima sur les arêtes. Pour une triangulation donnée \mathcal{T} de K , nous dénotons l'ensemble de toutes les fonctions affines continues par morceaux telles que décrites ci-dessus par $\text{CPA}[\mathcal{T}]$ et identifions chaque fonction avec la fonction correspondante $V_0 : V_{\mathcal{T}} \rightarrow \mathbb{R}$ par $V \sim (V_0(x))_{x \in V_{\mathcal{T}}}$. Pour chaque fonction dans $\text{CPA}[\mathcal{T}]$, nous lui associons une fonction spécifique V_0 définie sur $V_{\mathcal{T}}$ qui prend des valeurs dans \mathbb{R} . Cette association signifie que pour tout x dans $V_{\mathcal{T}}$, la valeur de la fonction est $V_0(x)$. Nous utilisons le symbole \sim pour indiquer cette association.

Le résultat énoncé par (S. F. HAFSTEIN, 2009) nous indique qu'il suffit de déterminer une fonction $V_0 : V_{\mathcal{T}} \rightarrow \mathbb{R}$ qui satisfait certaines inégalités en chaque sommet $x \in V_{\mathcal{T}}$ pour construire une fonction de Lyapunov pour un système autonome (1.9), car l'interpolation CPA de V_0 sur K est une fonction de Lyapunov CPA pour (1.9) sur tout K .

Nous désignons les entiers positifs par \mathbb{N} et les nombres réels strictement positifs par $\mathbb{R}_{>0}$. Pour un vecteur $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$, nous définissons les normes $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ pour $p \geq 1$, et nous notons pour un ensemble $A \subseteq \mathbb{R}^n$ son diamètre par $\text{diam}_2(A) := \sup_{x,y \in A} \|x - y\|_2$.

Théorème 33 *Considérons le système autonome (1.9) avec triangulation $\mathcal{T} = \{T_1, \dots, T_m\}$. Soit $V \sim (V_0(x))_{x \in V_{\mathcal{T}}} \in \text{CPA}[\mathcal{T}]$. Pour chaque $T_i \in \mathcal{T}$, définissons les constantes $h_i := \text{diam}_2(T_i)$ et (S. F. HAFSTEIN, 2009) :*

$$E_i := \frac{nM_i^2}{2} h_i^2 \quad \text{où} \quad (2.23)$$

$$M_i \geq \max_{m,r,s=1,2,\dots,n} \max_{z \in T_i} \left| \frac{\partial^2 f_m}{\partial x_r \partial x_s}(z) \right|$$

Supposons que pour un simplexe $T_i = \text{co}\{x_0, x_1, \dots, x_k\} \in \mathcal{T}$, l'inégalité suivante est vraie pour chaque sommet $x_j \in T_i$:

$$\nabla V_i \cdot f(x_j) + E_i \|\nabla V_i\|_1 < 0. \quad (2.24)$$

Par conséquent, nous obtenons :

$$\nabla V_i \cdot f(x) < 0 \text{ pour tout } x \in T_i.$$

Si l'inégalité (2.24) est vraie pour $T_{i_1}, T_{i_2}, \dots, T_{i_m}$, nous avons $\dot{V} < 0$ pour tout $x \in \left(\bigcup_{j=1}^m T_{i_j}\right)$.

L'égalité $V(0) = 0$, les inégalités $V(x) > 0$ et (2.24), peuvent être traitées comme des contraintes linéaires d'un programme d'optimisation linéaire ([Linear Programming \(LP\)](#)) avec les valeurs V_i de V sur $V_{\mathcal{T}} \setminus \{0\}$ comme variables. Comme ces valeurs de V sur $V_{\mathcal{T}}$ déterminent $V_{\mathcal{T}} \in \text{CPA}[\mathcal{T}]$, un problème LP peut être utilisé pour paramétrer une fonction de Lyapunov CPA. C'est l'idée de la méthode continue et affine par morceaux.

L'auteur dans ([S. F. HAFSTEIN, 2004, 2005](#)) a prouvé que pour des systèmes non linéaires avec un équilibre exponentiellement ou asymptotiquement stable, cette méthode aboutit toujours à la construction d'une fonction de Lyapunov si la triangulation est suffisamment fine et si un voisinage arbitrairement petit de l'origine est exclu du domaine de la fonction de Lyapunov. Si ce voisinage est exclu, alors la fonction de Lyapunov résultante assure la stabilité de l'équilibre, mais pas nécessairement la stabilité asymptotique.

Les auteurs dans ([P. GIESL et HAFSTEIN, 2012](#); [P. A. GIESL et HAFSTEIN, 2014](#)) ont prouvé qu'en utilisant une autre forme de triangulation à l'équilibre, une méthode CPA révisée peut calculer une fonction de Lyapunov CPA pour tout système non linéaire avec un équilibre exponentiellement stable. Ceci en remplissant toutes les conditions d'une fonction de Lyapunov, également dans un voisinage de l'équilibre. En particulier, un système avec $f \in C^2$ et un équilibre exponentiellement stable à l'origine possède une fonction de Lyapunov linéaire localement en forme de cône.

La méthode CPA a été étendue à des systèmes plus généraux : pour des systèmes commutés de manière arbitraire, non autonomes ([S. F. HAFSTEIN, 2009](#)), pour les inclusions différentielles ([BAIER et al., 2012](#)), pour les systèmes à temps discret ([P. GIESL et HAFSTEIN, 2014](#)) et pour les fonctions de Lyapunov ISS ([H. LI et al., 2014, 2015](#)).

Outre la méthode de programmation linéaire, il existe d'autres moyens de calculer les valeurs $V(x_i)$ aux sommets $x_i \in V_{\mathcal{T}}$. Après cette étape, nous pouvons vérifier que l'égalité $V(0) = 0$ ainsi que les inégalités $V(x) > 0$ et (2.24) sont respectées. Ces valeurs $V_i = V(x_i)$ doivent être déterminées de telle sorte que la fonction $V \in \text{CPA}[\mathcal{T}]$ (l'interpolation CPA) ait une probabilité d'être une fonction de Lyapunov. Cela a été l'objet de plusieurs publications récentes ([BJÖRNSSON et al., 2014](#); [S. HAFSTEIN et al., 2014](#)), en utilisant les théorèmes inverses de Massera ([MASSERA, 1949](#)) et de Yoshizawa ([YOSHIZAWA, 1966](#)), et en utilisant la méthode de la fonction de base radiale ([BJÖRNSSON et al., 2014](#); [P. GIESL et HAFSTEIN, 2015a](#)).

La mise en œuvre concrète de la méthode de la fonction de Lyapunov CPA nécessite une application minutieuse de techniques numériques. En raison de la complexité de cette méthode, nous ne détaillons pas ici d'exemple complet d'utilisation de la méthode. Les lecteurs intéressés qui souhaitent approfondir leur compréhension de la méthode CPA peuvent se référer aux exemples pratiques contenus dans les travaux de recherche cités précédemment dans cette section. Ces documents offrent une démonstration approfondie de la méthode CPA, en illustrant chaque étape de la procédure avec des exemples concrets et en discutant des défis potentiels qui peuvent survenir lors de l'application de cette méthode.

2.3.5 Méthode de la fonction de base radiale

Dans cette section, nous discutons de la fonction de base radiale ([Radial Basis Function \(RBF\)](#)), une méthode de collocation sans maillage pour résoudre les équations aux dérivées partielles linéaires ([Partial Differential Equation \(PDE\)](#)).

La collocation sans maillage fournit un outil puissant pour résoudre efficacement les problèmes d'interpolation généralisés, en particulier les PDE linéaires ([FRANKE et SCHABACK, 1998](#); [WENDLAND, 2004](#); [ZONGMIN, 1992](#)). La collocation sans maillage, en utilisant des fonctions de base radiales, a été

utilisée avec succès pour construire des fonctions de Lyapunov (classiques) sur le bassin d'attraction d'un équilibre exponentiellement stable.

La méthode RBF utilise la collocation pour résoudre l'équation de Zubov :

$$\nabla V(x) * f(x) = -p(x) \quad (2.25)$$

pour V , où f provient du système autonome $\dot{x} = f(x)$ et p est une fonction définie positive. La méthode RBF approxime la solution de (2.25) en utilisant une collocation sans maillage.

Définition 34 (Fonctions de Wendland) Soit $l \in \mathbb{N}$, $k \in \mathbb{N}_0$. Nous définissons par récurrence (WENDLAND, 1998) :

$$\psi_{l,0}(r) = (1 - r)_+^l \quad (2.26)$$

et

$$\psi_{l,k+1}(r) = \int_r^1 t \psi_{l,k}(t) dt \quad (2.27)$$

pour $r \in \mathbb{R}_0^+$. Ici, nous fixons $x_+ = x$ pour $x \geq 0$ et $x_+ = 0$ pour $x < 0$. En posant $l := \lfloor \frac{n}{2} \rfloor + k + 1$, $\Phi(x) := \psi_{l,k}(c\|x\|_2)$ appartient à $C^{2k}(\mathbb{R}^n)$ pour tout $c > 0$.

Premièrement, nous fixons un ensemble compact \mathcal{C} et choisissons une grille, c'est-à-dire un ensemble fini de N points distincts deux à deux $X = \{x_1, \dots, x_N\}$ dans \mathcal{C} , ne contenant pas d'équilibre. Ensuite, nous fixons une fonction de base radiale de Wendland $\Phi(x) = \psi(\|x\|_2)$ (P. GIESL, 2008b). Pour $n = 2, 3$ un choix approprié est :

$$\begin{aligned} \psi_1(r) &:= r^{-1} \psi'(cr) = -22c^2(1 - cr)^7 + [16c^2r^2 + 7cr + 1] \\ \psi_2(r) &:= r^{-1} \psi_1'(cr) = 528c^4(1 - cr)^6 + [6cr + 1] \end{aligned}$$

où $x^+ := \max(0, x)$ et $c > 0$ est un paramètre fixé. Ces attributions pour ψ_1 et ψ_2 sont les valeurs par défaut dans le logiciel (BJÖRNSSON et al., 2015).

Deuxièmement, la matrice $A = (a_{jk})_{j,k=1,2,\dots,N}$ est définie. Avec $z_{jk} := y_j - y_k$ et $z_{jk} := \|z_{jk}\|_2$, nous définissons :

$$a_{jk} = \psi_2(z_{jk})[z_{jk} * f(y_j)][z_{kj} * f(y_k)] - \psi_1(z_{jk})[f(y_j) * f(y_k)].$$

Troisièmement, l'équation linéaire :

$$A\beta = \alpha, \text{ où } \alpha_i = p(y_i)$$

est résolue pour β . La fonction $V_R(x)$ est définie comme suit :

$$V_R(x) := \sum_{k=1}^N \beta_k (y_k - x) * f(y_k) \psi_1(\|x - y_k\|_2) + V_0$$

où la constante V_0 est choisie telle que $V_R(0) = 0$.

La fonction $V_R(x)$ est maintenant une fonction de Lyapunov pour le système sur $co \mathcal{X}$, à l'exception d'un petit voisinage de l'origine, si la distance de remplissage :

$$h_{\mathcal{X}} := \max_{x \in co \mathcal{X}} \min_{k=1,2,\dots,N} \|x - y_k\|_2$$

des points de collocation dans \mathcal{X} est assez petite.

La mise en œuvre concrète de la méthode RBF pour trouver une fonction de Lyapunov nécessite, comme la méthode CPA, une application minutieuse de techniques numériques. En raison de la complexité de cette méthode, nous ne détaillons pas ici d'exemple complet d'utilisation de la méthode. Les lecteurs intéressés qui souhaitent approfondir leur compréhension de la méthode RBF peuvent se référer aux exemples pratiques contenus dans (BJÖRNSSON et al., 2015 ; P. GIESL, 2004, 2008a ; P. GIESL et HAFSTEIN, 2015a ; MOHAMMED et GIESL, 2015).

2.3.6 Synthèse des méthodes numériques

Au cours de cette étude, nous examinons différentes méthodes numériques développées pour l'analyse de la stabilité des systèmes. Bien que chaque méthode présente ses avantages propres, elle est également accompagnée de limites qui peuvent influencer le choix de la méthode en fonction du contexte d'application. La [Table 3.1](#) ci-dessous présente une synthèse des avantages et inconvénients des méthodes numériques étudiées dans cette section. Cette synthèse permet d'offrir une vision claire des forces et faiblesses de chaque méthode, facilitant ainsi le choix de la méthode la plus appropriée selon le problème à traiter.

Méthodes	Avantages	Inconvénients
Gradient Variable	<ul style="list-style-type: none"> • Procédure pour trouver une fonction de Lyapunov 	<ul style="list-style-type: none"> • Cette procédure rend la méthode conservatrice
Zubov	<ul style="list-style-type: none"> • Estimation analytique du domaine de stabilité 	<ul style="list-style-type: none"> • Choix de la fonction h
SOSTools	<ul style="list-style-type: none"> • Facilité d'utilisation et outil disponible gratuitement sur internet • Temps de calcul 	<ul style="list-style-type: none"> • Le système étudié doit être polynomial
CPA	<ul style="list-style-type: none"> • La fonction de Lyapunov construite est valide et fiable pour l'analyse de stabilité 	<ul style="list-style-type: none"> • Exigeante sur le temps de calcul \rightarrow pas pratique pour les systèmes complexes • Peu adaptée pour des types de stabilité complexes comme exponentielle et ISS
RBF	<ul style="list-style-type: none"> • Moins exigeante sur le temps de calcul 	<ul style="list-style-type: none"> • La fonction de Lyapunov construite est une approximation, une vérification a posteriori est nécessaire

TABLE 2.1 – Avantages et inconvénients des méthodes numériques

2.4 Méthodes automatiques

La stabilité des systèmes est un enjeu crucial en automatique, offrant des garanties sur le comportement des systèmes face à des perturbations ou des incertitudes. Depuis la proposition des premières méthodes pour étudier la stabilité, les fonctions de Lyapunov sont devenues un outil incontournable pour analyser la stabilité des systèmes non linéaires. Ces fonctions offrent une manière élégante et efficace d'évaluer la stabilité d'un point d'équilibre sans avoir besoin de résoudre les équations différentielles qui décrivent le système.

Cependant, la détermination d'une fonction de Lyapunov appropriée pour un système donné n'est pas une tâche triviale. Dans de nombreux cas, il est extrêmement difficile, voire impossible, de trouver une telle fonction par des méthodes analytiques. Cela a ouvert la porte à des techniques théoriques et numériques pour faciliter cette recherche, dont nous avons expliqué quelques méthodes.

Les approches théoriques, telles que les méthodes de linéarisation, de Krasovski et de LaSalle, fournissent des perspectives analytiques sur le problème, mais elles ont souvent des limites dans leur applicabilité à des systèmes complexes ou fortement non linéaires. Les méthodes numériques, telles que la méthode du gradient variable ou celle de la somme des carrés, ont élargi le champ des possibilités en permettant d'approcher la solution de manière algorithmique.

De notre point de vue, ces méthodes constituent un point de départ structurel pour la mise en œuvre de techniques plus évoluées. En effet, les méthodes décrites précédemment peuvent devenir trop conservatrices pour être utilisables dans le cas de systèmes industriels complexes pour estimer un domaine d'attraction suffisamment grand. De plus, dans la plupart des situations, les problèmes de stabilité plus complexes ne sont pas traités ou sont compliqués à mettre en œuvre, tels que la stabilité exponentielle ou la stabilité entrée - état (**ISS**). Aussi, ces approches paraissent compliquées à appliquer pour les systèmes complexes réels rencontrés en milieu industriel où une certaine flexibilité est requise. Enfin, malgré leur puissance, elles peuvent être coûteuses en termes de temps de calcul et ne garantissent pas toujours la découverte d'une fonction de Lyapunov ainsi que son domaine de définition associé, qui est un sous-domaine d'attraction (**DOA**).

Ainsi, il pourrait être intéressant de disposer d'outils faciles à utiliser de manière à calculer des fonctions de Lyapunov moins dépendantes de leur forme et avec moins d'hypothèses sur la forme du système.

L'évolution rapide des techniques d'AI, en particulier des réseaux de neurones et de l'apprentissage automatique, offre une nouvelle approche pour aborder ce problème. La capacité des réseaux de neurones à approximer n'importe quelle fonction en fait des candidats naturels pour la modélisation des fonctions de Lyapunov. Par ailleurs, les algorithmes d'optimisation, tels que ceux basés sur des essais de particules ou des techniques génétiques, offrent des outils supplémentaires pour la recherche de la meilleure fonction possible. L'un des premiers articles utilisant l'AI pour calculer une fonction de Lyapunov est (**PROKHOROV, 1994**), où est décrite une "Lyapunov Machine", qui est un réseau de neurones artificiels spécialement conçu pour l'approximation de la fonction de Lyapunov. L'auteur indique que l'algorithme proposé, la "Lyapunov Machine", présente une complexité algorithmique importante parmi d'autres problèmes à résoudre et reporte leur résolution à des travaux ultérieurs.

Alors que les méthodes théoriques et numériques ont été largement étudiées et documentées dans la littérature, l'état de l'art des méthodes automatiques, en particulier celles s'appuyant sur les réseaux de neurones et les algorithmes d'optimisation, est nettement moins fourni. Cela suggère une opportunité, mais aussi une grande marge de progression dans cette direction. Les avancées rapides et les succès sans précédent de l'intelligence artificielle dans divers domaines renforcent l'idée que ces techniques

pourraient être utilisées de manière innovante pour aborder la complexité de la détermination des fonctions de Lyapunov.

Dans ce contexte, il est essentiel d'explorer les méthodes automatiques comme une suite logique de la progression dans la recherche sur la détermination des fonctions de Lyapunov.

2.4.1 Réseaux de neurones

Les réseaux de neurones artificiels ([Artificial Neural Network \(ANN\)](#)) sont des approximateurs universels inspirés de la manière dont les neurones biologiques traitent l'information dans le cerveau humain. Ces modèles ont été introduits pour la première fois dans les années 1940, mais ce n'est que récemment, avec l'avènement des puissantes capacités de calcul et des grandes quantités de données, qu'ils ont véritablement montré leur potentiel.

Un ANN est constitué d'unités, appelées neurones, organisées en couches. Ces neurones sont inter-connectés par des poids, qui sont ajustés lors d'un apprentissage. L'information est introduite dans le réseau via une couche d'entrée, traitée successivement à travers une ou plusieurs couches cachées, et finalement transmise à une couche de sortie. La [Figure 2.1](#) illustre l'architecture typique d'un réseau de neurones avec une couche d'entrée, deux couches cachées et une couche de sortie.

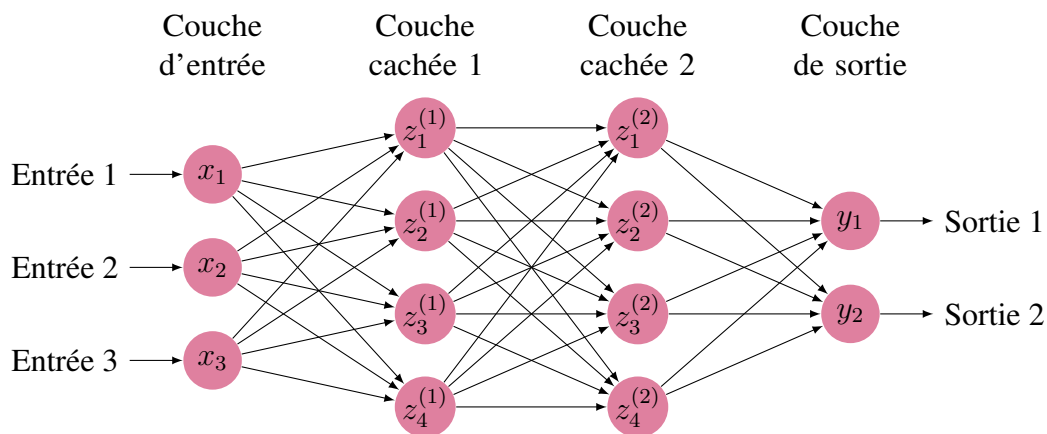


FIGURE 2.1 – Architecture d'un réseau de neurones artificiels

La puissance des ANN réside dans leur capacité à apprendre des représentations complexes à partir de données. Par exemple, dans le domaine de la reconnaissance d'images, un ANN peut apprendre à identifier des caractéristiques telles que les bords, les textures et les formes à partir de pixels bruts, et à combiner ces caractéristiques pour reconnaître des objets ou des scènes.

Cependant, les applications des ANN ne se réduisent pas à la reconnaissance d'images ou de voix. Récemment, des recherches ont exploré l'utilisation des ANN pour des tâches plus avancées et théoriques. Par exemple, la détermination des fonctions de Lyapunov à l'aide de réseaux de neurones est un domaine en plein essor. Les fonctions de Lyapunov sont essentielles pour analyser la stabilité des systèmes dynamiques. En utilisant des ANN pour approximer ces fonctions, il est possible d'obtenir des idées précieuses sur la dynamique de systèmes complexes.

Les réseaux de neurones ([Neural Network \(NN\)](#)) sont couramment reconnus comme des approximations universelles et peuvent être efficacement utilisés pour gérer la modélisation des incertitudes à des fins de contrôle robuste et des applications d'identification et de contrôle de systèmes ([J. LIU, 2013](#)). La modélisation d'un réseau de neurones a été proposée pour la première fois par McCulloch

et Pitts au milieu des années 1940 (McCulloch et Pitts, 1943). Les premiers modèles dynamiques de réseaux de neurones ont été développés au début des années 80, basés sur le travail de David Rumelhart (DE, 1986).

Avant de plonger dans un état de l'art détaillé des divers articles et travaux présentant les techniques pour déterminer une fonction de Lyapunov grâce aux réseaux de neurones, il est essentiel de poser les bases de ces techniques. Pour cela, nous présentons les principes élémentaires des réseaux de neurones. Cet aperçu des concepts élémentaires des NN a pour but d'introduire des structures mathématiques qui sont régulièrement employées dans l'élaboration d'un schéma d'optimisation pour la détermination automatique d'une fonction de Lyapunov, sujet central du chapitre suivant.

2.4.1.1 Principes fondamentaux des réseaux de neurones

a) Formulation du problème :

Considérons un système dynamique représenté par l'équation suivante :

$$\dot{x} = f(x)$$

où x représente l'état du système et f une fonction non linéaire définissant la dynamique du système. L'objectif principal est d'analyser la stabilité du système en trouvant une fonction appropriée qui respecte certaines conditions nécessaires.

b) Architecture du réseau :

L'idée de cette étape est de construire un réseau neuronal. Il existe plusieurs types d'architectures, adaptés à différents types de tâches. Pour approfondir ce sujet, nous pouvons nous référer à l'ouvrage (Hagan et al., 1997), souvent considéré comme la référence en matière de réseaux de neurones.

Réseaux feed-forward (Feed-Forward Neural Network (FFNN)) : Les réseaux feed-forward, dans leur simplicité, consistent en un flux d'informations qui se déplace uniquement de l'entrée vers la sortie, sans aucune boucle. Par conséquent, chaque neurone reçoit une entrée, la traite et transmet le résultat au neurone suivant. Ils représentent l'une des architectures les plus anciennes et les plus directes dans le domaine des réseaux de neurones.

Réseaux de neurones récurrents (Recurrent Neural Network (RNN)) : À l'opposé des architectures feed-forward, les réseaux de neurones récurrents sont équipés de boucles internes pour conserver l'information. Cela leur permet d'avoir une sorte de "mémoire" interne, ce qui les rend particulièrement efficaces pour traiter des séquences de données, où la sortie à un moment donné dépend des entrées précédentes.

Réseaux de neurones convolutifs (Convolutional Neural Network (CNN)) : Conçus spécifiquement pour traiter des données ayant une structure topologique, comme une image, les réseaux de neurones convolutifs se distinguent par leur utilisation d'opérations de convolution. Ces opérations filtrent les entrées à travers des couches successives pour extraire progressivement des caractéristiques pertinentes, rendant les CNN puissants pour des tâches telles que la reconnaissance d'images.

Réseaux de neurones profonds (Deep Neural Network (DNN)) : Comme leur nom l'indique, les réseaux de neurones profonds sont caractérisés par la profondeur de leur architecture, avec plusieurs couches cachées intermédiaires. Cette profondeur permet au réseau de traiter des fonctions hautement complexes, ce qui en fait une option puissante pour de nombreuses applications exigeant une granularité et une complexité élevées.

Réseaux antagonistes génératifs (Generative Adversarial Network (GAN)) : Les GAN sont en réalité constitués de deux réseaux de neurones distincts qui sont en compétition l'un avec l'autre. Le premier, appelé générateur, tente de produire de faux échantillons de données. Le second, le discriminateur, évalue ces échantillons. Au fil de l'apprentissage, le générateur devient de plus en plus doué pour créer des données qui ressemblent à de vraies données, tandis que le discriminateur s'affine pour distinguer les vraies des fausses. Cette dynamique collaborative rend les GAN idéaux pour générer de nouvelles données qui ressemblent étroitement à un ensemble de données d'entrée.

c) Apprentissage du réseau :

L'une des étapes importantes dans l'utilisation des réseaux de neurones est l'apprentissage du réseau. Cette phase consiste à ajuster les paramètres du réseau (souvent appelés poids) de manière à ce qu'ils puissent effectuer l'approximation souhaitée le plus fidèlement possible. L'apprentissage est guidé par une fonction de coût, qui évalue la performance du réseau par rapport à l'objectif fixé. En utilisant des techniques d'optimisation, le réseau ajuste progressivement ses poids pour minimiser cette fonction de coût, c'est-à-dire pour se rapprocher le plus possible de l'approximation idéale. Dans ce contexte, deux méthodes d'optimisation courantes, l'algorithme du gradient stochastique (**Stochastic Gradient Descent (SGD)**) et Adam (**Adaptive Moment Estimation (ADAM)**), sont présentées pour illustrer cette optimisation.

Fonction de coût : La fonction de coût est essentielle pour évaluer à quel point les prédictions du modèle s'écartent des vraies valeurs. Le but de cette fonction est de guider le processus d'optimisation en ajustant les paramètres du réseau. Elle doit être dérivable et fournir un moyen clair d'évaluer la performance du réseau. Idéalement, elle est conçue pour pénaliser les erreurs importantes davantage que les erreurs mineures et guider le modèle vers la meilleure approximation possible de la fonction de Lyapunov.

Entraînement : Il est nécessaire d'utiliser une méthode d'optimisation telle qu'Adam ou SGD pour entraîner le réseau en minimisant la fonction de coût choisie.

- **SGD :** Le SGD (**BOTTOU, 2010 ; BOTTOU et al., 2018**) est la version stochastique de la méthode de descente de gradient. Au lieu d'utiliser tout l'ensemble de données pour calculer le gradient, le SGD utilise un sous-ensemble (souvent appelé batch) pour estimer le gradient. L'idée derrière le SGD est que chaque mise à jour du gradient, même basée sur un seul échantillon ou un petit batch, peut pousser le modèle dans la bonne direction, même quand on le choisit aléatoirement, d'où le terme stochastique.

Mise à jour de SGD :

$$w_{t+1} = w_t - \eta \nabla J(w_t)$$

où :

- w_t est le vecteur des poids et des biais à l'itération t .
 - η est le taux d'apprentissage.
 - $\nabla J(w_t)$ est le gradient de la fonction de coût J évalué au point w_t .
- **Adam :** Adam (**KINGMA et BA, 2014**) est un algorithme d'optimisation qui combine les idées de deux autres extensions de SGD : Momentum et RMSProp.

Mise à jour de Adam :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

où :

- g_t est le gradient à l'itération t .
- m_t est une estimation du premier moment (moyenne) du gradient.
- v_t est une estimation du second moment (variance non centrée) du gradient.
- β_1 et β_2 sont des hyperparamètres pour le contrôle de la décroissance exponentielle des moments (généralement proches de 1).

Correction des estimations des moments pour l'initiation biaisée :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2}.$$

Mise à jour des poids :

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

où ϵ est un terme de lissage pour éviter la division par zéro (généralement une petite valeur proche de zéro).

d) Validation de la fonction approximée :

Une fois l'apprentissage terminée, le réseau de neurones fournit une approximation de la fonction qui aide à évaluer la stabilité du système $\dot{x} = f(x)$. Cependant, pour être utilisable dans un contexte d'ingénierie, cette approximation doit être validée. Cette étape est essentielle pour garantir la sécurité et la fiabilité des systèmes contrôlés en utilisant cette approximation.

2.4.1.2 État de l'art sur les fonctions de Lyapunov neuronales

Bien que l'intérêt croissant pour l'interface entre les réseaux de neurones et les fonctions de Lyapunov soit manifeste dans la littérature récente (CHANG et al., 2019; Y. CHEN et al., 2018; S. LI et al., 2017; S. LI et al., 2016; Z. LI et al., 2014; NIU et al., 2019; T. WANG et al., 2015; X.-M. WANG et al., 2018; WEN et al., 2015; B. ZHANG et al., 2014; G. ZHANG et al., 2020), la majorité de ces travaux s'oriente vers le contrôle de systèmes ou d'autres applications connexes. En revanche, une investigation approfondie concernant l'utilisation des réseaux de neurones, pour la détermination d'une fonction de Lyapunov et l'établissement d'une région de stabilité associée, demeure encore limitée. C'est dans cette direction spécifique que notre sujet de recherche se positionne.

Il y a eu des tentatives antérieures pour utiliser des réseaux de neurones afin d'approximer les fonctions de Lyapunov. Parmi les premiers à intégrer les réseaux de neurones pour la théorie des systèmes dynamiques, l'article (PROKHOROV, 1994) introduit la "Lyapunov Machine", un concept novateur pour l'approximation des fonctions de Lyapunov. Ce travail propose une direction avant-gardiste pour les recherches futures dans ce domaine. En particulier, les travaux dans l'article se

distinguent par la description de la "Lyapunov Machine", un réseau de neurones spécialement conçu pour l'approximation des fonctions de Lyapunov. Toutefois, l'auteur reconnaît que cette méthode présente une complexité algorithmique, en plus de n'offrir qu'une approximation des fonctions de Lyapunov, laissant entrevoir des défis importants à résoudre dans les études futures. Cet article marque donc une étape significative, ouvrant la voie à une exploration plus approfondie de l'intégration des réseaux de neurones dans la résolution des problèmes complexes de la théorie du contrôle et de la stabilité des systèmes dynamiques.

L'article (SERPEN, 2005) propose un algorithme d'apprentissage basé sur des incréments plutôt que sur des dérivées. Cette méthode met à jour les paramètres ou les valeurs par de petites augmentations ou modifications à chaque étape. Cet algorithme s'appuie sur des mises à jour successives des paramètres du réseau plutôt que sur un algorithme d'apprentissage standard. Bien que l'ensemble du schéma de calcul proposé dans cet article présente une idée initiale intéressante pour explorer des approches similaires, en l'absence d'arguments solides sur le plan mathématique et d'exemples concrets démontrant les performances de l'approche, il convient de rester prudent quant à l'adoption de cette méthode pour des applications pratiques dans le contexte de notre thèse. Une étude de simulation visant à établir la validité des propositions théoriques est en cours. Dans (PETRIDIS et PETRIDIS, 2006), seule une fonction de Lyapunov locale est calculée en utilisant des informations dérivées locales dans l'algorithme d'apprentissage. Dans l'article (NOROOZI et al., 2008), les coefficients d'une fonction de Lyapunov polynomiale sont calculés, plutôt que de représenter directement la fonction de Lyapunov par un réseau de neurones. Dans cet article, en plus des techniques d'approximation traditionnelles, les méthodes de propagation avant (Forward Method) et de rétro-propagation (Backward Method) ont été mises en avant. La propagation avant sert à évaluer le réseau sur une entrée donnée, tandis que la rétro-propagation ajuste les poids du réseau pour minimiser l'erreur entre la sortie du réseau et une sortie désirée. Dans (ABATE et al., 2020), une nouvelle méthode de synthèse de fonctions de Lyapunov pour les systèmes non linéaires autonomes a été proposée. Cette approche combine l'apprentissage automatique pour identifier de manière autonome les fonctions de Lyapunov tout en garantissant leur validité formelle. Concrètement, un apprenant numérique et un vérificateur symbolique collaborent pour élaborer des réseaux de neurones de Lyapunov dont la validité est prouvée. Cependant, bien que cette méthode présente des avantages notables en termes d'efficacité et de généralité, elle présente une limite majeure : elle est spécifiquement adaptée aux systèmes avec des dynamiques polynomiales.

2.4.2 Entraînement par métaheuristiques

L'optimisation, avec sa simplicité conceptuelle couplée à sa complexité algorithmique, a façonné de nombreux domaines scientifiques et ingénieriques. Depuis les travaux d'Euler au 18^{ème} siècle jusqu'aux avancées informatiques du 20^{ème} siècle, l'optimisation a constamment évolué et s'est développée. Aujourd'hui, que ce soit pour maximiser un profit, pour minimiser une dépense énergétique ou pour calibrer un modèle complexe, l'optimisation joue un rôle incontournable.

L'optimisation est au cœur de nombreuses décisions scientifiques et industrielles. À chaque étape de notre histoire technologique, nous avons cherché à affiner et à améliorer nos méthodes, à chercher le meilleur résultat possible. Cependant, chaque problème d'optimisation a ses singularités. Certains sont bien définis et relativement simples à résoudre, tandis que d'autres présentent des terrains plus ardues, marqués par de multiples optima locaux et une vaste étendue de solutions potentielles. Et le meilleur, ou l'optimal, peut souvent être hors de portée. Dans des environnements complexes et face à des problèmes d'une grande envergure, la recherche d'une solution parfaite peut s'avérer dure à trouver, voire impossible.

Dans ce contexte, les métaheuristiques se distinguent. Ces techniques offrent des solutions puissantes et efficaces, même lorsque nous sommes confrontés à des paysages d'optimisation complexes. Ce sont des techniques approximatives qui, bien qu'elles ne garantissent pas la solution optimale à chaque fois, ont le potentiel de fournir des solutions de qualité et souvent suffisantes pour la pratique. Cependant, le principal compromis réside souvent dans le temps de calcul nécessaire, qui, selon la complexité du problème et la méthode implémentée, peut s'avérer conséquent. La montée en popularité et l'amélioration continue des métaheuristiques ont conduit à une abondance de recherches et de publications dédiées à ce sujet, comme (EIBEN et SMITH, 2015 ; TALBI, 2009). Ces références fournissent une base solide pour comprendre, concevoir et mettre en œuvre des métaheuristiques dans divers contextes d'optimisation, tout en reconnaissant l'importance de gérer de manière judicieuse le temps de calcul.

Face à la multitude de métaheuristiques disponibles, il est essentiel de sélectionner les méthodes d'optimisation les plus adaptées à nos besoins pour déterminer une fonction de Lyapunov. Pour cette thèse, nous avons choisi de nous concentrer sur l'algorithme génétique (Genetic Algorithm (GA)) et l'optimisation par essaim particulaire (Particle Swarm Optimization (PSO)). Plusieurs raisons sous-tendent ce choix. Tout d'abord, ces deux méthodes ont démontré leur efficacité sur une large gamme de problèmes d'optimisation, tant dans des contextes académiques que pratiques. De plus, leur nature adaptative et leur capacité à explorer et exploiter l'espace de recherche les rendent particulièrement appropriées pour traiter des problèmes complexes et non linéaires. L'état de l'art en matière de GA et PSO est également riche et bien fourni, ce qui témoigne de l'importance et de la pertinence de ces techniques dans la communauté scientifique. Enfin, leur flexibilité et leur richesse en termes de paramétrisation offrent une opportunité unique d'adaptation et d'ajustement, permettant d'affiner les solutions en fonction des spécificités de chaque problème. Ainsi, en retenant le GA et le PSO, nous espérons fournir des solutions robustes et efficaces aux problèmes abordés dans cette thèse.

2.4.2.1 Algorithmes génétiques

Un algorithme génétique (GA) est une heuristique de recherche qui est inspirée par la théorie de l'évolution naturelle de Charles Darwin. Cet algorithme reflète le processus de sélection naturelle où l'individu le plus apte est sélectionné pour la reproduction afin de produire la progéniture de la prochaine génération.

En informatique, un algorithme génétique est couramment utilisé pour générer des solutions de haute qualité aux problèmes d'optimisation et de recherche en s'appuyant sur des opérateurs inspirés de la biologie tels que les mutations, les croisements et la sélection. John Holland a introduit l'algorithme génétique, sur la base du concept de la théorie de l'évolution de Darwin, en 1960 (HOLLAND, 1992). Le développement s'est poursuivi grâce aux efforts conjugués de Holland, de ses étudiants et de ses collègues à l'Université du Michigan durant les années 1960 et 1970 (DE JONG, 1975). La Figure 2.2 illustre le déroulement typique d'un algorithme génétique.

En matière d'évolution, la notion de sélection naturelle est centrale. Elle se base sur l'idée que, dans une population, les individus les plus aptes sont sélectionnés pour la reproduction. Ces individus privilégiés donnent naissance à une progéniture qui hérite de leurs caractéristiques distinctives. Si ces traits sont avantageux, la nouvelle génération est généralement mieux adaptée que la précédente, augmentant ainsi ses chances de survie. Ce mécanisme de sélection se répète continuellement, visant à produire l'individu le plus adapté à son environnement. Dans le contexte des algorithmes génétiques, cette idée est transposée à la recherche de solutions optimales. Nous considérons ainsi un ensemble de solutions possibles à un problème donné, et nous privilégions celles qui s'avèrent être les plus efficaces ou pertinentes.

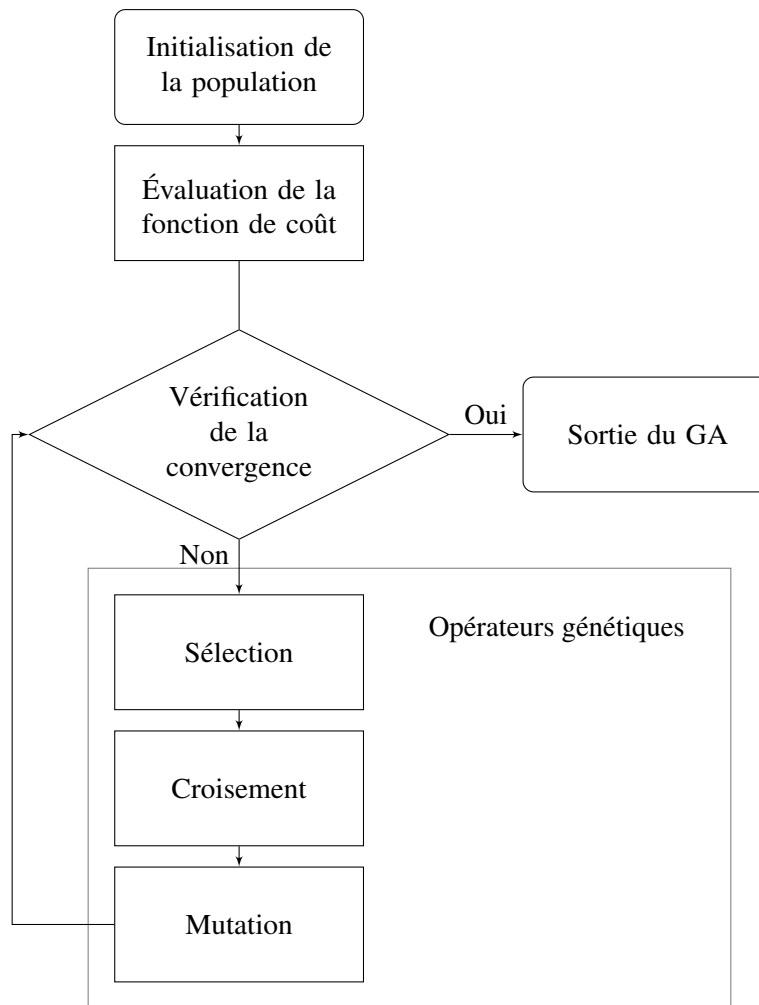


FIGURE 2.2 – Organigramme d'un algorithme génétique typique

Cinq éléments clés sont considérés dans un algorithme génétique :

- **Population initiale** : Le processus commence par un ensemble d'individus que nous appelons une population. Chaque individu est une solution au problème à résoudre. Un individu est caractérisé par un ensemble de paramètres (variables) appelés gènes. Les gènes sont joints en chaînes pour former un chromosome (solution). Dans l'algorithme génétique, l'ensemble des gènes d'un individu est représenté par une chaîne, tel un alphabet. Il est possible, par exemple, d'utiliser des valeurs binaires (chaînes de 1 et de 0). Nous disons que nous codons les gènes dans le chromosome.
- **Fonction de coût** : La fonction de coût détermine à quel point un individu est apte (la capacité d'un individu à rivaliser avec les autres individus). Elle donne le score de "fitness" à chaque individu. La probabilité qu'un individu soit sélectionné pour la reproduction est basée sur son score de "fitness".
- **Sélection** : L'idée de la phase de sélection est de trier les individus les plus aptes et de les laisser transmettre leurs gènes à la prochaine génération. Deux paires d'individus (parents) sont sélectionnés en fonction des scores de "fitness". Les individus ayant un "fitness" élevé ont plus de chances d'être choisis pour la reproduction.

- **Croisement** : Le croisement est une étape fondamentale dans un algorithme génétique, permettant de mélanger et d'intégrer les caractéristiques de deux parents pour former de nouveaux descendants. Il existe de nombreuses méthodes de croisement, chacune ayant ses propres spécificités. L'une des techniques courantes est le croisement en un point. Dans cette approche, un emplacement est choisi aléatoirement sur la chaîne de gènes. Les informations de deux parents sont alors échangées à partir de cet emplacement pour former deux nouveaux descendants. Une autre technique populaire est le croisement uniforme. Contrairement à la méthode précédente qui se base sur un unique point de croisement, le croisement uniforme examine chaque gène individuellement. À chaque position, un gène est sélectionné aléatoirement soit du premier parent, soit du second, en fonction d'une probabilité fixée. Cette méthode produit donc un descendant dont les gènes sont un mélange aléatoire des deux parents. Une fois les descendants créés, que ce soit par ces techniques ou d'autres, ils sont ajoutés à la population existante.
- **Mutation** : Pour certains des nouveaux descendants formés, certains de leurs gènes sont soumis à une mutation avec une faible probabilité aléatoire. Cela signifie que des éléments ou des valeurs de la représentation génétique de l'individu peuvent être modifiés. La mutation se produit pour maintenir la diversité au sein de la population et prévenir une convergence prématurée.

L'algorithme se termine si la population a convergé. Cela signifie que l'algorithme génétique a trouvé une solution qui est probablement proche de l'optimum pour notre problème, sans toutefois garantir qu'il s'agisse de la solution optimale.

2.4.2.2 Optimisation par essaim particulaire

L'algorithme d'optimisation par essaim de particules (PSO) est une technique d'optimisation stochastique basée sur l'essaim, qui a été proposée par (EBERHART et KENNEDY, 1995) et (KENNEDY et EBERHART, 1995). L'algorithme PSO simule le comportement social des animaux comme les insectes, les troupeaux, les oiseaux et les poissons. Ces essais adoptent une façon coopérative de trouver de la nourriture, et chaque membre de l'essaim modifie constamment le modèle de recherche en fonction des expériences d'apprentissage de lui-même et des autres membres. L'organigramme présenté à la Figure 2.3 illustre le déroulement typique d'un algorithme PSO.

L'idée principale de la conception de l'algorithme PSO est étroitement liée à deux recherches : l'algorithme évolutionnaire (Evolutionary Algorithm (EA)) et la vie artificielle.

Le principe de l'algorithme PSO s'inspire directement de la nature, empruntant aux stratégies collectives de déplacement observées chez certains animaux, comme les formations d'oiseaux ou les bancs de poissons. Cette méthode se rapproche des EA par son utilisation d'une population de solutions potentielles, mais elle s'en distingue par son mécanisme de mise à jour des solutions. Plutôt que de dépendre de la mutation ou du croisement comme dans les algorithmes évolutionnaires, chaque particule au sein de l'essaim de PSO ajuste sa position en considérant les meilleures découvertes personnelles ainsi que les réussites partagées au sein de l'essaim.

Dans ce contexte de la vie artificielle, le PSO simule le comportement d'agents autonomes qui s'adaptent et apprennent de leur environnement et de leurs pairs. Ces agents, qui sont les particules individuelles au sein de l'essaim, sont en quête continue de meilleures solutions dans l'espace défini par la fonction objectif, s'appuyant sur leurs expériences et celles des autres pour progresser. Cette coopération entre les découvertes individuelles et la connaissance collective permet à l'essaim de converger vers une solution optimale, rendant le PSO particulièrement efficace.

Millonas a formulé cinq principes essentiels pour développer des systèmes d'essaim artificiels coopératifs, s'appuyant sur l'étude du comportement des animaux sociaux selon la théorie de la vie artificielle (VAN DEN BERGH et al., 2007) :

- **Proximité** : L'essaim doit être capable d'effectuer des calculs spatiaux et temporels simples.
- **Qualité** : L'essaim doit être capable de détecter le changement de qualité dans l'environnement et d'y répondre.
- **Réponse diverse** : L'essaim ne doit pas limiter son moyen d'obtenir les ressources dans une portée étroite.
- **Stabilité** : L'essaim ne doit pas changer son mode de comportement à chaque changement environnemental.
- **Adaptabilité** : L'essaim doit changer son mode de comportement lorsque ce changement est significatif.

Pour une compréhension approfondie de l'algorithme d'optimisation par essaim particulaire, une stratégie d'optimisation inspirée du monde naturel, la consultation de l'étude réalisée dans (D. WANG et al., 2018) est recommandée.

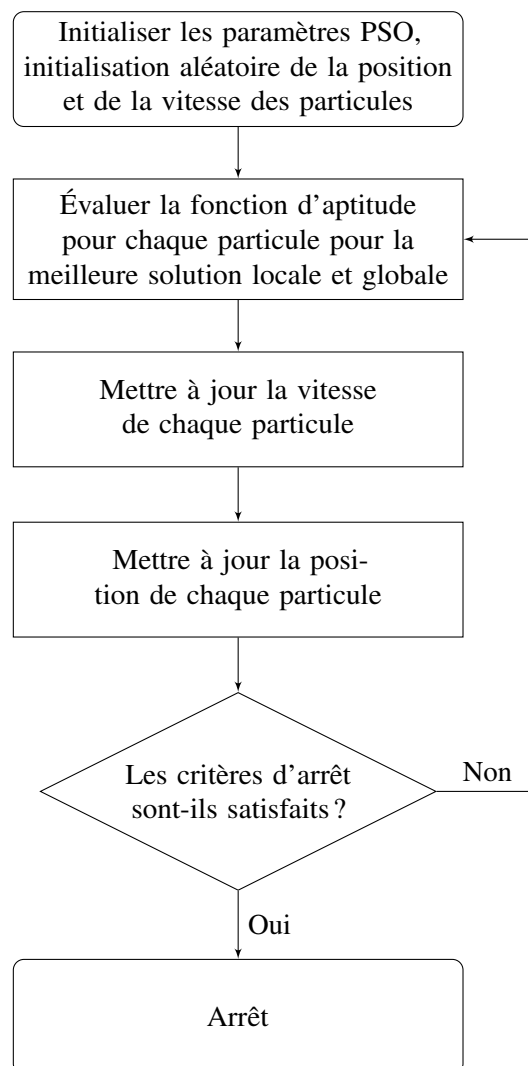


FIGURE 2.3 – Organigramme de l'optimisation par essaim de particules

2.4.2.3 État de l'art de l'optimisation par métaheuristiques

L'étude des fonctions de Lyapunov et leur rôle dans la commande et la stabilité des systèmes a suscité l'intérêt de la communauté de recherche. La détermination de ces fonctions à l'aide d'algorithmes d'optimisation est un champ d'étude en plein essor. Dans cette section, nous passons en revue les principales contributions et les techniques utilisées pour aborder ce défi.

Jusqu'à présent, il y a peu de littérature sur l'application des algorithmes d'optimisation pour déterminer une fonction de Lyapunov. Dans la plupart des cas, les auteurs se sont concentrés sur l'identification du système ou la conception du contrôleur en recherchant une loi de commande.

Dans (SABOURI et al., 2020), les auteurs se concentrent sur l'emploi d'un algorithme génétique pour élaborer une fonction de Lyapunov. Pour cela, une structure particulière de la fonction de Lyapunov est définie, prenant en compte les valeurs indéfinies des paramètres de cette fonction et le modèle dynamique du système analysé. La structure définie de la fonction de Lyapunov sert de point de départ pour le processus de l'algorithme génétique. L'article considère deux architectures principales de fonctions de Lyapunov avec des paramètres non déterminés au préalable comme candidats potentiels : une forme générale de la fonction de Lyapunov et une forme polynomiale de celle-ci.

La programmation génétique (Genetic Programming (GP)), proposée par Koza (KOZA, 1994), est un algorithme d'optimisation basé sur les principes de l'évolution naturelle et est capable de résoudre des problèmes d'optimisation en commande, robotique, régression symbolique, etc. (GROSMAN et LEWIN, 2005 ; SEARSON et al., 2010). Contrairement à de nombreux algorithmes évolutionnaires, tel que le GA, les agents dans le GP sont représentés sous forme de structures en forme d'arbre. En d'autres termes, chaque arbre composé de branches et de nœuds, comme le montre la Figure 2.4, représente une solution prospective au problème d'optimisation. Chaque nœud peut être défini comme un terminal ou une fonction. Les terminaux peuvent être soit une valeur constante, soit un nombre aléatoire, tandis que les fonctions peuvent être des opérateurs booléens ou arithmétiques. En particulier, il a été démontré que le GP peut être utilisé pour construire des fonctions candidates de Lyapunov en codant un ensemble de fonctions de base dans une structure arborescente. Une optimisation est ensuite effectuée sur le domaine d'attraction sous réserve des contraintes liées aux caractéristiques de la fonction de Lyapunov.

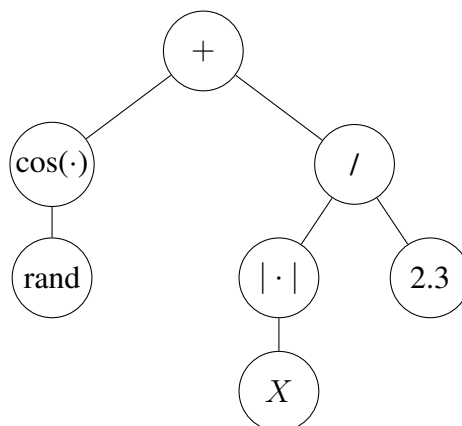


FIGURE 2.4 – Agent sous forme d'arbre représentant $\left[\cos(\text{rand}) + \frac{|X|}{2.3} \right]$

La majorité des publications relatives aux EAs se focalisent sur la conception et la synthèse de contrôleurs. Ces méthodes avancées sont exploitées pour développer des lois de contrôle pour des

systèmes dynamiques variés, comme dans (KOZA et al., 2000; REICHENSDÖRFER et al., 2017; SHIMOOKA et FUJIMOTO, 2000). Cependant, il existe également des travaux sur l'utilisation des EA pour l'optimisation structurelle des fonctions de Lyapunov.

Certains travaux de recherche se sont spécifiquement attachés à l'emploi du GP pour l'élaboration de fonctions de Lyapunov. Comme illustré dans (BANKS, 2002; GROSMAN et LEWIN, 2005), ces études appliquent le GP pour générer des fonctions de Lyapunov potentielles, lesquelles sont évaluées et sélectionnées en fonction de leur performance. Chaque fonction candidate est notée selon la taille de la sous-région d'attraction qu'elle délimite. La probabilité qu'une fonction de Lyapunov soit retenue pour la prochaine itération du processus évolutif dépend directement de cette note. Pour affiner leur méthode, les auteurs emploient des fonctions de coût distinctes afin d'optimiser la taille de cette sous-région d'attraction, s'efforçant ainsi d'identifier la fonction de Lyapunov la plus performante pour le système en question.

Les inconvénients fondamentaux associés à l'utilisation du GP pour la construction de candidats de Lyapunov sont que l'expression résultante pour la fonction de Lyapunov produite par le GP peut être extrêmement grande (ce phénomène est appelé "expression bloating"). Par conséquent, la convergence vers une solution peut être lente et / ou inexistante (MCGOUGH et al., 2010).

Dans (MCGOUGH et al., 2010), afin d'améliorer les performances et l'utilisabilité démontrées par l'approche basée sur le GP, les auteurs proposent d'utiliser l'évolution grammaticale (Grammatical Evolution (GE)). Bien que le GP et le GE soient tous deux des formes d'algorithmes évolutionnaires, la manière dont ils représentent ou codent la solution est en réalité assez différente (MCGOUGH et al., 2010). Il a été démontré que pour certaines applications, le GE est capable de surpasser le GP en termes de vitesse de convergence, d'exigence de stockage et de structure de solution (O'NEILL et al., 2001). Plus précisément, le GE ne souffre généralement pas de "expression bloating" comme le fait le GP, améliorant ainsi considérablement le taux de convergence et réduisant les coûts de stockage. De plus, les solutions analytiques résultantes sont souvent plus petites et nettement moins compliquées.

Dans (ALOUÏ et al., 2021), l'article vise à développer un algorithme numérique original pour construire des fonctions de Lyapunov polynomiales et maximiser le domaine d'attraction. Les auteurs améliorent la technique adoptée dans (NAJAFI et al., 2016) en combinant deux outils heuristiques puissants : l'algorithme d'acceptation de seuil (Threshold Algorithm (TA)) et le PSO afin de maximiser l'estimation du DOA.

2.4.3 Domaine de stabilité garanti

L'estimation du domaine de stabilité garanti est cruciale dans l'analyse des systèmes non linéaires, particulièrement dans un contexte industriel où la sécurité est prioritaire. Ce domaine de stabilité garanti peut être conceptualisé comme une estimation du DOA, à l'intérieur duquel tout point initial est assuré de converger vers un état stable. Ce domaine varie en fonction de la nature de la stabilité étudiée (au sens de Lyapunov, asymptotique, exponentielle, ou entrée - état).

Dans la littérature scientifique existante, l'analyse de stabilité des systèmes dynamiques se concentre fréquemment sur la caractérisation de la stabilité locale autour d'un point d'équilibre. Cependant, cette approche ne tient pas systématiquement compte de la taille du domaine de stabilité garanti, un sous-ensemble du DOA, qui est essentiel pour nos applications pratiques. Dans cette sous-section, nous énumérons et présentons succinctement certains travaux de recherche significatifs qui explorent cette problématique. Ces études sont d'une grande importance pour notre approche, car elles fournissent une base sur laquelle nous pouvons travailler afin de développer une méthode visant à obtenir une estimation maximale du domaine de stabilité garanti avec nos ambitions scientifiques et industrielles.

Dans la contribution de (VANNELLI et VIDYASAGAR, 1985), les auteurs ont introduit les fonctions de Lyapunov, dites maximales, qui sont des fonctions de type rationnel plutôt que des polynômes. Dans de telles fonctions, le dénominateur a un effet "d'explosion" près des limites du DOA, de sorte que leurs level sets représentent étroitement la région de stabilité dans cette partie de l'espace des états. Dans cet article, les auteurs ont proposé un algorithme récursif pour construire simultanément la fonction maximale et calculer l'estimation de ce domaine de stabilité. Dans (CHESI et al., 2003), les auteurs ont utilisé les inégalités matricielles linéaires (Linear Matrix Inequality (LMI)) pour aborder l'estimation du DOA d'un système dynamique basé sur les level sets des fonctions de Lyapunov. Une approche présentée dans (CHESI, 2007b) vise à approximer le DOA comme l'union d'un nombre infini de level sets de fonctions de Lyapunov, plutôt que d'un seul. Pour identifier ces level sets, l'article propose de résoudre une série de problèmes d'optimisation basés sur des LMI. Dans (CHESI, 2009), une stratégie pour estimer les DOAs des systèmes non polynomiaux a été proposée. L'idée est de convertir le problème d'optimisation non polynomial en un problème polynomial, qui peut être abordé via des techniques LMI convexes. Les termes non polynomiaux sont approximés par des séries de Taylor tronquées.

Des formulations, qui font usage de résultats sur l'optimisation globale déterministe basée sur la théorie des moments, ont récemment été proposées (HACHICHO, 2007 ; HACHICHO et TIBKEN, 2002). La technique permet l'identification du meilleur level set possible (le plus grand level set donc) d'une fonction de Lyapunov rationnelle qui constitue une estimation du DOA pour les systèmes dynamiques non linéaires de type polynomial.

Dans la section 1.5.6, nous avons abordé l'approche de maximisation du domaine de stabilité garanti à travers une méthode mise en lumière par (NAJAFI et al., 2016). Cette méthode est d'une importance capitale pour notre recherche, en raison de sa capacité à déterminer efficacement la valeur optimale de c . Ce faisant, elle permet de maximiser le level set d'une fonction de Lyapunov candidate, ce qui est déterminant pour assurer le plus large domaine de stabilité garanti possible.

Dans les travaux présentés dans (NAJAFI et al., 2016), l'approche d'échantillonnage est appliquée pour identifier le plus grand level set d'une fonction de Lyapunov candidate, garantissant ainsi la stabilité d'un système non linéaire. Cette méthode implique de sélectionner des points de manière systématique dans l'espace d'état pour vérifier la condition de dérivée temporelle négative de la fonction de Lyapunov. L'objectif est d'avoir le plus vaste ensemble possible où cette condition est satisfaite, assurant que toute trajectoire issue de cet ensemble aboutisse au point d'équilibre. Cette stratégie est particulièrement utile pour obtenir des estimations précises et rapides du domaine de stabilité garanti.

2.4.4 Fonction de Lyapunov fondée sur les réseaux de neurones

Alors que les réseaux de neurones et les algorithmes génétiques ont chacun laissé une marque profonde dans le monde de la recherche en intelligence artificielle, un travail récent (PETRIDIS et PETRIDIS, 2006) a dévoilé le potentiel remarquable de leur combinaison. Cette fusion novatrice a mené à une approche intéressante et prometteuse pour la construction de fonctions de Lyapunov modélisées par des réseaux de neurones. Afin de définir précisément cette fonction, il devient essentiel de calculer les valeurs des poids en respectant certaines conditions spécifiques. Les auteurs ont mis en place une approche singulière, en tenant compte des paramètres propres au système et des caractéristiques d'une fonction de Lyapunov, pour ajuster précisément les poids du réseau neuronal afin d'obtenir une fonction de Lyapunov. Ensuite, grâce à un algorithme d'optimisation, cette fonction de Lyapunov est déterminée.

Dans cette sous-section, nous décortiquons les innovations majeures et les limites de cet article et soulignons comment cette recherche sert de pierre angulaire à cette thèse.

Dans cet article, selon les auteurs, quatre conditions sont "suffisantes" pour que le point 0 d'un système autonome soit asymptotiquement stable :

(a1) $V(0) = 0$,

(a2) $V(x) > 0$ pour tous les x non nuls dans un voisinage de 0,

(a3) $\dot{V}(0) = 0$,

(a4) $\dot{V}(x) < 0$ pour tous les x non nuls dans un voisinage de 0,

où V est la fonction de Lyapunov recherchée et $\dot{V}(x)$ sa dérivée.

Ces conditions peuvent être traduites sur des caractéristiques de la fonction de Lyapunov, à savoir l'existence d'un minimum local en zéro et d'un maximum local de la dérivée \dot{V} en zéro. Ces contraintes sur la fonction de Lyapunov peuvent être vérifiées à partir de la première et deuxième dérivées de V et \dot{V} par rapport à x . Dans le cas de la deuxième dérivée, l'existence d'un minimum ou d'un maximum local est vérifiée à partir des valeurs propres des matrices obtenues lors de ces dérivations.

L'une des innovations majeures de cette étude est la représentation de la fonction de Lyapunov, $V(x)$, à travers un réseau de neurones. Cette approche offre non seulement une flexibilité sans précédent mais capitalise également sur la capacité d'approximation universelle des réseaux de neurones. Ces derniers peuvent modéliser des fonctions d'une complexité significative, permettant une analyse plus approfondie et précise des systèmes dynamiques.

De ce fait, les matrices des dérivées secondes, que ce soit pour V ou pour \dot{V} , sont calculées en tant que fonctions des poids du réseau de neurones. L'ajustement de ces poids est donc d'une importance capitale. Dans cette optique, les auteurs ont employé un algorithme génétique, cherchant à optimiser ces poids de manière efficace. Leur ambition principale est d'assurer que la matrice des dérivées secondes de \dot{V} soit définie négativement. La détermination d'une fonction de Lyapunov adéquate est ancrée dans cette démarche d'optimisation. La sélection de la fonction de coût n'est pas anodine : elle doit respecter les critères préalablement établis, mais elle est également conçue pour orienter l'algorithme génétique dans sa recherche d'une fonction de Lyapunov conforme aux caractéristiques essentielles des systèmes dynamiques.

Bien que prometteuse, cette approche présente plusieurs limites qui méritent une attention particulière. L'utilisation de la formulation "pour tous les x non nuls dans un voisinage de 0" pour décrire les conditions nécessaires à la détermination d'une fonction de Lyapunov pour un système donné présente une ambiguïté. Le terme "voisinage" de 0 demeure en effet assez vague et n'offre aucune indication précise sur le domaine de stabilité garanti associé, ce qui rend la détermination de la fonction de Lyapunov moins précise qu'elle ne pourrait l'être.

De plus, l'utilisation de fonctions barrières, qui évoquent les méthodes de pénalisation en optimisation, est un autre aspect critique. Si ces fonctions peuvent aider à gérer les contraintes pendant l'optimisation, elles ne garantissent pas nécessairement que le résultat final soit véritablement une fonction de Lyapunov, nécessitant ainsi une vérification a posteriori.

Enfin, la méthode pourrait ne pas être adaptée de manière universelle à une vaste classe de systèmes.

2.5 Synthèse des méthodes

La détermination des fonctions de Lyapunov a toujours été un sujet d'intérêt majeur dans le domaine de la recherche, en raison de son implication fondamentale dans la compréhension et l'analyse de la stabilité des systèmes. Dans cette revue de la littérature existante, nous avons abordé une diversité de méthodologies, chacune étant le reflet de la progression constante et de l'évolution de ce domaine.

Les méthodes théoriques, allant de la linéarisation à la méthode de Krasovski, et incluant le principe d'invariance de LaSalle, mettent en avant l'importance des fondations mathématiques rigoureuses. Ces méthodes ont été complétées par des approches numériques, telles que le gradient variable, la méthode de Zubov, ou encore la somme des carrés, pour n'en nommer que quelques-unes, qui cherchent à exploiter la puissance de calcul pour résoudre des problèmes qui, sans elle, seraient difficiles à aborder.

La montée en puissance des méthodes automatiques est indéniable, avec une place croissante accordée aux réseaux de neurones et aux algorithmes d'optimisation. Le recours à l'AI, et notamment aux réseaux de neurones, pour déterminer les fonctions de Lyapunov, reflète l'engagement de la communauté scientifique à intégrer les avancées technologiques dans leurs travaux.

Chaque méthode et chaque approche apportent ses contributions, mais aussi ses défis. Certains se sont révélés particulièrement prometteurs, comme la méthode qui utilise conjointement les réseaux de neurones et l'algorithme génétique que nous avons évoquée dans la section précédente. Bien qu'elle présente des limitations spécifiques, telles que l'ambiguïté associée au voisinage de 0 et l'utilisation de fonctions barrières, son approche constitue une base solide et inspirante pour la poursuite de nos travaux.

Toutefois, il est évident que la recherche dans ce domaine reste ouverte. Face aux nombreux défis posés par la détermination des fonctions de Lyapunov, il est impératif d'aller au-delà de la méthodologie existante et d'explorer de nouvelles perspectives. C'est précisément l'objectif de la section suivante, où nous introduisons notre méthode innovante, qui non seulement s'appuie sur les forces des travaux antérieurs, mais vise également à pallier leurs faiblesses.

Recherche automatique d'une fonction de Lyapunov par optimisation

Recherche automatique d'une fonction de Lyapunov par optimisation

Sommaire

3.1	Introduction	78
3.2	Contexte	79
3.2.1	Expression de H^V	80
3.2.2	Expression de H^G	80
3.3	Cas typiques	82
3.3.1	Forme quadratique	82
3.3.2	Réseau de neurones	83
3.4	Mise en œuvre de l'optimisation	87
3.4.1	Cadre d'optimisation	87
3.4.2	Définition du maillage	87
3.4.3	Estimation d'un DOA	88
3.4.3.1	Méthode 1 : Estimation indirecte d'un DOA	89
3.4.3.2	Méthode 2 : Estimation directe du DOA avec des level sets	90
3.4.4	Fonction de coût	90
3.4.4.1	Implémentation sous contraintes	91
3.4.4.2	Implémentation sans contrainte externe	94
3.4.5	Robustesse au maillage	95
3.4.5.1	Constante de Lipschitz	97
3.5	Différentes formes de stabilité	99
3.5.1	Stabilité exponentielle	99
3.5.1.1	Rappels théoriques	99
3.5.1.2	Mise en œuvre de la méthode	99
3.5.2	Stabilité entrée - état	101
3.5.2.1	Rappels théoriques	101
3.5.2.2	Mise en œuvre de la méthode	101
3.6	Extension à d'autres problèmes	103
3.6.1	Systèmes à temps discret	103
3.6.1.1	Théorie	103
3.6.1.2	Mise en œuvre de la méthode	103
3.6.1.3	Application à un réseau de neurones	105

3.6.2	Instabilité	109
3.6.2.1	Rappels théoriques	109
3.6.2.2	Mise en œuvre de la méthode	109
3.7	Méthode générique	111
3.7.1	Rappels théoriques	111
3.7.2	Expression de H^V	112
3.7.3	Expression de H^G	112
3.7.3.1	Cas du temps continu	112
3.7.3.2	Cas du temps discret	113
3.7.4	Application aux différents problèmes de stabilité	114
3.8	Récapitulatif et perspectives	115

3.1 Introduction

Face aux méthodes existantes dévoilées par la littérature actuelle, notre étude a révélé plusieurs défis significatifs. Premièrement, il est manifeste que l'approche conservatrice de certaines stratégies restreint leur portée, les rendant inadéquates pour une application élargie. Deuxièmement, la complexité inhérente à la mise en œuvre de ces méthodes traditionnelles pose un défi, en particulier pour leur intégration dans des contextes industriels. Il est nécessaire de disposer d'une méthode capable de s'adapter à diverses formes de stabilité et de se déployer aux systèmes qu'ils soient à temps continu ou à temps discret, une flexibilité que les techniques actuelles ne fournissent pas systématiquement.

Poursuivant l'examen des stratégies automatisées évoquées dans le chapitre précédent, notre recherche s'oriente désormais vers une méthode novatrice : l'élaboration d'une fonction de Lyapunov par une optimisation guidée, en utilisant un modèle de réseau neuronal. Bien que notre inspiration provienne de (PETRIDIS et PETRIDIS, 2006), nous avons adapté cette technique pour la rendre générique. Notre proposition souligne une approche visant à définir une fonction de Lyapunov, avec l'objectif de trouver une estimation d'un domaine d'attraction (DOA) la plus grande possible.

La méthode adopte le même formalisme que celui présenté dans l'article (PETRIDIS et PETRIDIS, 2006). Les auteurs proposent une approche pour la construction d'une fonction de Lyapunov qui est modélisée par un réseau neuronal en utilisant la capacité d'approximation universelle bien connue des réseaux neuronaux. Ils emploient ensuite un algorithme génétique, cherchant à optimiser les poids du réseau de neurones de manière efficace grâce à une fonction de coût permettant d'obtenir une approximation précise de la fonction de Lyapunov. Bien qu'efficace, la principale difficulté de cette approche est liée à l'absence d'estimation du DOA et à l'utilisation de fonctions de pénalité, qui ne garantissent pas que le résultat final soit une fonction de Lyapunov : une vérification a posteriori est nécessaire. Afin d'améliorer les performances et la facilité d'utilisation de cette approche, nous proposons d'utiliser un nouveau schéma d'optimisation sous contraintes tout en maximisant le domaine de stabilité garanti inclus dans le DOA.

Notre approche combine un cadre d'optimisation polyvalent capable de gérer divers scénarios de stabilité, de la stabilité au sens de Lyapunov jusqu'à la stabilité entrée - état. Que ce soit par l'emploi de fonctions de Lyapunov basiques, telles que la forme quadratique, ou par l'adoption de structures plus complexes comme celles fondées sur les réseaux de neurones, notre stratégie est conçue pour opérer de manière transversale et efficace. Elle se profile comme une solution adéquate tant pour les systèmes à temps continu que pour ceux à temps discret, offrant ainsi la faculté de résoudre une multitude de problématiques.

3.2 Contexte

Considérons toujours le système autonome (1.9), dans lequel nous supposons que le point d'équilibre, dont nous souhaitons étudier la stabilité, est le point 0 ($x = 0$). Ainsi, nous pouvons écrire :

$$f(0) = 0. \quad (3.1)$$

Supposons que $V(x)$ est une fonction scalaire, continue et différentiable et que sa dérivée par rapport au temps est donnée par :

$$G(x) = \frac{dV}{dt} = \sum_{j=1}^n \frac{\partial V}{\partial x_j} f_j(x) \quad (3.2)$$

avec $x = [x_1, \dots, x_n]^T$.

Ainsi, grâce aux définitions du [Chapitre 1](#), la fonction de Lyapunov recherchée $V(x)$ et sa dérivée $G(x)$ doivent satisfaire deux conditions :

- Dans le DOA, $V(x)$ a un minimum local en 0 et $V(0) = 0$.
- Dans le DOA, $G(x)$ a un maximum local en 0 et $G(0) = 0$.

Les conditions suffisantes pour que $V(x)$ ait un minimum local en 0 sont :

(v1) $V(0) = 0$,

(v2) $\left. \frac{\partial V}{\partial x_j} \right|_{x=0} = 0$ pour tout $j = 1, 2, \dots, n$,

(v3) H^V (la matrice des dérivées secondes de V en $x = 0$) est définie positive.

De la même manière, les conditions suffisantes pour que $G(x)$ ait un maximum local en 0 sont :

(g1) $G(0) = 0$,

(g2) $\left. \frac{\partial G}{\partial x_j} \right|_{x=0} = 0$ pour tout $j = 1, 2, \dots, n$,

(g3) H^G (la matrice des dérivées secondes de G en $x = 0$) est définie négative.

Nous notons $x = [x_1, \dots, x_n]^T$ et la matrice jacobienne de f par :

$$F^J := \{J_{qr}, q = 1, \dots, n \text{ et } r = 1, \dots, n.\} \quad (3.3)$$

où :

$$J_{qr} = \left. \frac{\partial f_q}{\partial x_r} \right|_{x=0} \quad (3.4)$$

Dans la suite, nous supposons que l'utilisateur a choisi une structure pour $V(x)$ qui est deux fois différentiable : forme polynomiale, réseau neuronal (à fonction d'activation deux fois différentiable), forme quadratique, réseau de forme de base radiale, etc.

Bien que notre approche soit bien adaptée aux systèmes différentiables, nous pouvons l'étendre à plusieurs autres classes de systèmes (potentiellement non différentiables) si ils sont préalablement identifiés comme tels. Par exemple, les saturations, les valeurs absolues, les fonctions de signe, etc., peuvent être approximées par des techniques de régression efficaces. Dans la section suivante, une formule générique pour H^V et H^G est définie.

3.2.1 Expression de H^V

La dérivée seconde de V peut être obtenue de manière directe :

$$H^V := \{V_{qr}, q = 1, \dots, n \text{ et } r = 1, \dots, n.\} \quad (3.5)$$

où :

$$V_{qr} = \left. \frac{\partial^2 V}{\partial x_q \partial x_r} \right|_{x=0} \quad (3.6)$$

3.2.2 Expression de H^G

Par définition, nous exprimons $G(x)$ par :

$$G(x) = \frac{dV}{dt} = \sum_{h=1}^n \frac{\partial V}{\partial x_h} f_h(x). \quad (3.7)$$

De ce fait, la dérivée première de G est la suivante, avec $l = 1, \dots, n$:

$$\frac{\partial G}{\partial x_l} = \sum_{h=1}^n \frac{\partial^2 V}{\partial x_h \partial x_l} f_h(x) + \sum_{h=1}^n \frac{\partial V}{\partial x_h} \frac{\partial f_h}{\partial x_l}. \quad (3.8)$$

Si la condition (v2) est satisfaite et en tenant compte de (3.1), la propriété suivante tient :

$$\left. \frac{\partial G}{\partial x_l} \right|_{x=0} = 0. \quad (3.9)$$

La dérivée seconde de G est la suivante :

$$\begin{aligned} \frac{\partial^2 G}{\partial x_l \partial x_p} &= \sum_{h=1}^n \frac{\partial^3 V}{\partial x_h \partial x_l \partial x_p} f_h(x) \\ &+ \sum_{h=1}^n \frac{\partial^2 V}{\partial x_h \partial x_l} \frac{\partial f_h}{\partial x_p} + \sum_{h=1}^n \left(\frac{\partial^2 V}{\partial x_h \partial x_p} \frac{\partial f_h}{\partial x_l} \right) \\ &+ \sum_{h=1}^n \frac{\partial V}{\partial x_h} \frac{\partial^2 f_h}{\partial x_l \partial x_p} \end{aligned} \quad (3.10)$$

avec $l = 1, \dots, n$ et $p = 1, \dots, n$.

Nous désignons par :

$$H^G := \{G_{lp}, l = 1, \dots, n \text{ et } p = 1, \dots, n.\} \quad (3.11)$$

où :

$$G_{lp} = \left. \frac{\partial^2 G}{\partial x_l \partial x_p} \right|_{x=0} \quad (3.12)$$

L'équation (3.10), en tenant compte de (3.1) et si (v2) est satisfaite, résulte en :

$$G_{lp} = \sum_{h=1}^n \left. \frac{\partial^2 V}{\partial x_h \partial x_l} \right|_{x=0} \frac{\partial f_h}{\partial x_p} + \sum_{h=1}^n \left. \frac{\partial^2 V}{\partial x_h \partial x_p} \right|_{x=0} \frac{\partial f_h}{\partial x_l}. \quad (3.13)$$

Sur la base des équations (3.4) et (3.6), nous pouvons écrire :

$$G_{lp} = \sum_{h=1}^n V_{hl} J_{hp} + \sum_{h=1}^n V_{hp} J_{hl}. \quad (3.14)$$

Afin de déterminer une fonction de Lyapunov et grâce aux équations précédentes, nous pouvons réduire les conditions (v1) à (v3) et (g1) à (g3) par ces quatre suivantes :

- (t1) $V(0) = 0$,
- (t2) $G(0) = 0$,
- (t3) H^V est définie positive,
- (t4) H^G est définie négative.

3.3 Cas typiques

La méthodologie décrite précédemment, ainsi que les formulations de H^V et H^G en fonction de X , offrent une flexibilité adaptable à une multitude de systèmes. Nous pouvons initialement explorer une fonction de Lyapunov quadratique, moins coûteuse en termes de calcul. Si le domaine de stabilité garanti obtenu est insuffisant, la recherche peut s'étendre vers des fonctions plus sophistiquées, comme celles fondées sur les réseaux de neurones comme dans (PETRIDIS et PETRIDIS, 2006), ajustant ainsi l'approche en fonction de l'exigence du système et optimisant progressivement la précision de l'analyse.

Afin de mieux saisir la portée et les implications pratiques de cette fonction de Lyapunov, il est utile de l'appliquer à des cas concrets. Cette section vise à illustrer, à travers quelques exemples typiques, la façon dont les conditions (t1) à (t4) peuvent être mises en œuvre pour différentes formes de ces fonctions. Chaque exemple servira non seulement à confirmer la validité de l'approche, mais aussi à mettre en lumière les nuances et les spécificités qu'il convient de prendre en compte lors de son application.

3.3.1 Forme quadratique

Dans cette section, nous considérons la fonction de Lyapunov $V(x)$ représentée par une fonction quadratique. Lorsqu'il s'agit de choisir une fonction de Lyapunov pour l'analyse de la stabilité d'un système, la forme quadratique ne doit pas être considérée comme un choix anodin. En effet, elle est souvent privilégiée car elle exprime bien la notion d'énergie qui s'annule en 0. En outre, lorsqu'une fonction de Lyapunov est quadratique et que sa matrice P est définie positive, cela garantit que la fonction est toujours positive pour tout x non nul, une propriété essentielle pour qu'une fonction de Lyapunov soit valide.

Les avantages des fonctions quadratiques ne s'arrêtent pas là. Leur nature offre des dérivées linéaires, simplifiant grandement l'analyse lorsqu'il faut vérifier si la dérivée temporelle de la fonction est négative. Par ailleurs, les fonctions quadratiques sont convexes (si $P > 0$), ce qui est une propriété désirable localement pour une fonction de Lyapunov. La convexité garantit qu'il n'y a qu'un seul minimum global (pour une forme quadratique définie positive), ce qui est utile pour des méthodes d'optimisation et de vérification.

En parcourant la littérature, il est manifeste que de nombreux papiers scientifiques recourent à diverses techniques pour déterminer une fonction de Lyapunov, lorsque cette dernière est représentée sous une forme quadratique (AMBROSINO et GARONE, 2012; CHESI et al., 2002; DAVISON et KURAK, 1971; EGHBAL et al., 2013; HU et LIN, 2003; JOHANSSON et RANTZER, 1997; JONES et PEET, 2023; PRAJNA et PAPACHRISTODOULOU, 2003; TESI et al., 1996). Néanmoins, bien que leur simplicité soit attrayante, le recours à ces fonctions se fait souvent au prix d'un conservatisme certain quant au résultat de l'analyse.

$V(x)$ est exprimée sous la forme :

$$V(x) = x^T P x. \quad (3.15)$$

En application directe de (3.6), nous obtenons pour $q = 1, \dots, n$ et $r = 1, \dots, n$:

$$V_{qr} = P_{qr} + P_{rq} \quad (3.16)$$

où P_{qr} et P_{rq} représentent les coefficients de la matrice P .

Avec P symétrique, V_{qr} devient :

$$V_{qr} = 2 P_{qr}. \quad (3.17)$$

La dérivée seconde de $G(X)$ est directement donnée par (3.13).

Les contraintes $V(0) = 0$ et $\left. \frac{\partial V}{\partial x_j} \right|_{X=0} = 0$ sont automatiquement remplies.

Calcul détaillé. Nous commençons par le développement de la forme quadratique $V(x)$:

$$V(x) = x^T P x.$$

Cette expression peut aussi être écrite comme une somme double :

$$V(x) = \sum_{i=1}^n \sum_{j=1}^n x_i P_{ij} x_j.$$

Considérons maintenant la dérivée partielle seconde V_{qr} par rapport à x_q et x_r . La dérivée partielle de $V(x)$ par rapport à x_q est :

$$\frac{\partial V}{\partial x_q} = \sum_{j=1}^n P_{qj} x_j + \sum_{i=1}^n P_{iq} x_i.$$

En dérivant une seconde fois par rapport à x_r , nous obtenons :

$$\frac{\partial^2 V}{\partial x_q \partial x_r} = P_{qr} + P_{rq}$$

ce qui permet d'obtenir (3.16) :

$$V_{qr} = P_{qr} + P_{rq}.$$

En utilisant maintenant le fait que la matrice P est symétrique, c'est-à-dire, $P = P_{qr} = P_{rq}$, l'équation (3.16) devient (3.17) :

$$V_{qr} = 2P_{qr}.$$

3.3.2 Réseau de neurones

Dans cette section, nous supposons la fonction de Lyapunov $V(x)$ représentée par un réseau neuronal où x_i sont les entrées, w_{ji} sont les poids de la couche cachée, a_i sont les poids de la couche de sortie, h_i sont les biais de la couche cachée, β est le biais de la couche de sortie ; $i = 1, \dots, n$ et $j = 1, \dots, K$ où K est le nombre de neurones de la couche cachée et σ est la fonction d'activation du réseau de neurones. La fonction de Lyapunov représentée par un réseau de neurones est illustrée par la Figure 3.1.

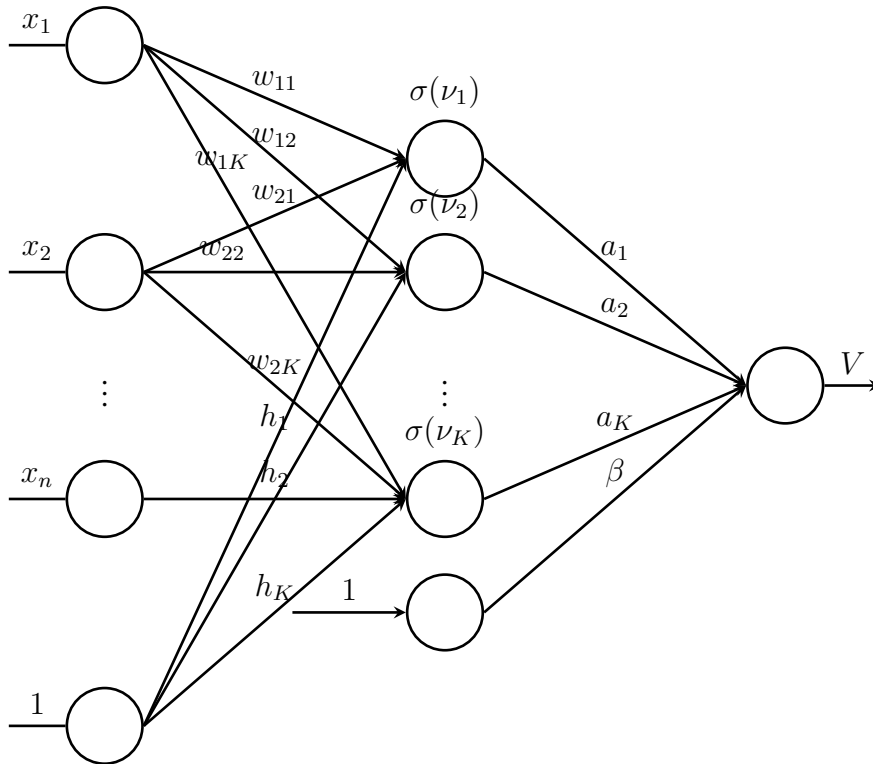


FIGURE 3.1 – Réseau neuronal de la fonction de Lyapunov

Par conséquent, $V(x)$ peut être exprimée comme suit :

$$\left\{ \begin{array}{l} V(X) = \sum_{j=1}^K a_j \sigma(\nu_j) + \beta \end{array} \right. \quad (3.18a)$$

$$\left\{ \begin{array}{l} \nu_j = \sum_{i=1}^n w_{ji} x_i + h_j \end{array} \right. \quad (3.18b)$$

La dérivée seconde de $V(x)$ peut facilement être calculée à partir de (3.18) :

$$\begin{aligned} V_{qr} &= \left. \frac{\partial^2 V}{\partial x_q \partial x_r} \right|_{x=0} \\ &= \sum_{i=1}^K a_i \left. \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} \right|_{x=0} \left. \frac{\partial \nu_i}{\partial x_r} \right|_{x=0} w_{qi} \\ &= \sum_{i=1}^K a_i \left. \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} \right|_{x=0} w_{ri} w_{qi}. \end{aligned} \quad (3.19)$$

En insérant l'expression de (3.19) dans l'équation (3.13), nous obtenons la dérivée seconde de $G(x)$ en fonction des poids du réseau neuronal, ce qui donne :

$$\begin{aligned}
G_{lp} &= \left. \frac{\partial^2 G}{\partial x_l \partial x_p} \right|_{x=0} \\
&= \sum_{j=1}^n \left(\sum_{i=1}^K a_i \left. \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} \right|_{x=0} w_{pi} w_{ji} \right) J_{jl} \\
&\quad + \sum_{j=1}^n \left(\sum_{i=1}^K a_i \left. \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} \right|_{x=0} w_{ji} w_{li} \right) J_{jp}
\end{aligned} \tag{3.20}$$

où $q = 1, \dots, n$; $r = 1, \dots, n$; $l = 1, \dots, n$ et $p = 1, \dots, n$.

Calcul détaillé. Par définition, $G(x)$ est donnée par :

$$\begin{aligned}
G(x) &= \frac{dV}{dt} = \sum_{j=1}^n \frac{\partial V}{\partial x_j} \frac{dx_j}{dt} \\
&= \sum_{j=1}^n \frac{\partial V}{\partial x_j} f_j = \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d\sigma(\nu_i)}{d\nu_i} w_{ji} \right) f_j.
\end{aligned}$$

De ce fait, la dérivée première de G est la suivante :

$$\begin{aligned}
\frac{\partial G(x)}{\partial x_p} &= \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} w_{pi} w_{ji} \right) f_j \\
&\quad + \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d\sigma(\nu_i)}{d\nu_i} w_{ji} \right) \frac{\partial f_j}{\partial x_p}.
\end{aligned}$$

Ainsi, nous obtenons :

$$\begin{aligned}
\frac{\partial^2 G(x)}{\partial x_l \partial x_p} &= \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d^3 \sigma(\nu_i)}{d\nu_i^3} w_{pi} w_{ji} w_{li} \right) f_j \\
&\quad + \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} w_{pi} w_{ji} \right) \frac{\partial f_j}{\partial x_l} \\
&\quad + \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} w_{ji} w_{li} \right) \frac{\partial f_j}{\partial x_p} \\
&\quad + \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d\sigma(\nu_i)}{d\nu_i} w_{ji} \right) \frac{\partial^2 f_j}{\partial x_l \partial x_p}.
\end{aligned}$$

En $x = 0$, nous avons $f_j(x) = 0$ par définition du point d'équilibre et $\frac{\partial f_j}{\partial x_l} \Big|_{x=0} = J_{jl}$. En conséquence, nous retrouvons l'équation (3.20) :

$$\begin{aligned} \frac{\partial^2 G(X)}{\partial x_l \partial x_p} \Big|_{x=0} &= \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} w_{pi} w_{ji} \right) J_{jl} \\ &+ \sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d^2 \sigma(\nu_i)}{d\nu_i^2} w_{ji} w_{li} \right) J_{jp} \\ &+ \underbrace{\sum_{j=1}^n \left(\sum_{i=1}^K a_i \frac{d\sigma(\nu_i)}{d\nu_i} w_{ji} \right)}_{\text{terme qui est nul par construction à la fin de l'algorithme}} \frac{\partial^2 f_j}{\partial x_l \partial x_p}. \end{aligned}$$

Les conditions (t1) à (t4), en choisissant \tanh comme fonction d'activation, définies précédemment deviennent alors :

- (t1) $\sum_{i=1}^K a_i \tanh(h_i) + \beta = 0$,
- (t2) $\sum_{i=1}^K a_i (1 - \tanh^2(h_i)) w_{qi} = 0$ pour $q = 1, \dots, n$,
- (t3) H^V donnée par (3.19) est définie positive,
- (t4) H^G donnée par (3.20) est définie négative.

Note : La contrainte (t1) peut se retirer en imposant $\beta = -\sum_{i=1}^K a_i \tanh(h_i)$ dans la construction du réseau de la fonction de coût.

3.4 Mise en œuvre de l'optimisation

C'est dans ce contexte que nous introduisons un cadre d'optimisation détaillé. Notre approche vise non seulement à identifier une fonction de Lyapunov pour un système donné mais aussi à maximiser le domaine de stabilité garanti correspondant pour obtenir une compréhension plus complète de la stabilité du système.

Les travaux décrits ici ont été publiés en conférence (BOCQUILLON et al., 2020b).

3.4.1 Cadre d'optimisation

Dans cette section, nous supposons qu'une structure appropriée pour la fonction de Lyapunov, notée V_θ (deux fois différentiable), a été choisie, où tous les paramètres qui visent à caractériser la fonction de Lyapunov sont rassemblés dans le vecteur θ .

Ensuite, pour déterminer une fonction de Lyapunov appropriée, les conditions (t1) à (t4) doivent être satisfaites. À cette fin, une fonction de coût, Q , doit être sélectionnée de manière à ce que, respectivement, la positivité et la négativité de H^V et H^G soient satisfaites. La matrice H^V est définie positive si toutes ses valeurs propres sont positives. Dans le même sens, la matrice H^G est définie négative si toutes ses valeurs propres sont négatives.

Dans l'article (PETRIDIS et PETRIDIS, 2006), les auteurs abordent la résolution de ce problème par optimisation. Bien que cette méthode offre des perspectives intéressantes, elle comporte plusieurs points délicats qui nécessitent un examen approfondi. En premier lieu, l'emploi du concept de "voisinage autour de 0", pour énoncer les conditions requises pour identifier une fonction de Lyapunov, introduit une certaine imprécision. En effet, ce concept de "voisinage" n'est pas clairement défini, ne fournissant pas d'information concrète sur le domaine de stabilité concerné, ce qui pourrait limiter la portée ou l'utilité de l'identification de la fonction de Lyapunov selon les objectifs visés.

Par ailleurs, l'intégration de fonctions barrières pour assurer la positivité de H^V et la négativité de H^G , rappelant les techniques de pénalisation en optimisation, pose également question. Bien que ces fonctions puissent faciliter la gestion des contraintes lors de l'optimisation, elles ne garantissent pas en soi que la solution obtenue soit effectivement une fonction de Lyapunov. Ceci implique une nécessité de vérification supplémentaire après le processus d'optimisation.

En conséquence, dans la prochaine section, nous élaborons une méthode permettant de maximiser le domaine de stabilité garanti. Avant d'entrer dans les détails de l'estimation du DOA, il est essentiel d'explicitier la méthodologie de la création d'un maillage. Cette étape est fondamentale car elle influence directement la qualité de nos résultats d'optimisation, ainsi que la robustesse et la précision des domaines d'attraction estimés.

3.4.2 Définition du maillage

Un maillage est un ensemble discret d'échantillons dans un espace continu, souvent utilisé pour approcher des fonctions continues à l'aide d'un ensemble fini de points. Dans notre contexte, le maillage est utilisé pour échantillonner l'espace d'état, nous permettant d'évaluer les conditions d'une fonction de Lyapunov sur cet ensemble discret plutôt que sur l'ensemble infini des états possibles.

Le maillage que nous utilisons pour l'approximation du DOA est constitué d'un ensemble \mathcal{Z} de points, qui est obtenu en subdivisant un hypercube centré sur l'origine. Si le point d'équilibre n'est pas à l'origine, il suffit de procéder à un changement d'origine comme discuté dans la section 1.2.11. Chaque subdivision ou "cellule" de cet hypercube contient un point unique. Cet hypercube est conçu

pour couvrir un domaine suffisamment vaste $\subset \mathbb{X}$ afin d'assurer une approximation de notre estimation du domaine d'attraction. La résolution de ce maillage, c'est-à-dire la distance entre les points adjacents, est déterminante pour la qualité de cette approximation. Il est essentiel que \mathcal{Z} soit choisi suffisamment grand pour couvrir intégralement la région d'intérêt de l'espace d'état.

Dans notre algorithme, nous laissons la possibilité de choisir le nombre de points du maillage, c'est-à-dire du nombre de subdivisions de l'hypercube. Toutefois, il est important de noter un compromis : un nombre élevé de points augmente la précision de l'approximation, mais le temps de calcul devient également plus important, croissant de manière exponentielle avec le nombre de points. Dans le contexte industriel, où les systèmes peuvent être d'une grande complexité, il est souvent préférable d'opter pour une solution équilibrée entre précision et temps de calcul. La robustesse de notre approche par rapport à la résolution du maillage ainsi que la stabilité entre deux points sont discutées plus en détail dans la section 3.4.5.

Il existe d'autres méthodes pour déterminer le maillage, mais dans nos travaux, nous avons utilisé celle présentée précédemment. Par exemple, une perspective possible consiste à utiliser le gradient du candidat actuel de la fonction de Lyapunov pendant l'optimisation. Une idée pourrait être d'analyser la première exécution de l'algorithme d'optimisation ; plus le gradient est grand, plus le maillage doit être fin dans les exécutions suivantes. Le choix d'un maillage adapté est une étape cruciale pour obtenir des résultats d'optimisation fiables. L'échantillonnage discret de l'espace d'état par le maillage permet d'évaluer les propriétés des fonctions de Lyapunov ainsi que sa dérivée sur un ensemble fini de points, rendant le problème d'optimisation solvable.

3.4.3 Estimation d'un DOA

La constitution de notre maillage étant défini, nous pouvons l'utiliser pour maximiser le domaine de stabilité garanti. Notre maillage, désigné par \mathcal{Z} , est un ensemble discret de points obtenus en subdivisant un hypercube. Pour cette section, nous considérons une fonction de Lyapunov d'une forme choisie au préalable V_θ donnée, et D son domaine de définition associée. Nous allons nous concentrer sur la stabilité au sens de Lyapunov et la stabilité asymptotique, en laissant la stabilité exponentielle et la stabilité entrée - état pour les sections ultérieures.

L'ambition ici est de maximiser le domaine de stabilité estimé garanti en identifiant le level set $\Omega(c)$ le plus grand (que nous appellerons Ω_c par la suite) dans lequel la dérivée temporelle de V_θ demeure négative, conformément au [Théorème 21](#). Cela revient essentiellement à déterminer la valeur maximale de c , soit c_* défini dans (1.48), pour laquelle Ω_c est à l'intérieur du DOA. En imposant une convexité locale à V et une concavité locale à G , nous garantissons que la notion de level set est bien définie.

Pour une illustration claire et une meilleure compréhension des relations entre le domaine d'attraction, le domaine de définition de la fonction de Lyapunov et le domaine de stabilité estimé garanti, nous renvoyons le lecteur à la [Figure 3.2](#) ci-dessous. Pour rappel, le domaine de stabilité garanti peut être conceptualisé comme une estimation du DOA, à l'intérieur duquel tout point initial est assuré de converger vers un état stable.

Lorsque nous considérons un V_θ fixe qui satisfait les conditions, il est nécessaire de trouver la valeur maximale $c_* \in \mathbb{R}$ pour laquelle Ω_c satisfait les conditions du [Théorème 21](#), voir le [Théorème 22](#). La valeur de c_* peut varier en fonction des échelles de V_θ , comme illustré par le fait que si V_θ est multiplié par un facteur positif α , c_* se transforme aussi en $\alpha \times c_*$. Néanmoins, ce qui est essentiel n'est pas la valeur absolue de c , mais plutôt la taille du level set associé à V_θ . Dans ce but, nous appliquons l'algorithme que nous avons précédemment décrit en pseudo-code (voir [Algorithme 1](#)), basé sur les principes énoncés dans ([NAJAFI et al., 2016](#)).



FIGURE 3.2 – Visualisation du domaine de stabilité estimé garanti

Afin de pouvoir maximiser le domaine de stabilité estimé garanti, nous utilisons le maillage \mathcal{Z} pour échantillonner l'espace d'état. La première étape vise à définir un ensemble de points à travers le maillage \mathcal{Z} qui couvre intégralement le domaine d'attraction estimé, garantissant ainsi que \mathcal{Z} contient entièrement l'ensemble Ω_c , inclus dans le DOA. La deuxième partie de la maximisation de la taille de Ω_c consiste à rechercher la borne supérieure du paramètre c_* dans l'équation (1.48), calculée dans tout l'espace de recherche \mathcal{Z} .

Le processus d'estimation du DOA est un élément central de notre exploration. Afin d'atteindre cet objectif, nous proposons deux méthodes distinctes, chacune adoptant une approche unique pour explorer et maximiser le domaine de stabilité estimé garanti. Ces méthodes sont conçues pour fournir un cadre flexible, permettant une analyse approfondie et une maximisation efficace de ce domaine. Dans les sections suivantes, nous détaillons ces deux méthodes, en exposant leur fonctionnement, leurs avantages, et la manière dont elles contribuent au processus global d'estimation du DOA.

3.4.3.1 Méthode 1 : Estimation indirecte d'un DOA

Dans la première méthode, nous travaillons directement sur les points vérifiant les conditions (t1) à (t4). Nous avons besoin d'une métrique pour évaluer la qualité de nos estimations. Introduisons le critère P_1 , qui comptabilise les points qui satisfont les caractéristiques requises d'une fonction de Lyapunov. Plus précisément, P_1 est calculé comme le nombre de points pour lesquels $V(x) > 0$ et $G(x) < 0$, respectant ainsi les conditions (t1) à (t4).

L'objectif est alors de maximiser P_1 (ou, de manière équivalente, de minimiser $-P_1$), soit maximiser les points respectant les conditions (t1) à (t4), sans considération préalable des implications sur le level set associé. Cette approche vise à optimiser le domaine de définition de la fonction de Lyapunov, noté D . Bien que cette méthode offre une opportunité d'estimer le DOA d'un système donné, elle n'offre aucune certitude et ne produit qu'une estimation initiale du domaine de stabilité garanti. Son avantage réside dans sa simplicité d'utilisation.

Toutefois, cette méthode présente des limites. En effet, bien que P_1 soit maximisé, il n'est pas garanti que le level set correspondant soit le plus grand possible. Un point peut vérifier les conditions (t1) à (t4), pour autant, la trajectoire d'état passant par ce point peut aller vers des points où $V(x)$ est plus petite, mais où $G(x)$ ne soit plus < 0 . En conséquence, rien ne garantit que le D trouvé soit invariant. De ce fait, une vérification a posteriori s'avère alors nécessaire pour évaluer et valider la grandeur du level set associé au D obtenu, ajoutant une couche d'effort calculatoire et conceptuel à la procédure. Nous introduisons alors une seconde méthode.

Pour pallier ce problème, l'utilisation des level sets (ensembles de sous-niveaux) est proposée dans la méthode 2. Ces ensembles vérifient $V(x) > 0$ et $G(x) < 0$, mais de plus, une fois la trajectoire dans cet ensemble, elle y reste pour tout $t > t_0$.

3.4.3.2 Méthode 2 : Estimation directe du DOA avec des level sets

Cette deuxième méthode propose une approche alternative, centrée explicitement sur la maximisation de la taille du level set. Dans cette stratégie, nous introduisons une métrique spécifique, dénommée P_2 , visant à identifier et comptabiliser les points qui correspondent aux caractéristiques d'un level set optimal. Plus précisément, P_2 est calculée comme le nombre de points pour lesquels $G(X) < 0$ et $V(x) \leq c_*$, respectant ainsi les conditions strictes du level set. Cette métrique P_2 sert d'indicateur direct et fiable de la grandeur du level set. Par conséquent, en maximisant P_2 (ou, de manière équivalente, en minimisant $-P_2$), nous obtenons une estimation d'un DOA, un domaine garanti, qui est intrinsèquement liée à la grandeur du level set. L'avantage principal de cette méthode réside dans sa focalisation directe sur le level set, permettant ainsi d'obtenir des résultats qui n'exigent pas de vérification a posteriori, simplifiant et accélérant le processus de validation. Ainsi, cette méthode se présente comme une alternative puissante et efficace pour maximiser directement le level set comme une estimation d'un sous-DOA, facilitant et optimisant le processus d'optimisation et de validation.

Une analyse comparative détaillée de ces deux méthodes est présentée dans la section 4.2. Cette comparaison permet de discerner la pertinence et l'efficacité relative de chaque approche dans différents contextes et configurations. Nous examinons de manière critique les avantages et les inconvénients de chaque méthode, en mettant en lumière leurs performances respectives, leur fiabilité, ainsi que leur aptitude à maximiser le level set tout en approximant de manière précise le DOA. Cette seconde méthode n'est pas seulement importante pour l'ajustement précis de la méthode choisie tout au long de nos travaux, mais elle incarne le cheminement intellectuel qui nous a orientés vers l'estimation du DOA avec des level sets.

3.4.4 Fonction de coût

La recherche d'une fonction de Lyapunov qui valide la stabilité d'un système dynamique est au cœur de nombreux problèmes d'ingénierie et de contrôle. La caractérisation d'une telle fonction est intrinsèquement liée à un problème d'optimisation non trivial. L'espace des solutions est non convexe, résultant des contraintes associées, et la fonction de coût est elle-même non convexe, présentant ainsi plusieurs minima locaux. Cela implique que des algorithmes d'optimisation conventionnels, qui sont conçus pour des espaces convexes, pourraient se retrouver facilement piégés dans un minimum local et donc ne pas fournir la solution optimale souhaitée.

Notre démarche vise à déterminer une fonction de Lyapunov en formulant des critères clés que cette fonction doit satisfaire. Rappelons les critères (t1) à (t4) que nous avons déterminés dans la section précédente :

(t1) $V(0) = 0$,

(t2) $G(0) = 0$,

(t3) H^V , telle que donnée par (3.6), est définie positive,

(t4) H^G , telle que donnée par (3.14), est définie négative.

Les contraintes (t1) et (t2) spécifient que la fonction de Lyapunov et sa dérivée doivent être égales à 0 au point d'équilibre. Les contraintes (t3) et (t4) de convexité locale pour V et de concavité locale pour G , sont cruciales pour justifier une fonction de Lyapunov à la fin du processus d'optimisation et pour définir la notion de level set. En imposant une convexité locale à V et une concavité locale à G , nous garantissons que la notion de level set est bien définie.

Pour naviguer efficacement dans cet espace complexe et potentiellement vaste de solutions, nous nous tournons vers des techniques d'optimisation bio-inspirées : l'algorithme génétique (GA), tel que défini à la section 2.4.2.1, et l'optimisation par essaim de particules (PSO), présentée dans la section 2.4.2.2. Ces deux algorithmes tirent leur inspiration de phénomènes naturels et ont été prouvés comme étant particulièrement efficaces pour des problèmes d'optimisation non convexes.

Le choix de ces algorithmes est stratégique. D'une part, le GA est particulièrement adapté à notre implémentation sous contraintes, respectant les contraintes (t1) et (t2), car il offre la possibilité d'appliquer des contraintes dures. Une contrainte dure est une contrainte qui doit impérativement être satisfaite. Si un candidat ne respecte pas cette contrainte, il est rejeté ou pénalisé sévèrement, garantissant ainsi que toutes les solutions générées par l'algorithme respectent ces critères.

D'autre part, le PSO est notre choix pour l'implémentation sans contrainte externe. En supprimant ces contraintes, nous simplifions la procédure d'optimisation, permettant ainsi une exécution plus rapide. De plus, cela nous offre une flexibilité accrue dans le choix des algorithmes d'optimisation, puisque nous ne sommes plus limités par ces contraintes. Cependant, il est important de mentionner que les contraintes (t1) et (t2) restent centrales dans notre étude. La section dédiée aborde leur intégration de manière appropriée.

Ainsi, cette section a pour objectif de mettre en place des fonctions de coût permettant de déterminer une fonction de Lyapunov répondant aux contraintes (t1) à (t4).

3.4.4.1 Implémentation sous contraintes

Nous formalisons maintenant le problème comme un problème sous contraintes pour éviter l'utilisation de la fonction barrière qui conduit à un problème sous-optimal comme montré dans l'article (PETRIDIS et PETRIDIS, 2006). Le schéma proposé ici est flexible de sorte que des problèmes plus complexes tels que la stabilité exponentielle et l'ISS peuvent être considérés avec le même schéma, et sont abordés efficacement dans les prochaines sections.

Le problème peut être exprimé sous la forme générale d'un problème d'optimisation dans lequel la fonction de coût Q doit être minimisée. Dans ce but, notons $\lambda_i^v, i = 1, \dots, n_v$ l'ensemble des n_v valeurs propres de H^V et $\lambda_i^g, i = 1, \dots, n_g$ l'ensemble des n_g valeurs propres de H^G . Si n est l'ordre du système, alors $n_v = n_g = n$. Les conditions de positivité et de négativité sur les matrices H^V et H^G nécessitent des signes opposés.

Afin de répondre à la minimisation de la fonction de coût, nous posons :

$$H^{V'} = H^V \times -1.$$

Notons $\lambda^{v'}$ les valeurs propres de $H^{V'}$:

$$\bar{\lambda}^{v'} = \max(\text{real}(\lambda^{v'}))$$

$$\bar{\lambda}^g = \max(\text{real}(\lambda^g))$$

$$\lambda_{\max} = \max(\bar{\lambda}^{v'}, \bar{\lambda}^g).$$

Alors, la fonction de coût à minimiser, dont l'expression de \mathcal{Q} est définie dans la section suivante, se présente comme suit, avec Θ représentant le vecteur des variables à déterminer :

$$\begin{aligned} & \min_{\Theta} \mathcal{Q} \\ & \text{subject to } (t1), (t2). \end{aligned}$$

Dans le cas de la stabilité asymptotique, Θ est réduit simplement à θ , qui représente tous les paramètres visant à caractériser la fonction de Lyapunov. Dans ce contexte, la manière dont nous calculons la fonction de coût \mathcal{Q} est décisive. Nous examinons plus en détail la formulation de cette fonction selon les deux méthodes distinctes que nous avons définies précédemment. Chacune de ces approches présente des caractéristiques uniques dans le calcul de \mathcal{Q} . Nous effectuons une exploration approfondie de leurs procédures respectives de calcul et des implications associées.

Méthode 1 : Estimation indirecte d'un DOA

Dans cette sous-section, nous détaillons la première méthode de calcul de la fonction de coût \mathcal{Q} , qui est centrée sur la maximisation du domaine de définition D . Le pseudo-code ci-dessous (voir [Algorithme 2](#)) clarifie ce processus en illustrant les étapes principales du calcul. Dans cet algorithme, x_{mesh} correspond au maillage que nous avons défini dans la section précédente, soit \mathcal{Z} .

Algorithme 2 : Pseudo-code pour le calcul de la fonction de coût \mathcal{Q} avec la méthode 1

Input : Maillage x_{mesh} , paramètres du modèle de la fonction de Lyapunov Θ

Result : La valeur de la fonction de coût \mathcal{Q}

$$\lambda_{\max} = \max(\bar{\lambda}^{v'}, \bar{\lambda}^g)$$

if $\lambda_{\max} \geq 0$ **then**

$$| \mathcal{Q} = \lambda_{\max};$$

else

$$| P_1 = 0;$$

for *pour chaque point m dans x_{mesh}* **do**

if $G(m) < 0$ **et** $V(m) > 0$ **then**

$$| | P_1 = P_1 + 1;$$

end

end

$$| \mathcal{Q} = -P_1;$$

end

return \mathcal{Q}

Cette fonction de coût est similaire à celle décrite dans (FEYEL, 2017), et a démontré son efficacité. Selon la définition de \mathcal{Q} , un candidat est une fonction de Lyapunov valide si $\mathcal{Q} < 0$, ce qui signifie que les conditions (t3) et (t4) sont satisfaites. Les conditions (t1) et (t2), qui sont cruciales, sont intégrées comme contraintes d'égalité dans l'algorithme d'optimisation, garantissant ainsi que les solutions respectent ces critères impératifs. Pour rappel, P_1 représente le nombre de points dans le maillage vérifiant les conditions $V(x) > 0$ et $G(x) < 0$. Mathématiquement, la fonction de coût \mathcal{Q} peut être exprimée comme suit :

$$\mathcal{Q} = \begin{cases} \lambda_{\max} & \text{si } \lambda_{\max} \geq 0 \\ -P_1 & \text{sinon} \end{cases} \quad (3.21)$$

Dans cette méthode, l'accent est mis sur la maximisation des points satisfaisant les conditions (t1) à (t4) sans prendre en compte, dans un premier temps, les implications de cette maximisation sur le level set associé. Bien que cette stratégie permette une optimisation intuitive et efficace de D , fournissant ainsi une approximation initiale robuste de la région d'attraction, elle n'est pas exempte d'une contrainte majeure. La limite principale de cette approche réside dans l'incertitude concernant la taille du level set associé. En d'autres termes, bien que D soit maximisé, il est possible que le level set correspondant ne le soit pas, conduisant à une optimisation non concluante car D n'est pas invariant. Dans un tel scénario, le processus d'optimisation pourrait aboutir à un D élargi, mais associé à un level set relativement restreint, compromettant ainsi l'efficacité de l'optimisation. Ainsi, une étape de vérification a posteriori est indispensable pour évaluer et confirmer la taille du level set associé au D identifié. Cette étape supplémentaire, nécessaire pour assurer une optimisation significative, ajoute une couche de complexité et requiert des efforts calculatoires supplémentaires.

Méthode 2 : Maximisation directe sur le level set

Cette sous-section introduit une seconde approche de calcul pour la fonction de coût, où l'accent est mis sur la maximisation directe d'un level set associé au DOA. Cette méthode vise à obtenir le plus grand level set directement lors du processus d'optimisation, sans nécessité d'une vérification a posteriori, simplifiant ainsi le processus d'évaluation.

Afin de clarifier ce processus, nous fournissons ci-dessous un pseudo-code détaillé qui illustre les étapes principales de ce calcul (voir [Algorithme 3](#)). Dans cet algorithme, x_{mesh} correspond au maillage que nous avons défini dans la section précédente, soit \mathcal{Z} . C est une fonction de coût similaire à celle que nous pouvons trouver dans ([FEYEL, 2017](#)) et qui a prouvé son efficacité.

Algorithme 3 : Pseudo-code pour le calcul de la fonction de coût Q avec la méthode 2

Input : Maillage x_{mesh} , paramètres du modèle de la fonction de Lyapunov Θ

Result : La valeur de la fonction de coût Q

```

 $\lambda_{\max} = \max(\bar{\lambda}^{v'}, \bar{\lambda}^g)$ 
if  $\lambda_{\max} \geq 0$  then
    |  $Q = \lambda_{\max}$ ;
else
    |  $P_2 = 0$ ;
    | Utilisez l'Algorithme 1 pour définir  $c_*$ ;
    | for pour chaque point  $m$  dans  $x_{\text{mesh}}$  do
    | | if  $G(m) < 0$  et  $V(m) < c_*$  then
    | | |  $P_2 = P_2 + 1$ ;
    | | end
    | end
    |  $Q = -P_2$ ;
end
return  $Q$ 

```

Selon la définition de la fonction Q , un candidat est une fonction de Lyapunov si $Q < 0$. Soit les conditions (t3) et (t4) sont vérifiées. Les conditions (t1) et (t2) définissent les propriétés spécifiques que nous souhaitons obtenir pour la fonction de Lyapunov, et sont intégrées comme contraintes d'égalité dans l'algorithme d'optimisation, garantissant ainsi que les solutions obtenues respectent ces critères.

S'appuyant sur l'efficacité de l'Algorithme 1 présenté dans la section 1.5.6 pour la détermination optimale de la valeur de c_* , notre méthode intègre cette technique fondamentale pour optimiser le calcul de P_2 . Pour rappel, la métrique P_2 est calculée comme le nombre de points pour lesquels $G(x) < 0$ et $V(x) \leq c_*$.

Mathématiquement, la fonction de coût \mathcal{Q} peut être exprimée comme suit :

$$\mathcal{Q} = \begin{cases} \lambda_{\max} & \text{si } \lambda_{\max} \geq 0 \\ -P_2 & \text{sinon} \end{cases} \quad (3.22)$$

Synthèse des méthodes

Les deux méthodes révèlent une structure commune dans la procédure, comme illustré dans les pseudo-codes de l'Algorithme 2 et de l'Algorithme 3. La distinction réside dans l'espace où nous comptons les points : la méthode 1 effectue le comptage directement sur le domaine D , tandis que la méthode 2 ajoute un processus intermédiaire pour déterminer la valeur de c_* au cours du processus d'optimisation.

Plusieurs avantages découlent de l'utilisation de ces méthodes d'implémentation sous contraintes :

- En premier lieu, il est important de certifier que les valeurs propres sont réellement positives et négatives pour H^V et H^G . Une solution aurait pu être de seulement compter les points à maximiser afin de post-vérifier ces conditions sur H^V et H^G . Cependant, cela ne garantit pas que la fonction de Lyapunov trouvée à la fin de l'optimisation possède un minimum local en 0 et/ou que sa dérivée possède un maximum local en 0.
- Le deuxième avantage de ces approches est sa grande flexibilité. Elle permet une adaptation facile à une multitude de problèmes complexes, comme étendre à d'autres types de stabilité.
- Enfin, aucun paramètre supplémentaire n'est à régler.

Les inconvénients des deux méthodes sont le temps de calcul du comptage de chaque point du maillage pour savoir si les conditions sont assurées, et la robustesse au maillage que nous abordons plus en détail dans une section ultérieure.

3.4.4.2 Implémentation sans contrainte externe

L'importance de l'algorithme d'optimisation dans la détermination d'une fonction de Lyapunov ne peut être sous-estimée. La méthode d'implémentation précédente, bien qu'efficace, introduit deux contraintes d'égalité dans l'algorithme d'optimisation (les conditions (t1) et (t2)). Bien que ces contraintes soient essentielles pour garantir que la solution trouvée soit effectivement une fonction de Lyapunov, elles réduisent également le nombre d'algorithmes d'optimisation que nous pouvons utiliser. Cette restriction est importante, car avoir une variété d'algorithmes à notre disposition peut nous permettre d'explorer un ensemble plus large de solutions potentielles, et par conséquent peut augmenter nos chances de trouver une solution optimale.

C'est pourquoi nous avons opté pour une approche différente avec cette nouvelle implémentation. En supprimant les contraintes d'égalité externes, nous avons créé une méthode qui permet une plus grande flexibilité dans le choix de l'algorithme d'optimisation. Cette modification ouvre la possibilité d'utiliser des algorithmes tels que le PSO, que nous avons introduit dans la section 2.4.2.2.

Nous procédons maintenant à une explication détaillée de cette nouvelle implémentation à travers le pseudo-code suivant. L'Algorithme 4 décrit les étapes importantes de ce calcul, en mettant l'accent

sur l'intégration des contraintes dans la fonction de coût. La fonction $my_nlconst_Lyap(x, n)$ n'est pas détaillée dans le pseudo-code mais établit les contraintes (t1) et (t2) que l'optimisation doit satisfaire. Notons que x_{mesh} fait référence au maillage défini précédemment, représentant l'ensemble \mathcal{Z} .

Algorithme 4 : Pseudo-code pour le calcul de la fonction de coût Q avec ajustement des contraintes

Input : Maillage x_{mesh} , paramètres du modèle de la fonction de Lyapunov Θ

Result : La valeur de la fonction de coût Q

```

 $\lambda_{max} = \max(\bar{\lambda}^{v'}, \bar{\lambda}^g);$ 
 $[\sim, ceq] = my\_nlconst\_Lyap(\theta, n);$ 
 $E = \max(abs(ceq));$ 
 $tol = 1 * 10^{-10};$ 
if  $E - tol$  then
    |  $\lambda_{max} = \max(\lambda_{max}, E - tol);$ 
end
if  $\lambda_{max} \geq 0$  then
    |  $Q = \lambda_{max};$ 
else
    |  $P_2 = 0;$ 
    | Utilisez l'Algorithme 1 pour définir  $c_*$ ;
    | for pour chaque point  $m$  dans  $x_{mesh}$  do
    | | if  $G(m) < 0$  et  $V(m) < c$  then
    | | |  $P_2 = P_2 + 1;$ 
    | | end
    | end
    |  $Q = -P_2;$ 
end
return  $Q$ 

```

Ainsi, notre implémentation sans contrainte externe offre une plus grande liberté dans le choix des algorithmes d'optimisation, et offre donc une nouvelle perspective sur la manière optimale d'identifier les fonctions de Lyapunov. La spécificité de cette méthode repose sur l'intégration des contraintes (t1) et (t2) directement dans la fonction de coût, non pas comme des pénalités, mais comme des conditions intégrées à cette fonction. Cette approche garantit le respect des contraintes avec une précision de $1 * 10^{-10}$, une tolérance qui est similaire à celle utilisée par le GA de MATLAB.

3.4.5 Robustesse au maillage

La robustesse au maillage désigne la capacité d'une méthode ou d'un algorithme à maintenir des résultats fiables et précis entre deux points du maillage, malgré les variations de qualité, comme la densité et la distribution des nœuds. Dans le cadre spécifique de notre étude, qui concerne l'estimation d'un domaine d'attraction en utilisant la théorie de Lyapunov, la robustesse au maillage est d'une importance particulière. En résumé, nous voulons vérifier que la fonction considérée est bien une fonction de Lyapunov entre deux points du maillage.

Dans un premier temps, il est primordial de s'assurer de la robustesse au maillage de notre méthode pour des raisons mathématiques. S'assurer mathématiquement que le maillage permet de vérifier

certaines contraintes entre chaque point est essentiel pour la fiabilité et la robustesse de notre approche, c'est-à-dire que les points intermédiaires non considérés dans le maillage vérifient aussi la contrainte.

Dans un second temps, un autre défi que nous avons évoqué est celui du temps de calcul. Pour des systèmes d'ordre élevé, que nous retrouvons régulièrement dans l'industrie, le maillage représente un véritable enjeu. Chaque point ajouté au maillage augmente considérablement la complexité du calcul. Si nous devons recourir à un maillage extrêmement fin pour prouver la stabilité de ces systèmes, le temps de calcul augmenterait considérablement. Garantir la robustesse de notre méthode vis-à-vis du maillage est essentiel non seulement d'un point de vue théorique, mais aussi pratique. En effet, cela nous autorise à utiliser un maillage plus grossier, tout en maintenant un degré acceptable de précision. Cette approche réduit significativement le temps de calcul, rendant ainsi notre méthode adaptée aux systèmes de haute dimension.

La [Figure 3.3](#) représente la fonction $V(x)$ sur l'espace d'état. Alors que les points du maillage sont clairement marqués (représentés par la ligne pointillée en bleu), il est essentiel d'observer les régions situées entre ces points. En effet, comme nous pouvons le constater, il est possible qu'une région entre deux points du maillage s'avère problématique. Notamment, si le comportement de V change de manière significative dans cet intervalle, V devient négative dans notre exemple. Ensuite, la [Figure 3.4](#) montre le comportement de la fonction $G(x)$. Les préoccupations soulevées pour $V(x)$ entre les points de maillage s'appliquent de la même manière.

C'est ici que se pose le défi : dans notre cas, comment garantir que la positivité de V et la négativité de G qui sont vérifiées sur les points du maillage, le sont également sur les points intermédiaires, ceux qui ne sont pas directement représentés dans le maillage ?

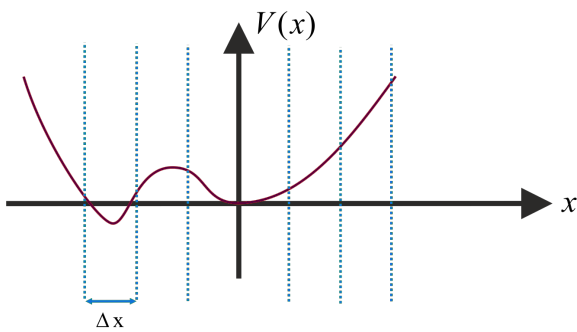


FIGURE 3.3 – Variation de V en fonction du maillage

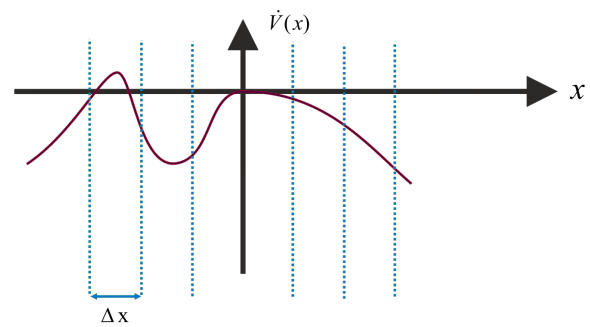


FIGURE 3.4 – Variation de \dot{V} en fonction du maillage

Pour surmonter ce problème, il est nécessaire d'adopter une approche qui va au-delà de la simple vérification des points du maillage. Grâce à la théorie des fonctions de Lipschitz, nous sommes en mesure de prédire des propriétés des points intermédiaires à partir de celles des points du maillage. En d'autres termes, si nous pouvons garantir que les "contraintes" sont satisfaites sur l'ensemble du maillage, alors, en utilisant la continuité et les propriétés liées, nous pouvons extrapoler cette garantie aux points intermédiaires. Cette extrapolation est décisive pour s'assurer que notre méthode est robuste et fiable sur l'ensemble de l'espace d'état, et pas seulement sur un sous-ensemble discrétisé.

Une solution, selon ([RICHARDS et al., 2018](#)), est de vérifier la contrainte $G(x) < -L_G \cdot \Delta x$ (excepté au point 0) à un ensemble fini de points qui couvrent Ω_c , où $L_G \in \mathbb{R}^+$ est estimée entre les points du maillage et représente la constante de Lipschitz de G , et $\Delta x \in \mathbb{R}^+$ correspond à la largeur du maillage ([BERKENKAMP et al., 2017](#)). Il est important de noter que la constante de Lipschitz doit être évaluée entre les points du maillage pour assurer une estimation précise.

De notre côté, nous avons également ajouté la contrainte $V(x) > L_V \cdot \Delta x$ (excepté au point 0) à un ensemble fini de points qui couvrent Ω_c . Ici aussi, $L_V \in \mathbb{R}^+$ doit être estimée entre les points du maillage et non pas aux points eux-mêmes, agissant comme la constante de Lipschitz pour V , avec Δx représentant toujours la largeur du maillage.

Pour assurer la robustesse au maillage dans nos différentes approches, nous intégrons une procédure standardisée dans tous nos algorithmes. En intégrant cette procédure dans l'Algorithme 2, nous obtenons l'Algorithme 5.

Algorithme 5 : Pseudo-code pour le calcul de la fonction de coût Q avec la méthode 1, incluant la robustesse au maillage

Input : Maillage x_{mesh} , paramètres du modèle de la fonction de Lyapunov Θ

Result : La valeur de la fonction de coût Q

$\lambda_{\max} = \max(\bar{\lambda}^{v'}, \bar{\lambda}^g)$

if $\lambda_{\max} \geq 0$ **then**

$Q = \lambda_{\max};$

else

$P_1 = 0;$

$L_V, L_G, \Delta x;$

for *pour chaque point m dans x_{mesh}* **do**

if $G(m) < -L_G \cdot \Delta x$ *et* $V(m) > L_V \cdot \Delta x$ **then**

$P_1 = P_1 + 1;$

end

end

$Q = -P_1;$

end

return Q

3.4.5.1 Constante de Lipschitz

Le calcul de la constante de Lipschitz peut se faire de différentes formes. Il est possible d'utiliser la définition de Lipschitz pour une fonction en plusieurs variables. Pour rappel, une fonction f est dite Lipschitzienne sur un ensemble E si et seulement si il existe une constante L telle que pour tous les points x et y dans E , nous avons (WOOD et ZHANG, 1996) :

$$|f(x) - f(y)| \leq L \cdot \|x - y\|$$

où la constante L est appelée la constante de Lipschitz de la fonction.

Dans notre cas d'utilisation, l'estimation de la constante de Lipschitz pour une fonction donnée ne se réalise pas simplement en parcourant tous les points du maillage. En réalité, le calcul de la constante de Lipschitz L sur chaque intervalle repose essentiellement sur le gradient de la fonction au sein de cet intervalle. Concrètement, la constante de Lipschitz est obtenue en maximisant la norme du gradient de la fonction sur chaque intervalle. Pour la fonction $V(x)$, la démarche est la suivante :

- Nous disposons déjà de la fonction $V(x)$.
- Nous calculons formellement son gradient, $dV(x)$.
- Nous utilisons la fonction `fminsearch` dans MATLAB pour maximiser la norme $\|dV(X)\|$ entre chaque point du maillage.

La même procédure est adoptée pour $G(x)$. Il est essentiel de comprendre que cette approche offre une représentation de la variation de la fonction sur l'intervalle choisi, et non uniquement basée sur les points discrétisés du maillage. Cette méthode peut être coûteuse en terme de temps de calcul, surtout si le nombre de points dans le maillage est grand.

3.5 Différentes formes de stabilité

La stabilité d'un système est un concept central en théorie du contrôle et joue un rôle important dans la conception et l'analyse de systèmes dynamiques. Elle détermine la manière dont un système réagit à des perturbations, que ce soit en revenant à son état d'équilibre ou en s'éloignant de celui-ci. Cependant, il existe différentes nuances et définitions de stabilité, en fonction des propriétés spécifiques que nous souhaitons garantir pour un système donné. Ces différentes formes de stabilité, bien que reposant sur des principes communs, présentent des caractéristiques et des exigences uniques qui nécessitent des techniques d'analyse spécifiques. Dans cette section, nous explorons plusieurs de ces formes de stabilité, à savoir la stabilité exponentielle et la stabilité entrée - état, en détaillant leurs fondements théoriques et les méthodes associées pour les étudier et les garantir.

De plus, la mise en œuvre de l'optimisation, définie précédemment, a été conçue pour simplifier considérablement l'étude de la stabilité de ces formes complexes. Il est à noter que peu de paramètres nécessitent d'être modifiés dans l'algorithme, rendant ainsi son application flexible et efficace.

3.5.1 Stabilité exponentielle

3.5.1.1 Rappels théoriques

Considérons le système donné par (3.23) :

$$\dot{x} = f(x). \quad (3.23)$$

Soit $x_e = 0$ un point d'équilibre pour (3.23) et un ensemble $D \subset \mathbb{R}^n$ contenant 0 dans son intérieur. Soit $V : D \rightarrow \mathbb{R}$ une fonction continue telle que :

$$0 < k_1 \|x\|^a \leq V(x) \leq k_2 \|x\|^a \text{ dans } D \setminus \{0\} \quad (3.24)$$

$$G(x) \leq -k_3 \|x\|^a < 0 \text{ dans } D \setminus \{0\} \quad (3.25)$$

où k_1, k_2, k_3 et a sont des constantes positives, alors le point x_e est exponentiellement stable.

Note : $\tau < \frac{k_2}{k_3}$ est une borne supérieure du taux de convergence exponentielle de $\|x(t)\|$ (PEUTEMAN et AEYELS, 2002).

3.5.1.2 Mise en œuvre de la méthode

Il est important de rappeler que le but de cette section est de trouver un point d'équilibre qui satisfait les conditions énoncées dans les rappels théoriques.

D'un point de vue mathématique, la forme et la définition de la fonction de Lyapunov restent inchangées. La fonction de Lyapunov V doit toujours avoir un minimum local en 0 et sa dérivée G un maximum local en 0. De ce fait, les matrices H^V et H^G introduites dans les sections 3.2.1 et 3.2.2 sont définies de la même manière.

La principale distinction entre la stabilité exponentielle et la stabilité asymptotique réside dans la manière dont nous mesurons la "non-conformité" à la stabilité souhaitée, mais également dans la prise en compte de τ , qui représente la borne supérieure du taux de convergence exponentielle de $\|x(t)\|$. Nous considérons toujours \mathcal{Z} comme un ensemble de points obtenus à partir d'un hypercube dont les faces sont quadrillées afin de couvrir une surface suffisamment grande $\subset \mathbb{X}$.

Un paramètre τ_{max} est introduit manuellement dans l'algorithme, et sa valeur peut être modifiée à chaque exécution. En imposant la contrainte $\tau < \frac{k_2}{k_3} < \tau_{max}$, nous définissons une limite supérieure sur la vitesse de convergence du système étudié. La contrainte $\tau < \tau_{max}$, que nous pouvons désigner par la contrainte (t5), est prise en compte de la même manière que les autres contraintes comme (t1) et (t2) dans notre approche d'optimisation. La prise en compte de ce paramètre est effective pour les deux méthodes présentées dans la suite.

De même, bien qu'il soit impossible de maximiser le DOA car il est intrinsèque à la nature du système, il n'est pas non plus possible de maximiser la vitesse de convergence d'un système vers son point d'équilibre; nous pouvons seulement estimer le temps de convergence de celui-ci.

Pour la méthode 1, estimation indirecte d'un DOA, le compteur P_1 sert à quantifier la mesure dans laquelle les conditions nécessaires pour cette forme de stabilité sont respectées. Plus précisément, P_1 est calculé comme le nombre de points pour lesquels $0 < k_1\|x\|^a \leq V(x) \leq k_2\|x\|^a < c$ et $G(x) \leq -k_3\|x\|^a < 0$. En cherchant à maximiser P_1 , nous visons à élargir le domaine de définition D de la stabilité exponentielle. Toutefois, il est indispensable d'effectuer une vérification a posteriori du level set associé pour s'assurer de la validité et de la pertinence des résultats obtenus par cette maximisation.

Concernant la méthode 2 de la maximisation directe sur le level set, nous présentons ci-dessous un pseudo-code mis à jour qui prend en compte les critères de stabilité exponentielle. Ce pseudo-code, défini par l'Algorithme 6, assure que les conditions spécifiques à la stabilité exponentielle sont intégrées dans le processus de calcul, conformément aux rappels théoriques.

Algorithme 6 : Pseudo-code pour le calcul de la fonction de coût Q pour la stabilité exponentielle avec la méthode 2

Input : x_{mesh} , paramètres du modèle de la fonction de Lyapunov Θ , τ_{max}
Result : La valeur de la fonction de coût Q

```

 $\lambda_{max} = \max(\bar{\lambda}^{v'}, \bar{\lambda}^g);$ 
if  $\lambda_{max} \geq 0$  then
    |  $Q = \lambda_{max};$ 
else
    |  $P_2 = 0;$ 
    | Utilisez l'Algorithme 1 pour définir  $c_*$ ;
    | for pour chaque point  $m$  dans  $x_{mesh}$  do
    | | if  $G(m) \leq -k_3\|m\|^a < 0$  et  $0 < k_1\|m\|^a \leq V(m) \leq k_2\|m\|^a < c$  then
    | | |  $P_2 = P_2 + 1;$ 
    | | end
    | end
    |  $Q = -P_2;$ 
end
return  $Q$ 

```

Ainsi, la fonction de coût à minimiser a la forme suivante, où l'expression de Q a été définie dans le pseudo-code de l'Algorithme 6, et où Θ représente le vecteur des variables à déterminer :

$$\begin{aligned} & \min_{\Theta} Q \\ & \text{subject to } (t1), (t2), \frac{k_2}{k_3} < \tau_{max}. \end{aligned}$$

Dans le contexte de la stabilité exponentielle, Θ englobe non seulement θ , qui représente tous les paramètres destinés à caractériser la fonction de Lyapunov, mais également les constantes k_1, k_2, k_3, a si elles sont > 0 .

La méthode d'optimisation reste la même que précédemment, similaire au pseudo-code détaillé de l'Algorithme 2 avec la nouvelle définition de P_1 et la prise en compte de τ , ou avec le nouveau pseudo-code de l'Algorithme 6. Avec cette nouvelle définition, notre méthode est maintenant adaptée à la recherche d'une fonction de Lyapunov qui assure la stabilité exponentielle de notre système.

3.5.2 Stabilité entrée - état

3.5.2.1 Rappels théoriques

Considérons le système donné par (3.26) :

$$\dot{x} = f(x, \Gamma) \quad (3.26)$$

avec Γ un signal exogène et où $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ est une application localement lipschitzienne d'un domaine $x \in \mathbb{X} \subset \mathbb{R}^n$, $\Gamma \in \mathbb{L} \subset \mathbb{R}^m$ et il existe au moins un point d'équilibre $x_e = 0$, pour le système autonome, c'est-à-dire $\Gamma = 0$. Soit $V : D \rightarrow \mathbb{R}$ une fonction continue telle que :

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \quad (3.27)$$

$$G(x) \leq -\alpha_3(\|x\|) + \alpha_4(\|\Gamma\|) \quad (3.28)$$

où $\alpha_1, \alpha_2, \alpha_3$ et α_4 sont \mathcal{K}^∞ , alors $x_e = 0$ est ISS. Il est important de rappeler que la fonction $h(\|x\|)$ est \mathcal{K}^∞ si h est une fonction scalaire croissante, $h(0) = 0$ et $h_\infty \rightarrow +\infty$. Ainsi, $\alpha_i(\|x\|) = c_i\|x\|^{b_i}$ sont des fonctions appropriées ($c_i > 0, b_i > 0$) selon (KHALIL, 2002).

Note : Selon (KHALIL, 2002), la stabilité entrée - état requiert la stabilité asymptotique lorsque $\Gamma = 0$.

3.5.2.2 Mise en œuvre de la méthode

Sur le plan mathématique, la structure et la définition de la fonction de Lyapunov restent inchangées. La fonction de Lyapunov V doit toujours avoir un minimum local en 0 et sa dérivée G un maximum local en 0. Par conséquent, les matrices H^V et H^G introduites dans les sections 3.2.1 et 3.2.2 demeurent identiques.

Le défi spécifique à la stabilité entrée - état réside dans la manière dont nous traitons l'entrée exogène Γ lors de la mesure de la "non-conformité" à la stabilité. Notre approche consiste à évaluer la non-conformité à la stabilité pour diverses valeurs de Γ et à optimiser notre fonction de coût en conséquence.

Avant d'examiner la stabilité entrée - état, il est important d'établir que le système est globalement asymptotiquement stable dans \mathbb{R}^n . Il n'est alors pas judicieux, pour la stabilité entrée - état, d'utiliser deux méthodes distinctes pour maximiser le domaine de stabilité garanti, comme cela a été envisagé précédemment. En effet, la stabilité globale asymptotique garantit déjà la stabilité du système dans tout l'espace. Ainsi, notre attention doit se porter sur l'évaluation de l'impact de l'entrée exogène sur la stabilité du système. Le défi consiste donc à déterminer si, en présence d'une entrée exogène, le système conserve ses propriétés de stabilité.

Nous définissons toujours \mathcal{Z} comme un ensemble de points obtenus à partir d'un hypercube dont les faces sont quadrillées afin de couvrir une surface suffisamment grande $\subset \mathbb{X} \times \mathbb{L}$.

Pour la stabilité entrée - état, le compteur P_1 est utilisé pour quantifier la mesure dans laquelle les conditions nécessaires pour cette forme de stabilité sont respectées, en prenant en compte l'effet de l'entrée exogène Γ . Plus précisément, P_1 est calculé comme le nombre de points pour lesquels $\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|)$ et $G(x) \leq -\alpha_3(\|x\|) + \alpha_4(\|\Gamma\|)$. En cherchant à maximiser P_1 , l'objectif est de conserver les propriétés de stabilité du système en présence d'une entrée exogène. Ainsi, la fonction de coût à minimiser a la forme suivante, où $\mathcal{Q} = -P_1$, et où Θ représente le vecteur des variables à déterminer :

$$\begin{aligned} & \min_{\Theta} \mathcal{Q} \\ & \text{subject to } (t1), (t2). \end{aligned}$$

Dans le contexte de la stabilité entrée - état, Θ englobe non seulement θ , qui représente tous les paramètres destinés à caractériser la fonction de Lyapunov, mais également les fonctions $\alpha_1, \alpha_2, \alpha_3$ et α_4 . Pour rappel, $\alpha_i(\|x\|) = c_i \|x\|^{b_i}$ avec $c_i, b_i > 0$. La méthode d'optimisation reste la même que précédemment, similaire au pseudo-code détaillé de l'[Algorithme 2](#).

3.6 Extension à d'autres problèmes

Alors que nous avons exploré en profondeur la résolution de notre problème principal et l'étude des différentes formes de stabilité, il est naturel de se demander comment les méthodologies et les solutions présentées peuvent être adaptées ou étendues à d'autres défis. Cette section vise à discuter des potentialités de l'application de nos approches à un éventail plus large de questions. En particulier, notre analyse met l'accent sur les systèmes à temps discret et sur l'instabilité.

À travers des exemples pratiques, nous illustrons par quel moyen l'intégration de nouvelles fonctionnalités à notre algorithme le rend non seulement plus robuste, mais aussi davantage flexible pour répondre à une diversité de problèmes.

3.6.1 Systèmes à temps discret

3.6.1.1 Théorie

Considérons le système autonome à temps discret donné par (3.29) :

$$x(k+1) = f(x(k)) \quad (3.29)$$

où $f : \mathbb{X} \rightarrow \mathbb{R}^n$ est une application localement lipschitzienne d'un domaine $\mathbb{X} \subset \mathbb{R}^n$ dans \mathbb{X} et il existe au moins un point d'équilibre x_e , qui est $f(x_e) = x_e$.

Théorème 35 (Théorème de stabilité de Lyapunov - temps discret) Soit $x_e = 0$ un point d'équilibre pour (3.29) et un ensemble $D \subset \mathbb{R}^n$ contenant 0 dans son intérieur. Soit $V : D \rightarrow \mathbb{R}$ une fonction continue telle que (KHALIL, 2002) :

$$V(0) = 0 \text{ et } V(x(k)) > 0 \text{ dans } D \setminus \{0\} \quad (3.30)$$

et

$$V(x(k+1)) - V(x(k)) \leq 0 \text{ dans } D \quad (3.31)$$

alors $x_e = 0$ est stable au sens de Lyapunov. De plus, si :

$$V(x(k+1)) - V(x(k)) < 0 \text{ dans } D \quad (3.32)$$

alors $x_e = 0$ est asymptotiquement stable au sens de Lyapunov.

3.6.1.2 Mise en œuvre de la méthode

Considérons le système autonome (3.29), dans lequel nous supposons que le point d'équilibre qui nous intéresse pour l'analyse de stabilité est le point 0. En conséquence, nous établissons l'équation suivante :

$$f(0) = 0. \quad (3.33)$$

De plus, prenons $V(x(k))$ la fonction de Lyapunov pour le cas des systèmes à temps discret et :

$$G(x(k)) = V(x(k+1)) - V(x(k)) \quad (3.34)$$

avec $x(k) = [x_1(k), \dots, x_n(k)]^T$.

Dans le cas du temps discret, les conditions (v1) à (v3) et (g1) à (g3), énoncées dans la section 3.2, qui doivent justifier que $V(x(k))$ doit avoir un minimum local en 0 et $G(x(k))$ doit avoir un maximum local en 0, sont équivalentes. De plus, les conditions nécessaires (t1) à (t4) pour que le point 0 du système soit asymptotiquement stable sont toujours les mêmes que dans la section précédente.

Toutefois, soulignons une distinction fondamentale entre la mise en œuvre présentée ici et celle définie dans la section 3.4.4, qui concerne les systèmes à temps continu. En effet, alors que dans le contexte à temps continu, la "dérivée" de la fonction $V(x)$, à savoir $G(x)$, possède une définition spécifique, cette définition varie lorsqu'il s'agit du temps discret.

Cette nuance nous conduit à redéfinir la matrice H^G pour notre contexte actuel. Cependant, il est à noter que la matrice H^V demeure inchangée par rapport au cas à temps continu. La dérivée première de $G(x(k))$ peut être calculée à partir de (3.34) :

$$G_l = \frac{\partial G}{\partial x_l} = \sum_{h=1}^n \frac{\partial V}{\partial x_h} \frac{\partial f_h}{\partial x_l} - \frac{\partial V}{\partial x_l}. \quad (3.35)$$

De même, la dérivée seconde de $G(x(k))$ est donnée par :

$$\begin{aligned} G_{lp} &= \frac{\partial^2 G}{\partial x_l \partial x_p} \\ &= \sum_{h=1}^n \frac{\partial^2 V}{\partial x_h \partial x_p} \frac{\partial f_h}{\partial x_l} + \sum_{h=1}^n \frac{\partial V}{\partial x_h} \frac{\partial^2 f_h}{\partial x_l \partial x_p} \\ &\quad - \frac{\partial^2 V}{\partial x_l \partial x_p} \end{aligned} \quad (3.36)$$

avec $l = 1, \dots, n$ et $p = 1, \dots, n$.

Les équations (3.35) et (3.36), en tenant compte de la condition (v2), résultent en :

$$G_l(0) = 0 \text{ pour tout } l = 1, 2, \dots, n \quad (3.37)$$

et

$$\begin{aligned} G_{lp}(0) &= \sum_{h=1}^n \frac{\partial^2 V}{\partial x_h \partial x_p} \bigg|_{x(k)=0} \frac{\partial f_h}{\partial x_l} \bigg|_{x(k)=0} \\ &\quad - \frac{\partial^2 V}{\partial x_l \partial x_p} \bigg|_{x(k)=0} \end{aligned} \quad (3.38)$$

Sur la base de (3.38), nous pouvons écrire :

$$H^G = [G_{lp}(x(k) = 0)] = H^V J - H^V \quad (3.39)$$

car H^V est symétrique. H^G est une matrice dont les éléments sont $G_{lp}(0)$ et J est la jacobienne de $f(x(k))$ en $x(k) = 0$.

Pour conclure, le point $x(k) = 0$ est asymptotiquement stable si les conditions (t1) à (t4) sont satisfaites avec la nouvelle définition de la matrice H^G (3.39). La méthode d'optimisation reste la même que dans la section 3.4.4, similaire aux pseudo-codes détaillés de l'Algorithme 2 ou de l'Algorithme 3.

$$u_q = w_{q1}f_1(x) + \dots + w_{qn}f_n(x) + h_q$$

alors :

$$\begin{aligned} \frac{\partial V(f(x))}{\partial x_j} &= a_1 \left[w_{11} \frac{\partial f_1}{\partial x_j} + \dots + w_{1n} \frac{\partial f_n}{\partial x_j} \right] \sigma'(u_1) + \dots \\ &+ a_K \left[w_{K1} \frac{\partial f_1}{\partial x_j} + \dots + w_{Kn} \frac{\partial f_n}{\partial x_j} \right] \sigma'(u_K). \end{aligned}$$

Nous reformulons comme suit :

$$\begin{aligned} \frac{\partial V(f(x))}{\partial x_j} &= \frac{\partial f_1}{\partial x_j} \Big|_{x=0} [a_1 w_{11} \sigma'(u_1) + \dots + a_K w_{K1} \sigma'(u_K)] + \dots \\ &+ \frac{\partial f_n}{\partial x_j} \Big|_{x=0} [a_1 w_{1n} \sigma'(u_1) + \dots + a_K w_{Kn} \sigma'(u_K)] \end{aligned}$$

soit :

$$\frac{\partial V(f(x))}{\partial x_j} = \sum_{k=1}^n \left(\frac{\partial f_k}{\partial x_j} \left[\sum_{q=1}^K a_q w_{qk} \sigma'(u_q) \right] \right)$$

avec $u_q = w_{q1}f_1(x) + \dots + w_{qn}f_n(x) + h_q$.

Par la suite, nous avons :

$$\frac{\partial V(f(x))}{\partial x_j} \Big|_{x=0} = \sum_{k=1}^n \left(\frac{\partial f_k}{\partial x_j} \Big|_{x=0} \left[\sum_{q=1}^K a_q w_{qk} \sigma'(h_q) \right] \right) = 0.$$

De la même manière, la dérivée première de $V(x)$ est donnée par l'équation suivante :

$$\begin{aligned} \frac{\partial V}{\partial x_j} &= \sum_{q=1}^K a_q w_{qj} \sigma'(w_{q1}x_1 + \dots + w_{qn}x_n + h_q) \\ &= \sum_{q=1}^K a_q w_{qj} \sigma'(v_q) \end{aligned}$$

et :

$$\frac{\partial V}{\partial x_j} \Big|_{x=0} = \sum_{q=1}^K a_q w_{qj} \sigma'(h_q).$$

Au final, nous souhaitons avoir :

$$\frac{\partial G}{\partial x_j} \Big|_{x=0} = \frac{\partial V(f(x))}{\partial x_j} \Big|_{x=0} - \frac{\partial V}{\partial x_j} \Big|_{x=0} = 0.$$

Il s'ensuit que :

$$\begin{aligned} \frac{\partial^2 G}{\partial x_j \partial x_i} &= \sum_{k=1}^n \frac{\partial^2 f_k}{\partial x_j \partial x_i} \left[\sum_{q=1}^K a_q w_{qk} \sigma'(u_q) \right] \\ &+ \sum_{k=1}^n \frac{\partial f_k}{\partial x_j} \left(\frac{\partial f_1}{\partial x_i} (a_1 w_{1k} w_{11} \sigma''(u_1) + \dots + a_K w_{Kk} w_{K1} \sigma''(u_K)) \right. \\ &\quad + \dots + \\ &\quad \left. \frac{\partial f_n}{\partial x_i} (a_1 w_{1k} w_{1n} \sigma''(u_1) + \dots + a_K w_{Kk} w_{Kn} \sigma''(u_K)) \right) \\ &- \frac{\partial^2 V}{\partial x_j \partial x_i}. \end{aligned}$$

Nous aboutissons donc à :

$$\begin{aligned} \frac{\partial^2 G}{\partial x_j \partial x_i} &= \sum_{k=1}^n \frac{\partial^2 f_k}{\partial x_j \partial x_i} \left[\sum_{q=1}^K a_q w_{qk} \sigma'(u_q) \right] \\ &+ \sum_{k=1}^n \frac{\partial f_k}{\partial x_j} \left(\sum_{l=1}^n \frac{\partial f_l}{\partial x_i} \left(\sum_{q=1}^K a_q w_{qk} w_{ql} \sigma''(u_q) \right) \right) \\ &- \frac{\partial^2 V}{\partial x_j \partial x_i}. \end{aligned}$$

La dérivée seconde de $G(x)$ est donnée par :

$$\begin{aligned} \frac{\partial^2 G}{\partial x_j \partial x_i} \Big|_{x=0} &= \sum_{k=1}^n \frac{\partial f_k}{\partial x_j} \Big|_{x=0} \left(\sum_{l=1}^n \frac{\partial f_l}{\partial x_i} \Big|_{x=0} \left(\sum_{q=1}^K a_q w_{qk} w_{ql} \sigma''(h_q) \right) \right) \\ &- \sum_{q=1}^K a_q w_{qj} w_{qi} \sigma''(h_q). \end{aligned}$$

Au final, dans le cas discret, la fonction $V(x)$ cherchée doit être telle que :

- (t1) $\sum_{i=1}^K a_i \sigma(h_i) + \beta = 0$,
- (t2) $\sum_{q=1}^K a_q w_{qj} \sigma'(h_q) = 0$ pour $j = 1, \dots, n$,
- (t3) $H^V > 0$, $H_{ij}^V = \sum_{q=1}^K a_q w_{qj} w_{qi} \sigma''(h_q)$,
- (t4) $H^G < 0$, $H_{ij}^G = -H_{ij}^V + \sum_{k=1}^n \frac{\partial f_k}{\partial x_j} \Big|_{x=0} \left(\sum_{l=1}^n \frac{\partial f_l}{\partial x_i} \Big|_{x=0} \left(\sum_{q=1}^K a_q w_{qk} w_{ql} \sigma''(h_q) \right) \right)$.

Ces travaux sur cette thématique ont abouti à la rédaction d'un article (BOCQUILLON et al., 2020a).

3.6.2 Instabilité

Dans cette section, nous adaptons le schéma d'optimisation proposée dans la section 3.4 à un algorithme pour étudier l'instabilité d'un point d'équilibre.

3.6.2.1 Rappels théoriques

Considérons le système donné par (3.41) :

$$\dot{x} = f(x). \quad (3.41)$$

Soit $x_e = 0$ un point d'équilibre pour (3.41). Soit $V : U \rightarrow \mathbb{R}$ une fonction continue telle que :

$$V(0) = 0 \text{ et } V(x) > 0 \text{ dans } U \setminus \{0\} \quad (3.42)$$

$$\dot{V}(x) > 0 \text{ dans } U \setminus \{0\} \quad (3.43)$$

alors $x_e = 0$ est instable, et V est alors une fonction de Chetaev.

3.6.2.2 Mise en œuvre de la méthode

Le but de cette section est de trouver un point d'équilibre qui satisfait les conditions énoncées dans les rappels théoriques, c'est-à-dire que le point est instable. Nous ajoutons cette section à notre développement car, comme décrit dans la section 1.5.8, il est difficile de trouver une fonction de Lyapunov associée à un système. S'il nous est impossible de trouver cette dite fonction de Lyapunov, cela peut être pour deux raisons :

- La première est que notre algorithme n'est pas assez performant pour trouver cette fonction de Lyapunov ainsi que son domaine de stabilité associé, mais cette fonction existe bien pour autant.
- La deuxième est que le point n'est tout simplement pas stable, donc instable. Dans ce cas, il n'existe pas de fonction de Lyapunov.

Dans cette section, nous supposons qu'une structure appropriée pour V_θ a été choisie, où tous les paramètres qui visent à caractériser la fonction de Chetaev sont rassemblés dans les Θ .

Afin de définir un point instable, il est nécessaire d'identifier un sous-ensemble où $V(x)$ et $\dot{V}(x)$ ont le même signe par définition de (3.42) et (3.43).

Afin de déterminer une fonction de Chetaev appropriée, nous introduisons de nouvelles conditions désignées par (c1) à (c4) :

(c1) $V(0) = 0$,

(c2) $G(0) = 0$,

(c3) H^V est définie positive,

(c4) H^G est définie positive.

Nous pouvons utiliser la même méthode que la section 3.4.4 avec une fonction de coût similaire. À l'aide de notre méthode décrite précédemment, nous pouvons exprimer ce problème sous la forme générale d'un problème d'optimisation dans lequel la fonction de coût \mathcal{J} doit être minimisée.

Dans ce but, notons $\lambda_i^v, i = 1, \dots, n_v$ l'ensemble des n_v valeurs propres de H^V et $\lambda_i^g, i = 1, \dots, n_g$ l'ensemble des n_g valeurs propres de H^G . Afin que les matrices H^V et H^G soient positives et afin de répondre à la minimisation de la fonction de coût, nous remplaçons (λ^v) par $(-\lambda^v)$ et (λ^g) par $(-\lambda^g)$. Nous posons :

$$\bar{\lambda}^v = \max(\text{real}(-\lambda^v))$$

$$\bar{\lambda}^g = \max(\text{real}(-\lambda^g))$$

$$\lambda_{\max} = \max(\bar{\lambda}^v, \bar{\lambda}^g).$$

Contrairement aux sections précédentes où deux méthodes distinctes étaient employées pour maximiser le domaine de stabilité garanti, ici, une telle distinction n'est pas pertinente. En effet, notre objectif dans ce contexte n'est pas de maximiser un domaine, mais plutôt de déterminer si le point d'équilibre est instable. Ainsi, une méthode unique est utilisée, comme illustré par l'[Algorithme 7](#).

Algorithme 7 : Pseudo-code pour la détermination d'un point d'équilibre instable

Input : Paramètres du modèle de la fonction de Lyapunov Θ

Result : État Instable

$$\lambda_{\max} = \max(\bar{\lambda}^v, \bar{\lambda}^g)$$

while $\lambda_{\max} \geq 0$ **do**

 | **return** λ_{\max}

end

return *Instable*

Selon ce pseudo-code, un candidat est une fonction de Chetaev si $\lambda_{\max} < 0$. C'est-à-dire que les conditions (c3) et (c4) sont vérifiées. Les conditions (c1) et (c2) définissent les propriétés spécifiques que nous souhaitons obtenir pour la fonction de Chetaev. Elles sont intégrées dans l'algorithme d'optimisation, comme contraintes d'égalité. Il est alors garanti que les solutions obtenues respectent les conditions (t1) à (t4).

Dans cette section, nous adoptons une approche méthodologique différente. Contrairement aux tentatives d'estimation d'un DOA ou de maximisation d'un "domaine d'instabilité" garanti, notre objectif ici est plus direct et simplifié. Comme illustré par l'[Algorithme 7](#), nous nous concentrons uniquement sur la détermination de l'instabilité d'un point d'équilibre donné. En effet, si nous observons que les valeurs propres deviennent négatives, nous pouvons conclure avec certitude que le point est instable. Cette méthode cible alors spécifiquement la question de savoir si le point d'équilibre analysé est instable, sans chercher à évaluer ou à maximiser un domaine spécifique.

3.7 Méthode générique

Les travaux de recherche antérieurs dans le domaine des fonctions de Lyapunov ont souvent traité des systèmes spécifiques avec des fonctions prédéfinies. Cette spécificité peut rendre difficile la généralisation et l'application à d'autres types de systèmes et à différentes formes de stabilité. Cependant, après analyse des sections précédentes, il ressort que la généralité de notre méthode est un de ses points forts.

Dans un premier temps, la méthode proposée ne se limite pas à une forme particulière de la fonction de Lyapunov. Elle offre une flexibilité importante : au lieu d'imposer une forme fonctionnelle, elle offre la liberté de choisir celle qui convient le mieux au système étudié. Cette liberté est essentielle car elle reconnaît et respecte la diversité et la complexité des systèmes dynamiques.

Par exemple, pour un système relativement simple, l'utilisation d'une fonction de Lyapunov quadratique ou polynomiale peut être suffisante, et présente l'avantage d'une efficacité algorithmique. Cependant, les systèmes industriels, souvent marqués par leur complexité et leurs interconnexions, peuvent nécessiter des fonctions de Lyapunov plus élaborées, telles que celles modélisées par un réseau de neurones. Il est également envisageable d'utiliser des fonctions de Lyapunov sous la forme d'une fonction de base radiale, sous la forme d'une fonction exponentielle, ou encore sous la forme d'une forme continue et affine par morceaux, parmi d'autres configurations possibles. Ces alternatives offrent une flexibilité accrue pour adapter la fonction de Lyapunov aux spécificités et aux besoins de chaque système étudié.

En outre, la généralité de cette méthode ne s'arrête pas à la sélection de la forme de la fonction de Lyapunov. Elle s'étend également à l'analyse de divers types de systèmes, qu'ils soient linéaires, non linéaires, continus ou discrets. De plus, pour passer d'une forme de stabilité à une autre, seuls quelques paramètres supplémentaires nécessitent d'être ajustés dans l'algorithme. Cette adaptabilité renforce l'atout de la méthode comme outil polyvalent pour étudier la stabilité dans une variété de contextes.

Enfin, cette approche générique invite à une collaboration interdisciplinaire. L'utilisateur n'est pas contraint par des choix algorithmiques ou mathématiques prédéfinis, permettant à des experts de différents domaines d'apporter leurs perspectives et leurs compétences. Il s'agit donc d'une méthode ouverte, adaptable et prête à accueillir les innovations futures.

Ainsi, dans cette section, nous proposons de généraliser les différentes méthodes notées précédemment par un cadre générique pour déterminer une fonction de Lyapunov, en abordant toujours la stabilité au sens de Lyapunov, la stabilité asymptotique, la stabilité exponentielle et la stabilité entrée-état.

Dans la suite de cette section, les notations suivantes sont utilisées :

- $G(x) = \dot{V}(x)$ dans le cas du système à temps continu $\dot{x} = f(x)$.
- $G(x) = \Delta V(x)$ dans le cas du système à temps discret $x(k+1) = f(x(k))$.

Sur la base de ces travaux, nous avons rédigé un article ([BOCQUILLON et al., 2022](#)).

3.7.1 Rappels théoriques

Dans la suite, nous supposons que $x_e = 0$ est un point d'équilibre. Selon le [Chapitre 1](#), tous les problèmes de stabilité (de la stabilité au sens de Lyapunov au cas ISS) recherchent $V(x) > 0$ et $G(x) < 0$ (≤ 0 pour la stabilité au sens de Lyapunov). Les problèmes de stabilité sont simplement différents en ce qui concerne les contraintes sur les valeurs de $V(x)$ et $G(x)$.

Ainsi, pour aborder tout problème de stabilité, deux conditions sont nécessaires pour la fonction de Lyapunov recherchée $V(x)$:

- Dans Ω_c , $V(x)$ a un minimum local en 0 et $V(0) = 0$.
- Dans Ω_c , $G(x)$ a un maximum local en 0 et $G(0) = 0$.

Ω_c est le domaine de stabilité garanti, que nous cherchons à maximiser. Les conditions suffisantes pour que $V(x)$ ait un minimum local en 0 et que $G(x)$ ait un maximum local en 0 sont toujours les mêmes. Le calcul de H^V est le même pour les systèmes à temps continu et à temps discret. Cependant, selon le [Chapitre 1](#), H^G dépend du cas considéré. Dans les deux sections suivantes, une formule générique pour H^V et H^G est calculée.

Par la suite, nous supposons que l'utilisateur a choisi une structure pour $V(x)$ qui est deux fois différentiable (forme polynomiale, forme de réseau de neurones, forme quadratique, etc.).

De plus, nous notons que $x = [x_1, \dots, x_n]^T$ et que la matrice jacobienne de f représente :

$$F^J := \{J_{qr}, q = 1, \dots, n \text{ et } r = 1, \dots, n.\} \quad (3.44)$$

où :

$$J_{qr} = \left. \frac{\partial f_q}{\partial x_r} \right|_{x=0} \quad (3.45)$$

3.7.2 Expression de H^V

La dérivée seconde de V peut être obtenue de manière directe :

$$H^V := \{V_{qr}, q = 1, \dots, n \text{ et } r = 1, \dots, n.\} \quad (3.46)$$

où :

$$V_{qr} = \left. \frac{\partial^2 V}{\partial x_q \partial x_r} \right|_{x=0}. \quad (3.47)$$

3.7.3 Expression de H^G

En raison de la définition de G , nous devons considérer les deux cas suivants.

3.7.3.1 Cas du temps continu

Par définition, nous exprimons $G(x)$ par :

$$G(x) = \frac{dV}{dt} = \sum_{h=1}^n \frac{\partial V}{\partial x_h} f_h(x). \quad (3.48)$$

La dérivée seconde de G est la suivante :

$$\begin{aligned} \frac{\partial^2 G}{\partial x_l \partial x_p} &= \sum_{h=1}^n \frac{\partial^3 V}{\partial x_h \partial x_l \partial x_p} f_h(x) \\ &+ \sum_{h=1}^n \frac{\partial^2 V}{\partial x_h \partial x_l} \frac{\partial f_h}{\partial x_p} + \sum_{h=1}^n \left(\frac{\partial^2 V}{\partial x_h \partial x_p} \frac{\partial f_h}{\partial x_l} \right) \\ &+ \sum_{h=1}^n \frac{\partial V}{\partial x_h} \frac{\partial^2 f_h}{\partial x_l \partial x_p} \end{aligned} \quad (3.49)$$

avec $l = 1, \dots, n$ et $p = 1, \dots, n$.

Nous indiquons par :

$$H^G := \{G_{lp}, l = 1, \dots, n \text{ et } p = 1, \dots, n.\} \quad (3.50)$$

où :

$$G_{lp} = \left. \frac{\partial^2 G}{\partial x_l \partial x_p} \right|_{x=0} \quad (3.51)$$

En tenant compte de (v2) et $f(0) = 0$, (3.49) résulte en :

$$G_{lp} = \sum_{h=1}^n V_{hl} J_{hp} + \sum_{h=1}^n V_{hp} J_{hl}. \quad (3.52)$$

3.7.3.2 Cas du temps discret

Il est important de noter que la variable temporelle k est omise lors des prochaines équations pour des raisons de lisibilité. Par définition, $G(x)$ est exprimée comme suit :

$$G(x) = \Delta V(x) = V(f(x)) - V(x). \quad (3.53)$$

La dérivée seconde de G est donnée, pour $l = 1, \dots, n$ et $p = 1, \dots, n$, par :

$$\begin{aligned} \frac{\partial^2 G}{\partial x_l \partial x_p} &= \sum_{h=1}^n \left(\frac{\partial f_h}{\partial x_l} \sum_{k=1}^n \frac{\partial f_k}{\partial x_p} \left[\left. \frac{\partial^2 V}{\partial x_k \partial x_p} \right|_{f(x)} \right] \right. \\ &\left. + \frac{\partial^2 f_h}{\partial x_l \partial x_p} \left[\left. \frac{\partial V}{\partial x_h} \right|_{f(x)} \right] \right) - \frac{\partial^2 V}{\partial x_l \partial x_p}. \end{aligned} \quad (3.54)$$

En tenant compte de (v2) et $f(x_e) = x_e$, (3.54) résulte en :

$$G_{lp} = \sum_{h=1}^n J_{hl} \sum_{k=1}^n J_{kp} V_{kp} - V_{lp}. \quad (3.55)$$

3.7.4 Application aux différents problèmes de stabilité

Considérons \mathcal{Z} un ensemble de points obtenus à partir d'un hypercube dont les faces sont quadrillées de manière à couvrir un domaine suffisamment grand $\subset \mathbb{X}$. Nous supposons qu'une structure appropriée pour $V_\theta(x)$ a été choisie, où tous les paramètres visant à caractériser la fonction de Lyapunov sont rassemblés dans θ . Ensuite, tous les paramètres recherchés sont rassemblés dans Θ , le vecteur des variables à déterminer.

Dans le cadre de ce manuscrit, nous avons exploré deux méthodes pour l'estimation et la maximisation du domaine de stabilité estimé garanti. Comme cela a été exposé dans les sections précédentes, ces méthodes partagent une structure commune tout en étant suffisamment génériques pour s'adapter à divers problèmes de stabilité. Cette polyvalence a été illustrée à travers l'application concrète dans différents contextes. La méthode 1 se focalise sur le comptage des points au sein même du domaine D , procédant à une évaluation directe sans étape intermédiaire. Cependant, rien ne garantit que le D trouvé soit invariant. De ce fait, une vérification a posteriori s'avère alors nécessaire pour évaluer et valider la grandeur du level set associé au D obtenu. Quant à la méthode 2, celle-ci introduit un niveau supplémentaire en intégrant un processus intermédiaire pour déterminer la valeur de c_* au sein du processus d'optimisation.

Pour approfondir cette comparaison et mettre en évidence la généralité de ces approches, nous prenons la méthode 1 comme exemple. Bien que les descriptions suivantes correspondent à la première méthode, les ajustements peuvent être consultés dans les pseudo-codes pour comprendre les adaptations nécessaires de la méthode 2. Il est important de souligner que pour le problème spécifique de l'ISS, seule la méthode 1 est adaptée. Voici un résumé de la façon dont la méthode 1 s'applique aux différents problèmes de stabilité :

Pour le problème d'ISS :

- \mathcal{Z} est défini comme $\mathbb{X} \times \mathbb{L}$.
- Θ est défini comme $[\theta, c_i, b, i = 1, \dots, 4]$.

Pour le problème de la stabilité exponentielle :

- \mathcal{Z} est réduit à \mathbb{X} .
- Θ est réduit à $[\theta, c_i, b, i = 1, \dots, 3]$.
- Des contraintes supplémentaires relatives au taux de convergence de $\|x\|$ peuvent être appliquées conformément à la section 3.5.1.

Pour le problème de la stabilité asymptotique :

- \mathcal{Z} est réduit à \mathbb{X} .
- Θ est réduit à θ .

Pour le problème de la stabilité de Lyapunov :

- \mathcal{Z} est réduit à \mathbb{X} .
- Θ est réduit à θ .
- $ratio_{dv}$ est le nombre de points $X \subset \mathcal{Z}$ où $G(x) > 0$ est évalué.

3.8 Récapitulatif et perspectives

Face aux différents enjeux identifiés lors de l'état de l'art, nous avons adopté une nouvelle méthode reposant sur un schéma d'optimisation sous contraintes pour déterminer une fonction de Lyapunov. Bien que cette stratégie soit initialement axée sur les réseaux de neurones, elle révèle une portée bien plus vaste. En réalité, il semble judicieux d'autoriser l'adoption de n'importe quelle forme pour les fonctions de Lyapunov, du moment qu'elles demeurent différentiables. Par exemple, dans le cas de systèmes moins complexes, une fonction de Lyapunov quadratique peut s'avérer judicieuse pour optimiser le temps de calcul. En revanche, face à des systèmes industriels de grande complexité, l'utilisation de fonctions de Lyapunov élaborées, à l'instar de celles modelées par un réseau de neurones, devient impérative.

À travers ce chapitre, nous avons mis en lumière une méthodologie généralisée, adaptée tant aux systèmes à temps continu qu'aux systèmes à temps discret. Non seulement cette approche permet d'identifier une fonction de Lyapunov représentée par n'importe quelle fonction différentiable, mais elle s'étend également pour traiter un large éventail de problèmes de stabilité. Nous avons couvert la stabilité au sens de Lyapunov, la stabilité asymptotique, la stabilité exponentielle, la stabilité entrée-état. Nous avons également adapté la méthode pour aborder des points d'équilibre instables. Ainsi, cette méthodologie se révèle être un outil précieux, polyvalent, et adaptatif pour l'analyse de la stabilité des systèmes dynamiques.

Dans le chapitre suivant, nous mettons concrètement en œuvre notre méthode d'analyse de la stabilité. Nous explorons en détail les applications sur des systèmes à temps continu ainsi que sur des systèmes à temps discret. Cette mise en pratique nous permet de valider empiriquement les principes théoriques que nous avons développés.

Pour évaluer de manière approfondie la pertinence et l'efficacité de notre méthode, il est essentiel de mener des tests de performance complets. Ces tests ne se limitent pas à la comparaison avec d'autres approches existantes, mais incluent une analyse entre nos différentes méthodes. Nous débutons par une étude comparative détaillée. Celle-ci est axée sur la comparaison de différentes formes de fonctions de Lyapunov. Cette étude offre un aperçu clair et approfondi des avantages et des limites propres à chaque forme. Par la suite, nous effectuons un test de performance. Ce test a pour objectif de confronter notre méthode avec certaines évoquées dans la littérature actuelle. Elle nous permet de positionner notre approche dans le contexte plus large des travaux existants. Dans le cadre de ce même test, nous comparons successivement l'efficacité de l'algorithme génétique sous contraintes puis de l'algorithme génétique sans contrainte externe et enfin de l'optimisation par essaim particulière avec certaines techniques documentées dans la littérature existante.

Au-delà de ces analyses, nous consacrons une section entière aux perspectives de notre recherche. Actuellement, nous travaillons sur l'application de notre méthode à un système industriel spécifique : un système par impédance adaptatif. Ce système est alors introduit, mettant en lumière les défis uniques qu'il pose. Bien que ce travail soit toujours en cours, nous partageons également les problématiques rencontrées ainsi que les solutions envisagées. Cette discussion offre une vision claire des directions futures de notre recherche et des innovations potentielles dans le domaine de l'analyse de la stabilité.

Études détaillées et perspectives

Études détaillées et perspectives

Sommaire

4.1	Introduction	119
4.2	Validations et applications pratiques	120
4.2.1	Stabilité asymptotique	120
4.2.1.1	Équation du système	120
4.2.1.2	Choix de la fonction de Lyapunov	121
4.2.1.3	Choix de l'algorithme d'optimisation	121
4.2.1.4	Paramètres de l'algorithme d'optimisation	121
4.2.1.5	Illustrations	122
4.2.1.6	Performances	122
4.2.2	Instabilité	124
4.2.2.1	Équation du système	124
4.2.2.2	Choix de la fonction de Chetaev	124
4.2.2.3	Choix de l'algorithme d'optimisation	125
4.2.2.4	Paramètres de l'algorithme d'optimisation	125
4.2.2.5	Illustrations	125
4.2.3	Systèmes à temps discret	125
4.2.3.1	Équation du système	126
4.2.3.2	Choix de la fonction de Lyapunov	126
4.2.3.3	Choix de l'algorithme d'optimisation	126
4.2.3.4	Paramètres de l'algorithme d'optimisation	127
4.2.3.5	Illustrations	127
4.2.3.6	Performances	128
4.2.4	Stabilité exponentielle	128
4.2.4.1	Illustrations	129
4.2.5	Stabilité entrée - état	129
4.2.5.1	Équation du système	130
4.2.5.2	Choix de la fonction de Lyapunov	130
4.2.5.3	Choix de l'algorithme d'optimisation	131
4.2.5.4	Paramètres de l'algorithme d'optimisation	131
4.2.5.5	Illustrations	131
4.2.5.6	Performances	132

4.3	Performance des approches	133
4.3.1	Performance de notre approche	133
4.3.1.1	Fonction de coût	133
4.3.1.2	Système étudié	134
4.3.1.3	Évaluation des performances	134
4.3.2	Notre approche face à l'état de l'art	135
4.4	Enjeux d'une application industrielle concrète	138
4.4.1	Modélisation du système	138
4.4.2	Perspectives et collaborations	142

4.1 Introduction

Après avoir exploré en profondeur les différentes notions de stabilité et les méthodologies associées dans les chapitres précédents, il est essentiel d'évaluer et de comparer l'efficacité de ces méthodes. Ce chapitre se consacre à cette évaluation. Nous mettons en lumière les forces et faiblesses de chaque technique en les soumettant à une série de tests et d'analyses. Ces évaluations permettent non seulement de déterminer la pertinence de chaque méthode dans différents contextes, mais aussi de guider les choix futurs pour des applications spécifiques.

Au fil du chapitre qui s'annonce, plusieurs scénarios sont déployés pour évaluer les méthodes d'étude de la stabilité que nous avons introduites. Ces exemples couvrent une gamme étendue de situations : de la stabilité asymptotique à la stabilité entrée - état (ISS), en passant par les systèmes à temps discret et les cas d'instabilité. Pour ces évaluations, nous utilisons les deux formes de fonctions de Lyapunov que nous avons définies dans 3.3 et faisons appel aux deux algorithmes d'optimisation mentionnés précédemment dans 2.4.2.

Dans la continuité, ce chapitre se consacre à une exploration exhaustive de nos méthodes. Tout d'abord, nous confrontons notre approche à l'état actuel de la recherche. Ensuite, nous examinons la performance de l'algorithme génétique (GA) et de l'optimisation par essaim de particules (PSO). De plus, une étude comparative sur différents systèmes est également menée, mettant en évidence les avantages et les inconvénients des différentes fonctions de Lyapunov que nous avons explorées. Cette démarche vise à éclairer la pertinence de l'utilisation d'une fonction de Lyapunov modélisée par un réseau de neurones.

En conclusion, une description détaillée d'un système industriel typique est présentée. Les défis inhérents à l'étude de la stabilité de ce système complexe sont discutés, tout comme les stratégies envisagées pour les surmonter. Cette exploration met en avant les perspectives d'avenir et les obstacles à surmonter pour une mise en application efficace de nos méthodes dans le monde industriel.

Il convient de noter que l'ensemble des tests a été réalisé sur une machine équipée d'un processeur Intel Core i5 - 8400H (2.5 GHz) avec 16 GB de RAM, garantissant ainsi une évaluation rigoureuse et fiable des méthodes étudiées.

4.2 Validations et applications pratiques

Dans cette section, nous illustrons concrètement l'efficacité et l'applicabilité de la méthode générique que nous avons détaillée précédemment dans la section 3.7. Notre objectif est de montrer la manière dont cette approche se déploie dans divers scénarios théoriques et son adaptation pour traiter différents types de problèmes de stabilité.

À travers ces illustrations théoriques, nous pouvons noter parfois le choix d'une méthode spécifique ou d'un algorithme particulier en lieu et place d'un autre. Ces choix stratégiques sont le fruit d'une évaluation minutieuse. En effet, en parallèle des exemples développés dans cette section, nous avons conduit des tests de performance qui ont été cruciaux pour affiner notre méthode et optimiser les paramètres associés. Ces tests ont non seulement guidé notre sélection d'algorithmes et de méthodes, mais ont également permis d'ajuster finement nos paramètres d'optimisation afin d'assurer les meilleures performances possibles dans chaque scénario. Les critères et métriques de performance que nous avons utilisés pour guider ces choix sont détaillés et discutés en profondeur dans la section suivante 4.3. Ainsi, chaque choix méthodologique et algorithmique est justifié et validé par des tests rigoureux, dont les résultats sont exposés et analysés dans les sections suivantes.

4.2.1 Stabilité asymptotique

Lors de l'application de notre approche générique à la stabilité asymptotique, nous avons choisi de nous concentrer sur un système non linéaire afin de démontrer à la fois l'efficacité et la polyvalence de la méthode. À titre d'exemple, nous examinons un pendule amorti. Ce système, étant bidimensionnel, nous offre une opportunité de visualiser aisément les résultats obtenus. Il est important de noter que nous nous concentrons sur un système continu.

4.2.1.1 Équation du système

Afin d'illustrer concrètement l'application de notre méthode, prenons le cas du pendule amorti décrit par le système différentiel suivant :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin(x_1) - x_2 \end{cases}$$

Les points d'équilibre du système sont déterminés en trouvant les solutions du système pour lesquelles $\dot{x}_1 = 0$ et $\dot{x}_2 = 0$. En utilisant les équations précédentes :

- Pour $\dot{x}_1 = 0$, nous obtenons directement $x_2 = 0$.
- Pour $\dot{x}_2 = 0$, en substituant $x_2 = 0$, nous obtenons $-\sin(x_1) = 0$. Les solutions de cette équation trigonométrique sont $x_1 = n\pi$, où n est un entier.

Parmi ces points d'équilibre, l'origine $x_1 = 0$ et $x_2 = 0$ correspond au pendule en position verticale basse, qui est un point d'équilibre stable car le pendule tend à retourner à cette position après une petite perturbation. De plus, tous les points $x_1 = 2n\pi$, où n est un entier non nul, sont également des points d'équilibre stables. Cela signifie que, après une perturbation, le pendule tendra à s'immobiliser en position basse, indépendamment de la direction initiale de son mouvement.

Dans le cas du pendule amorti, le point $x_1 = 0$ et $x_2 = 0$ (l'origine) est un point d'équilibre. Nous étudions la stabilité asymptotique de ce point d'équilibre, que nous savons stable.

4.2.1.2 Choix de la fonction de Lyapunov

Pour ce système, une fonction quadratique semble être un choix approprié pour la fonction de Lyapunov. Rappelons qu'une fonction de Lyapunov quadratique est typiquement de la forme :

$$V(x) = x^T P x$$

où P est une matrice définie positive. Les dérivées nécessaires pour analyser la stabilité à l'aide de cette fonction peuvent être obtenues en utilisant les équations (3.46) pour H^V et (3.55) pour H^G .

4.2.1.3 Choix de l'algorithme d'optimisation

Pour la détermination des poids optimaux associés à la fonction de Lyapunov, l'utilisation d'algorithmes d'optimisation est essentielle. Dans ce cadre, nous avons choisi le GA, un algorithme d'optimisation globale basé sur les principes de la sélection naturelle et de la génétique. Le GA est particulièrement efficace pour explorer un large espace de recherche et trouver une solution optimale ou quasi-optimale, ce qui le rend particulièrement adapté pour notre application.

Nous avons utilisé l'implémentation de GA fournie dans la boîte à outils d'Optimisation Globale de MATLAB (VENKATARAMAN, 2009). Cet outil, bien documenté et largement utilisé, offre une série de paramètres configurables pour affiner la recherche. Dans le travail de (KRISHNA et al., 2019), le GA de cette boîte à outils a été utilisé pour des problèmes similaires. Dans cet exemple, nous utilisons l'implémentation sous contraintes, comme détaillé dans l'Algorithme 2. Une analyse comparative des performances de cette implémentation sous contraintes par rapport à la version sans contrainte externe du GA est présentée dans la section 4.3, permettant d'évaluer l'efficacité de chaque approche dans le contexte d'un système plus complexe.

4.2.1.4 Paramètres de l'algorithme d'optimisation

Lors de la mise en œuvre du GA pour notre application, il est essentiel de définir précisément les paramètres pour garantir la pertinence et l'efficacité de la recherche. Les paramètres utilisés dans cette illustration sont détaillés ci-dessous :

- **Domaine de recherche, \mathcal{Z}** : Deux méthodes sont possibles pour définir le domaine de recherche, définies dans la section 3.4.3.
 - *Méthode 1 : Estimation indirecte d'un DOA* : Cette approche vise à maximiser directement le nombre de points vérifiant les conditions (t1) à (t4). Le critère P_1 utilisé, est défini comme le nombre de points pour lesquels $V(x) > 0$ et $G(x) < 0$. La maximisation de P_1 (ou, de manière équivalente, la minimisation de $-P_1$) sert à optimiser le domaine de définition de la fonction de Lyapunov, noté D . La méthode est détaillée sous la forme d'un pseudo-code, dans l'Algorithme 2. Bien que le nombre de points avec $V(x) > 0$ et $G(x) < 0$ soit maximisé, il n'est pas garanti que le level set correspondant soit le plus grand possible. Une vérification a posteriori s'avère alors nécessaire pour évaluer et valider la grandeur du level set associé au domaine obtenu, ce qui garantit la validité des résultats trouvés.
 - *Méthode 2 : Estimation directe du DOA avec des level sets* : Cette stratégie se concentre directement sur la maximisation de la taille du level set associé au DOA. Le critère P_2 compte les points pour lesquels $G(x) < 0$ et $V(x) \leq c_*$, respectant ainsi les conditions strictes du level set. P_2 est maximisé (ou, de manière équivalente, $-P_2$ est minimisé) pour obtenir une estimation d'un sous-DOA, un domaine de stabilité garanti étroitement lié à

la taille du level set. Cette méthode est détaillée sous la forme d'un pseudo-code, comme illustré dans l'Algorithme 3.

- **Nombre de variables, Θ** : Le nombre total de variables à optimiser est fixé à 4. Ceci comprend une matrice de taille $n \times n$, reflétant la dimensionnalité du système. Les paramètres associés à la fonction V sont contraints à rester dans l'intervalle $[-4; 4]$.
- **Paramètres de l'algorithme génétique** : Tous les paramètres du GA ont été laissés à leurs valeurs par défaut fournies par MATLAB. Cette décision a été prise pour tirer parti des réglages standards éprouvés du GA tout en se concentrant sur les spécificités de notre problème.

4.2.1.5 Illustrations

Les intervalles pour x_1 et x_2 sont $x_1 \in [-1; 1]$ et $x_2 \in [-1; 1]$. La stabilité de $x_e = [0; 0]$ est étudiée : la Figure 4.1 représente la fonction de Lyapunov pour ce point, et la Figure 4.2 représente la dérivée temporelle de la fonction de Lyapunov construite.

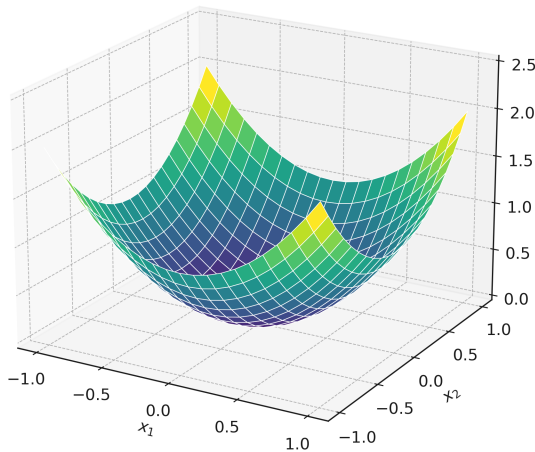


FIGURE 4.1 – Fonction de Lyapunov construite pour $x_e = [0; 0]$

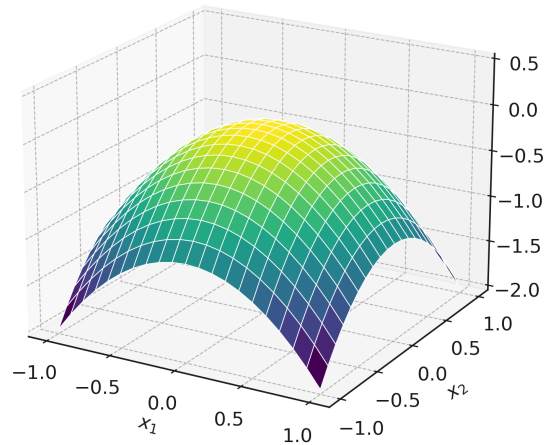


FIGURE 4.2 – Dérivée temporelle de la fonction de Lyapunov construite

Nous pouvons aisément constater que la fonction quadratique, qui décrit la dynamique du système, est une fonction de Lyapunov. Ainsi, ce point d'équilibre $x_e = [0; 0]$ du système est asymptotiquement stable. Tel que souligné à divers moments dans cette thèse, lors d'une analyse de stabilité, ce n'est pas uniquement le point d'équilibre qui est important, mais surtout le domaine de stabilité garanti associé à ce point. Dans la Figure 4.3, nous pouvons observer le plus grand level set se trouvant dans l'ensemble, qui est associé au DOA. Les points verts symbolisent les points qui sont situés dans le level set estimé, offrant ainsi une garantie de stabilité pour ces conditions initiales. La présence d'un point dans ce level set garantit sa stabilité. Cependant, l'absence d'un point dans ce level set ne confirme pas nécessairement son instabilité. Il se pourrait que ce point appartienne toujours au domaine d'attraction (DOA), mais sans la garantie de stabilité offerte par le level set.

4.2.1.6 Performances

Les illustrations et résultats présentés dans la section précédente sont issus de l'application de la méthode 2. Pour fournir un élément de comparaison, nous avons également appliqué la méthode 1

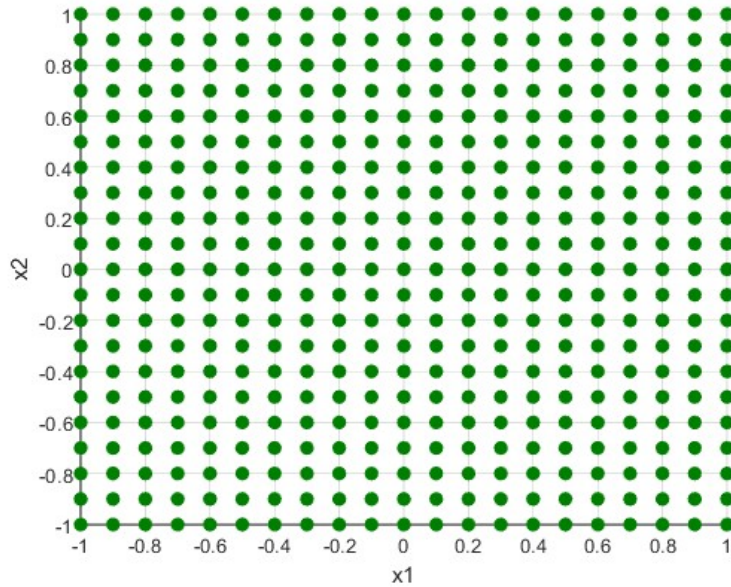


FIGURE 4.3 – Domaine de stabilité estimé garanti pour le point d'équilibre

sur le même système. Dans cette approche, la fonction de Lyapunov est représentée par un réseau de neurones (NN) défini comme suit :

$$\begin{cases} V(x) &= \sum_{j=1}^K a_j \sigma(\nu_j) + \beta \\ \nu_j &= \sum_{i=1}^n w_{ji} x_i + h_j \end{cases}$$

La Figure 4.4 et la Figure 4.5 montrent que les résultats obtenus avec la méthode 1 sont similaires à ceux précédemment observés avec la méthode 2. Comme souligné à plusieurs reprises, l'emploi d'un NN comme fonction de Lyapunov s'avère plus pertinent pour des systèmes complexes. En outre, la seconde méthode a été développée après de nombreux travaux déjà réalisés dans le cadre de cette thèse, visant à obtenir une estimation du DOA la plus précise possible. Pour cet exemple, les résultats sont donc similaires. Cependant, pour les analyses futures dans ce chapitre, nous privilégions la méthode 2.

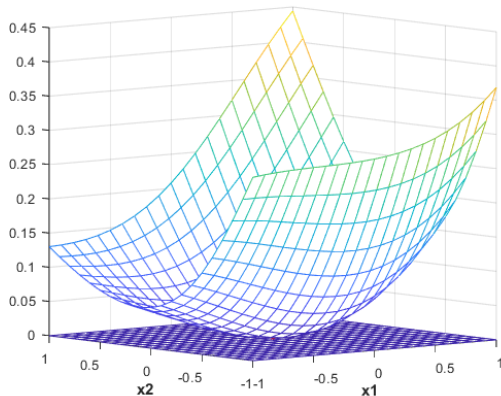


FIGURE 4.4 – Fonction neuronale de Lyapunov construite pour $x_e = [0 ; 0]$

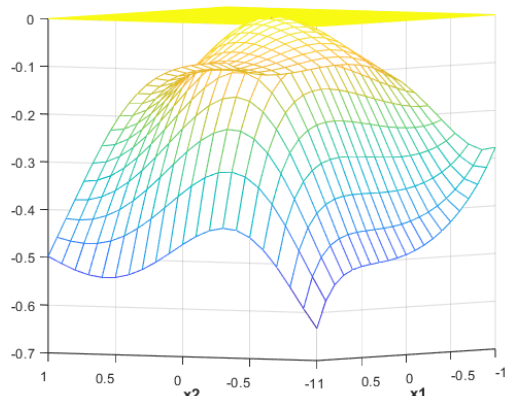


FIGURE 4.5 – Dérivée temporelle de la fonction de Lyapunov construite

Pour l'exemple présenté, la visualisation directe des figures représentant la fonction de Lyapunov et sa dérivée temporelle offre une compréhension immédiate du domaine de stabilité garanti associé.

Cependant, dans le cadre des discussions sur la stabilité exponentielle et la stabilité entrée - état (ISS), l'importance des figures représentant la fonction de Lyapunov et sa dérivée temporelle devient moins évidente. En effet, pour ces sujets spécifiques, la visualisation du domaine de stabilité garanti devient alors essentielle.

Tout en mettant en évidence la stabilité du pendule amorti en position basse, il est nécessaire de noter que ce même système présente également des points d'équilibre instables. Ces points, où le pendule ne tend pas à revenir à une position stable après une perturbation, offrent une perspective intéressante sur la dynamique du système et mettent en évidence l'importance d'analyser à la fois les comportements stables et instables.

4.2.2 Instabilité

Afin de compléter notre analyse du pendule amorti, il est important d'examiner également ses points d'équilibre instables. Ces points, tout aussi importants que les points stables, nous offrent une compréhension complète de la dynamique du système.

Contrairement à l'analyse de stabilité asymptotique, où nous recherchons une fonction de Lyapunov dont la dérivée le long des trajectoires du système est négative, l'analyse de l'instabilité requiert l'utilisation d'une fonction de Chetaev. Cette fonction, tout comme la fonction de Lyapunov, doit être positive dans la zone d'instabilité. Cependant, la particularité de la fonction de Chetaev réside dans le fait que sa dérivée le long des trajectoires du système doit également être positive.

4.2.2.1 Équation du système

Comme nous l'avons précisé précédemment, le pendule amorti est décrit par le système différentiel suivant :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin(x_1) - x_2 \end{cases}$$

Pour identifier les points d'équilibre instables, nous utilisons les mêmes équations que précédemment. Les solutions de l'équation $-\sin(x_1) = 0$ sont $x_1 = (2n + 1)\pi$, où n est un entier. Cela signifie que tous les points où $x_1 = (2n + 1)\pi$ et $x_2 = 0$ sont des points d'équilibre. Parmi ceux-ci, le point $x_1 = \pi$ et $x_2 = 0$ correspond à la position verticale haute du pendule. Ce point est un point d'équilibre instable car une petite perturbation entraînera la chute du pendule, celui-ci ne revenant pas à sa position initiale.

4.2.2.2 Choix de la fonction de Chetaev

Pour ce système, une fonction quadratique est toujours un choix approprié pour la fonction de Chetaev. Rappelons qu'une fonction de Chetaev quadratique est typiquement de la forme :

$$V(x) = x^T P x$$

où P est une matrice définie positive.

Les dérivées nécessaires pour analyser la stabilité à l'aide de cette fonction peuvent être obtenues en utilisant les équations (3.46) pour H^V et (3.55) pour H^G .

4.2.2.3 Choix de l’algorithme d’optimisation

Comme pour l’analyse de la stabilité asymptotique, le GA peut être utilisé pour optimiser la fonction de Chetaev associée à l’instabilité. Nous avons utilisé l’implémentation de GA, toujours sous contraintes, fournie dans la boîte à outils d’Optimisation Globale de MATLAB.

4.2.2.4 Paramètres de l’algorithme d’optimisation

Dans le contexte de l’étude de l’instabilité, plusieurs paramétrages restent similaires à ceux définis dans l’exemple précédent traitant de la stabilité asymptotique. Toutefois, il est fondamental de souligner une distinction majeure.

Alors que l’analyse de stabilité se focalise sur le domaine d’attraction entourant un point d’équilibre, l’analyse d’instabilité vise spécifiquement un point d’équilibre donné. De ce fait, le domaine de recherche \mathcal{Z} est nettement plus petit. Il se présente sous la forme d’un maillage régulier modelant un rectangle de dimensions 9×9 , centré sur le point d’équilibre $x_e = [\pi; 0]$. Ensuite, aucune métrique n’est introduite pour quantifier une région. En effet, nous nous concentrons uniquement sur la détermination de l’instabilité d’un point d’équilibre donné.

Mis à part ces modifications, les autres aspects des paramétrages, tels que le nombre de variables Θ et les configurations de l’algorithme génétique, demeurent inchangés par rapport à ce qui a été décrit dans l’analyse de stabilité.

4.2.2.5 Illustrations

Les bornes pour x_1 et x_2 sont désormais $x_1 \in [\pi - 0.1; \pi + 0.1]$ et $x_2 \in [-0.1; 0.1]$, compte tenu du décalage de x_1 de π pour centrer l’analyse sur le point d’équilibre $x_e = [\pi; 0]$. La Figure 4.6 et la Figure 4.7 montrent le résultat de cette étude.

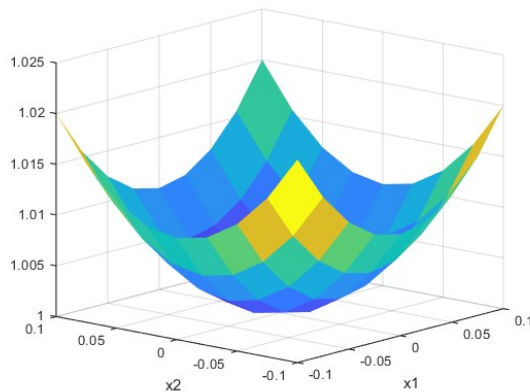


FIGURE 4.6 – Fonction de Chetaev construite pour $x_e = [\pi; 0]$

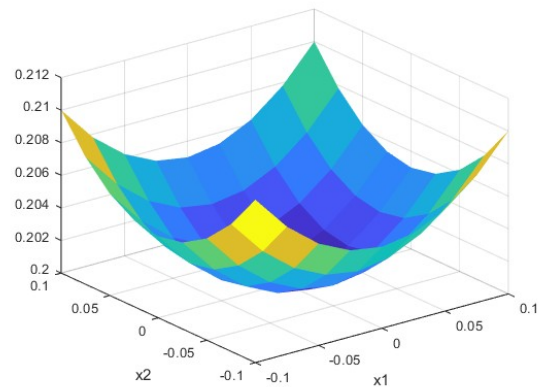


FIGURE 4.7 – Dérivée temporelle de la fonction de Chetaev construite

Il est facile de vérifier que la fonction quadratique est une fonction de Chetaev pour ce système. Par conséquent, le point d’équilibre $x_e = [\pi; 0]$ du système est instable.

4.2.3 Systèmes à temps discret

Dans la section précédente, nous avons concentré notre attention sur les systèmes continus, en particulier sur le pendule amorti. Cependant, de nombreux systèmes réels, notamment ceux impliquant

des opérations numériques ou des systèmes de commande, sont naturellement modélisés comme des systèmes à temps discret.

Le passage du temps continu au temps discret offre une nouvelle perspective sur la stabilité et la commande. Dans cette section, nous explorons la stabilité des systèmes à temps discret, en utilisant des techniques adaptées à leur nature particulière. Notre objectif est de montrer que, malgré les différences fondamentales entre ces deux classes de systèmes, notre approche générique peut être adaptée avec succès pour traiter les systèmes à temps discret.

4.2.3.1 Équation du système

Considérons le système à temps discret suivant :

$$\begin{cases} x_{1_{k+1}} &= -0.125x_{2_k} - 0.125(1 - x_{1_k}^2 - x_{2_k}^2)x_{1_k} \\ x_{2_{k+1}} &= 0.125x_{1_k} - 0.125(1 - x_{1_k}^2 - x_{2_k}^2)x_{2_k} \end{cases}$$

Le point d'équilibre de ce système est clairement $x_1 = 0$ et $x_2 = 0$, qui correspond à l'origine. Dans les sections suivantes, nous étudions la stabilité de ce point d'équilibre en utilisant notre approche générique adaptée aux systèmes à temps discret. Étant donné que ce système est en deux dimensions, il nous offre une excellente opportunité de visualiser facilement les résultats obtenus.

4.2.3.2 Choix de la fonction de Lyapunov

Pour ce système, la fonction de Lyapunov est représentée par un réseau de neurones de la forme :

$$\begin{cases} V(x) &= \sum_{j=1}^K a_j \sigma(\nu_j) + \beta \\ \nu_j &= \sum_{i=1}^n w_{ji} x_i + h_j \end{cases}$$

Nous supposons que la fonction de Lyapunov $V(x)$ est représentée par un réseau neuronal où x_i sont les entrées, w_{ji} sont les poids de la couche cachée, a_i sont les poids de la couche de sortie, h_i sont les biais de la couche cachée, β est le biais de la couche de sortie ; $i = 1, \dots, n$ et $j = 1, \dots, K$ où K est le nombre de neurones de la couche cachée.

Les dérivées nécessaires pour analyser la stabilité à l'aide de cette fonction peuvent être obtenues en utilisant les équations (3.46) pour H^V et (3.55) pour H^G .

4.2.3.3 Choix de l'algorithme d'optimisation

Pour la détermination des poids optimaux associés à la fonction de Lyapunov dans le contexte du système à temps discret, l'utilisation d'algorithmes d'optimisation reste fondamentale. Alors que le GA a été employé pour les systèmes continus, pour cette application particulière à temps discret, nous avons opté pour l'algorithme PSO.

L'efficacité du PSO dans la résolution de problèmes d'optimisation non linéaires, sa simplicité d'implémentation, et sa capacité à éviter de rester bloqué dans des minima locaux en font un choix judicieux pour notre application à temps discret. De plus, comme le GA, le PSO a également été largement utilisé pour des problèmes similaires, comme démontré dans le travail de (ROSSO et al., 2021), attestant de sa robustesse et de son efficacité dans différents domaines d'application.

Nous avons utilisé l'implémentation de PSO fournie dans la boîte à outils d'Optimisation Globale de MATLAB. Cette implémentation, bien documentée et largement utilisée, offre une gamme complète de paramètres configurables pour s'adapter à la nature spécifique de notre problème. Dans la section 4.3, une analyse comparative des performances entre le PSO et le GA est réalisée.

4.2.3.4 Paramètres de l'algorithme d'optimisation

Lors de la mise en œuvre du PSO pour notre application, il est essentiel de définir précisément les paramètres pour garantir la pertinence et l'efficacité de la recherche. Les paramètres utilisés dans cette illustration sont détaillés ci-dessous :

- **Domaine de recherche, \mathcal{Z}** : Notre domaine est représenté comme un maillage de dimension 21×21 régulier sous forme d'un rectangle, centré autour de l'origine. Ce maillage sert à identifier et compter les points qui répondent aux exigences spécifiques du level set. Pour garantir une couverture exhaustive de la dynamique du système, nous évaluons les conditions $G(x) < 0$ et $V(x) \leq c$ en 441 points distincts, couvrant l'ensemble du domaine Ω_c , à l'aide de l'Algorithme 3.
- **Structure du réseau de neurones** : Nous considérons une couche cachée et le nombre de neurones de cette couche cachée arbitrairement fixé à $K = 12$.
- **Nombre de variables, Θ** : Le nombre total de variables à optimiser est fixé à $K \times (2n) + 1 = 49$. Les paramètres associés à la fonction V sont contraints à rester dans l'intervalle $[-4; 4]$.
- **Fonction d'activation, $\sigma(\nu)$** : Dans cet exemple, nous utilisons la fonction d'activation hyperbolique tangente, c'est-à-dire $\sigma(\nu) = \tanh(\nu)$.
- **Paramètres de l'algorithme PSO** : Tous les paramètres du PSO ont été laissés à leurs valeurs par défaut fournies par MATLAB. Cette décision a été prise pour tirer parti des réglages standards éprouvés du PSO tout en se concentrant sur les spécificités de notre problème.

4.2.3.5 Illustrations

Les résultats obtenus grâce à notre approche sont mis en évidence par la Figure 4.8. Les variables x_1 et x_2 sont définies sur l'intervalle $[-1; 1]$. Les points verts dans cette figure symbolisent le domaine de stabilité garanti, indiquant les états initiaux à partir desquels le système converge vers son point d'équilibre. Les croix rouges, en revanche, représentent les états pour lesquels nous ne pouvons pas conclure quant à la convergence vers l'équilibre.

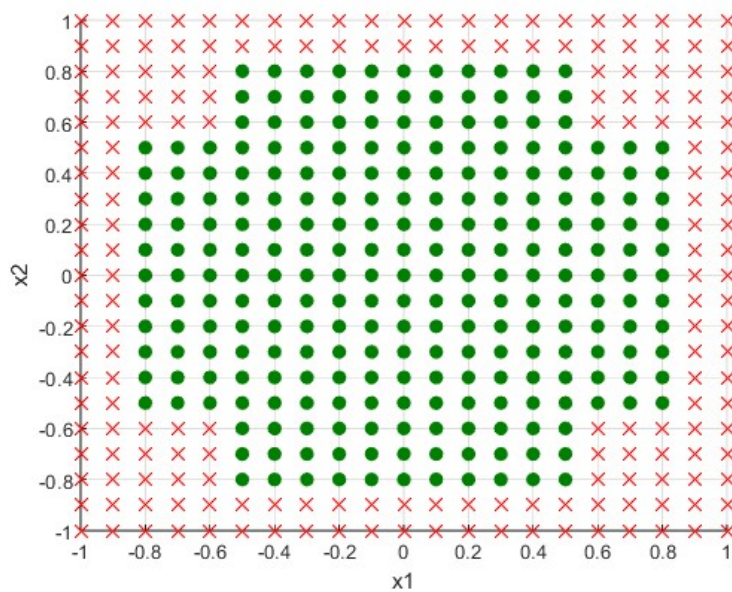


FIGURE 4.8 – Domaine de stabilité garanti estimé pour $x_e = [0; 0]$

4.2.3.6 Performances

Dans notre étude, le système en question est de nature polynomiale. Cependant, en utilisant l'outil SOSTOOLS dans MATLAB, comme discuté précédemment dans la section 2.3.3, nous avons constaté qu'aucune solution n'a été trouvée pour notre système à temps discret. Ceci pourrait s'expliquer par l'absence de référence sur l'utilisation de SOSTOOLS pour de tels systèmes, d'où l'incapacité de trouver une fonction de Lyapunov adéquate.

Pourtant, il est pertinent de comparer notre outil à celui de SOSTOOLS. Considérons un système polynomial :

$$\begin{cases} \dot{x}_1 &= -x_1^3 + x_2 \\ \dot{x}_2 &= -x_1 - x_2 \end{cases}$$

Lorsque nous utilisons SOSTOOLS, l'outil nous renvoie une fonction de Lyapunov, $V = 1.195x_1^2 + 5.801 \times 10^{-5}x_1x_2 + 1.195x_2^2$, en moins d'une seconde. Notre algorithme trouve également une fonction de Lyapunov, entre les bornes -1 et 1 pour x_1 et x_2 , en une vingtaine de secondes. Cependant, il est fondamental de souligner un aspect important : bien que SOSTOOLS fournisse une fonction de Lyapunov, elle ne fournit pas d'information sur le domaine de stabilité garanti associé au point d'équilibre. C'est précisément sur cette caractéristique particulière que notre approche se distingue, en offrant une analyse plus approfondie, bien que cela implique un temps de calcul plus important.

4.2.4 Stabilité exponentielle

La stabilité asymptotique est un concept central pour assurer la convergence des systèmes vers un point d'équilibre. Toutefois, dans de nombreux cas pratiques, il est souhaitable d'avoir une idée de la rapidité de cette convergence. C'est là que le concept de stabilité exponentielle entre en jeu. Non seulement il garantit la convergence, mais également la vitesse de convergence qui se caractérise par un taux de décroissance exponentielle.

Nous examinons une propriété importante en utilisant un système en temps continu, avec le point d'équilibre $x_e = [0 ; 0]$. Le système est défini par les équations suivantes :

$$\begin{cases} \dot{x}_1 &= -\tan(x_1) + x_2^2 \\ \dot{x}_2 &= -x_2 + x_1 \end{cases}$$

Ayant validé préalablement la stabilité asymptotique du système pour $x_1 \in [-1.5 ; 1.5]$ et $x_2 \in [-1.5 ; 1.5]$, l'étape suivante consiste à explorer sa stabilité exponentielle. Le choix de la fonction de Lyapunov et de l'algorithme d'optimisation demeure identique à celui de la section précédente. En ce qui concerne les paramètres de l'algorithme d'optimisation, ils restent majoritairement identiques, à l'exception de quelques points cruciaux :

- Les conditions qui ne satisfont pas aux critères spécifiques du level set sont maintenant évaluées selon la méthode proposée par l'Algorithme 6.
- Il est nécessaire de préciser une valeur pour τ , qui représente le taux de convergence exponentielle de $\|x(t)\|$, c'est-à-dire le temps que le point initial choisi met pour atteindre la convergence vers le point d'équilibre.

4.2.4.1 Illustrations

Figure 4.9 - Domaine de stabilité garanti pour $\tau = 9$: Dans cette illustration, les points verts représentent les états initiaux qui non seulement convergent vers l'origine, mais le font également dans un temps inférieur ou égal à 9 secondes.

Figure 4.10 - Domaine de stabilité garanti pour $\tau = 4$: Similairement, cette figure illustre les états initiaux qui convergent vers l'origine dans un temps inférieur ou égal à 4 secondes. Comme il était prévisible, le domaine d'attraction est plus restreint par rapport à la figure précédente, étant donné le temps alloué pour la convergence plus court. Le maillage utilisé dans cette représentation est beaucoup plus fin, offrant ainsi une meilleure précision dans la détermination des points appartenant au domaine de stabilité garanti.

En comparant les deux figures, il est clair que le choix de τ influence significativement le domaine de stabilité garanti. Une valeur plus élevée de τ donne un domaine de stabilité garanti plus étendu, car le système dispose de plus de temps pour converger vers l'équilibre.

Afin de prouver la validité des résultats, nous utilisons Simulink avec différents points de départ appartenant ou non au domaine de stabilité garanti illustré dans la Figure 4.11. Il est important de noter que, bien que les points sur les figures représentant le domaine de stabilité garanti soient identifiés en fonction de leur capacité à converger en moins de τ secondes, la norme $\|x\|$ présentée dans la Figure 4.11, représentant la vérification temporelle sur Simulink, peut varier. Ceci est dû au fait que $\|x\|$ est calculée à partir des valeurs combinées de x_1 et x_2 , reflétant ainsi une mesure de convergence globale qui peut différer des valeurs individuelles de x_1 ou x_2 utilisées pour marquer les points sur le graphique de stabilité. Sur la Figure 4.11, le tracé bleu ciel illustre le taux de décroissance exponentielle pour $\tau = 9$. Le tracé orange, représentant la trajectoire à partir du point initial $[1 ; 1.5]$, présente une décroissance plus lente que la décroissance exponentielle. Cette observation est corroborée par la Figure 4.9, qui montre clairement que le point initial $[1 ; 1.5]$ ne converge pas vers un état stable en moins de 9 secondes.

4.2.5 Stabilité entrée - état

Jusqu'à présent, nous avons examiné la stabilité intrinsèque des systèmes sans prendre en compte les perturbations ou les entrées externes. Cependant, dans des situations réelles, il est souvent essentiel de comprendre comment un système réagit aux entrées externes, notamment pour garantir une

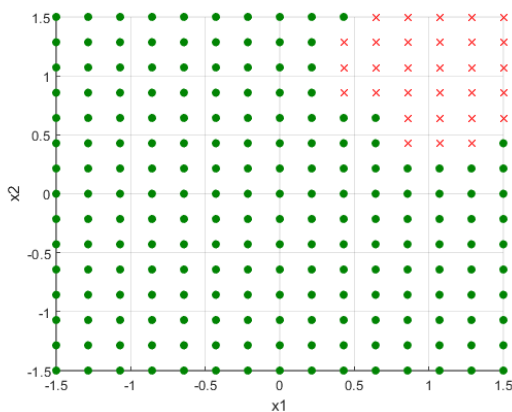


FIGURE 4.9 – Domaine de stabilité garanti associé à $\tau = 9$

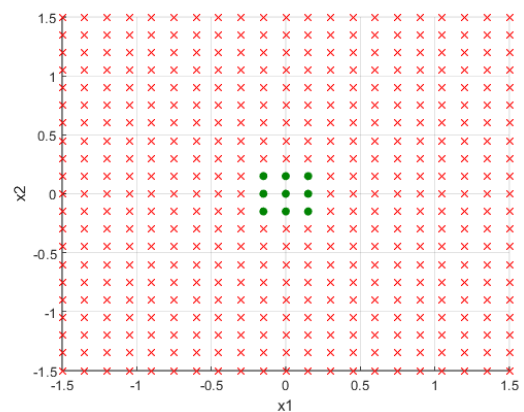


FIGURE 4.10 – Domaine de stabilité garanti associé à $\tau = 4$

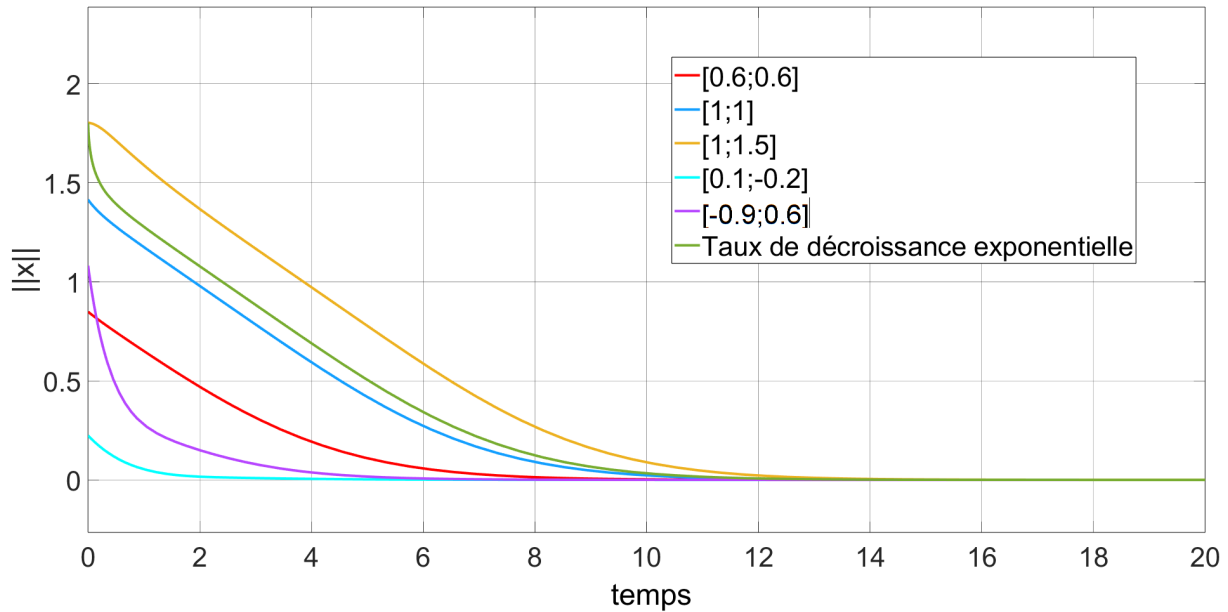


FIGURE 4.11 – Vérification temporelle sur Simulink

performance robuste en présence de perturbations. C’est dans ce contexte que le concept de stabilité entrée - état prend de l’importance. Cette notion se concentre sur la manière dont les conditions initiales du système et les entrées externes influencent la trajectoire future du système.

4.2.5.1 Équation du système

Considérons le système suivant :

$$\begin{cases} \dot{x}_1 = -x_1 + x_2^2 \\ \dot{x}_2 = -x_2 + \Gamma \end{cases}$$

Les plages de valeurs pour x_1 , x_2 et Γ sont $x_1 \in [-2; 2]$, $x_2 \in [-2; 2]$ et $\Gamma \in [0; 10]$.

Dans le cadre de l’analyse de la stabilité entrée - état, il est impératif d’examiner la stabilité globale asymptotique du système dans les plages de valeurs définies pour x_1 et x_2 , lorsque $\Gamma = 0$. Cette manipulation a été précédemment exécutée par nos soins, révélant que le système est effectivement asymptotiquement globalement stable. Cela signifie que, indépendamment de la position initiale du système dans les intervalles spécifiés, la trajectoire de l’état converge invariablement vers l’unique point d’équilibre $x_e = [0; 0]$ au fil du temps.

4.2.5.2 Choix de la fonction de Lyapunov

Pour ce système, l’utilisation d’un réseau de neurones pour la fonction de Lyapunov semble être un choix pertinent, de la forme :

$$\begin{cases} V(x) = \sum_{j=1}^K a_j \sigma(\nu_j) + \beta \\ \nu_j = \sum_{i=1}^n w_{ji} x_i + h_j \end{cases}$$

Nous supposons que la fonction de Lyapunov $V(x)$ est représentée par un réseau neuronal où x_i sont les entrées, w_{ji} sont les poids de la couche cachée, a_i sont les poids de la couche de sortie, h_i sont les biais de la couche cachée, β est le biais de la couche de sortie ; $i = 1, \dots, n$ et $j = 1, \dots, K$ où K est le nombre de neurones de la couche cachée.

Les dérivées nécessaires pour analyser la stabilité à l'aide de cette fonction peuvent être obtenues en utilisant les équations (3.46) pour H^V et (3.55) pour H^G .

4.2.5.3 Choix de l'algorithme d'optimisation

Nous avons choisi d'utiliser le PSO pour le concept de stabilité entrée - état. Cette implémentation est fournie dans la boîte à outils d'Optimisation Globale de MATLAB.

4.2.5.4 Paramètres de l'algorithme d'optimisation

Lors de la mise en œuvre du PSO pour notre application, il est essentiel de définir précisément les paramètres pour garantir la pertinence et l'efficacité de la recherche. Les paramètres utilisés dans cette illustration sont détaillés ci-dessous :

- **Domaine de recherche, \mathcal{Z}** : Notre domaine (défini comme $\mathbb{X} \times \mathbb{L}$ cette fois-ci) est représenté comme un maillage de dimension $21 \times 21 \times 21$ régulier sous forme d'un parallélépipède rectangle, centré autour de l'origine. Pour garantir une couverture exhaustive de la dynamique du système, nous évaluons les conditions $\alpha_1(\|x\|) \leq V(X) \leq \alpha_2(\|x\|)$ et $G(x) \leq -\alpha_3(\|x\|) + \alpha_4(\|\Gamma\|)$ en 9 261 points distincts, couvrant l'ensemble du domaine Ω_c . Ces conditions sont intrinsèquement liées au critère P_2 , qui comptabilise les points qui satisfont aux caractéristiques requises du level set.
- **Structure du réseau de neurones** : Nous considérons une couche cachée et le nombre de neurones de cette couche cachée arbitrairement fixé à $K = 12$.
- **Nombre de variables, Θ** : Le nombre total de variables à optimiser est fixé à $K \times (2n) + 1 + 4 \times 2 = 57$. Les paramètres associés à la fonction V sont contraints à rester dans l'intervalle $[-4; 4]$ et ceux associés aux fonctions α_i dans $[0; 1000]$.
- **Fonction d'activation, $\sigma(\nu)$** : Dans cet exemple, nous utilisons la fonction d'activation hyperbolique tangente, c'est-à-dire $\sigma(\nu) = \tanh(\nu)$.

Tous les paramètres du PSO ont été laissés à leurs valeurs par défaut fournies par MATLAB. Cette décision a été prise pour tirer parti des réglages standards éprouvés du PSO tout en se concentrant sur les spécificités de notre problème.

4.2.5.5 Illustrations

Nous pouvons alors vérifier la stabilité entrée - état. La [Figure 4.12](#) montre le domaine de stabilité garanti estimé lorsque $\Gamma \in [0; 10]$. Les points verts indiquent la région où l'origine est stable en entrée - état.

Afin de valider ces résultats, nous avons utilisé l'outil Simulink avec différents points de départ et différentes valeurs de Γ . La [Figure 4.13](#) illustre cette validation.

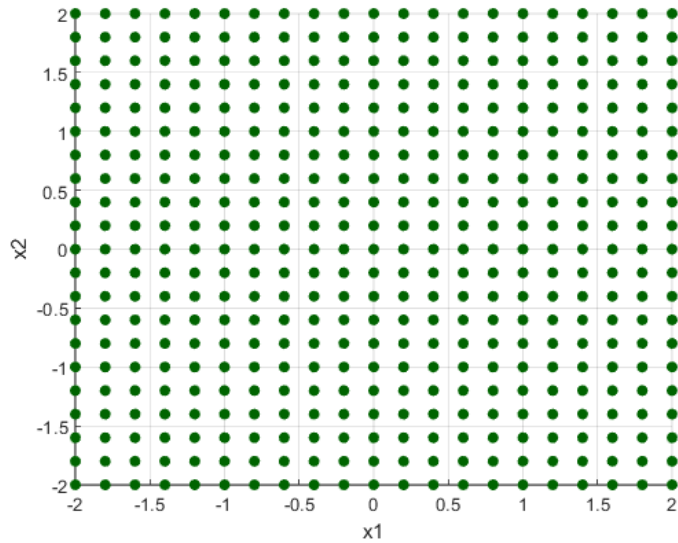


FIGURE 4.12 – Domaine de stabilité garanti estimé pour $\Gamma \in [0 ; 10]$

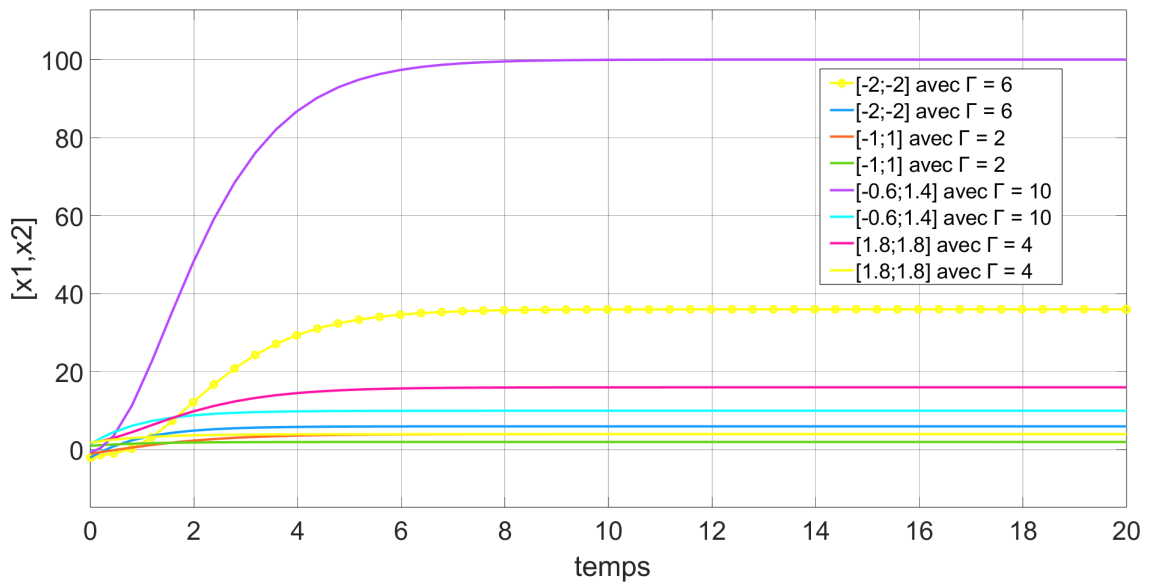


FIGURE 4.13 – Vérification temporelle sur Simulink

4.2.5.6 Performances

Soulignons que, ayant cherché un système avec une entrée externe qui serait ISS ou non en fonction de la valeur de Γ , nous n'avons pas réussi à trouver un tel système. Certaines valeurs de la vérification temporelle sur Simulink peuvent paraître importantes, mais notre objectif principal est d'estimer un sous-DOA et de maximiser le domaine de stabilité garanti. Nous cherchons à garantir la convergence pour n'importe quelle valeur initiale avec une entrée donnée. Nous mettons en avant un aspect fondamental : il faut absolument éviter que les points identifiés comme stables (c'est-à-dire les points verts) dans notre domaine de stabilité garanti ne finissent par diverger. Cette fiabilité est particulièrement cruciale dans un contexte industriel, où la garantie de stabilité est synonyme d'efficacité opérationnelle.

4.3 Performance des approches

Dans ce chapitre, nous entreprenons une évaluation approfondie et méthodique des performances de nos approches, dans le but de valider son efficacité, mais aussi de la situer précisément dans le paysage de la recherche actuelle. Les raisons d'une telle démarche analytique sont multiples. Cette évaluation permet, entre autres, d'assurer la légitimité scientifique de notre travail, d'identifier clairement ses points forts et également ses limites, et de guider les efforts futurs d'amélioration.

Nous commençons notre analyse en examinant de près les performances intrinsèques de notre approche. Étant donné notre méthode stochastique, une évaluation attentive de son efficacité est essentielle pour comprendre sa fiabilité dans différents scénarios et conditions d'utilisation. Cette évaluation initiale nous fournit une compréhension précise de la manière dont notre approche fonctionne en pratique et des résultats qu'elle est capable de produire.

Après avoir établi un aperçu clair des performances de notre approche, nous la mettons ensuite en perspective avec l'état de l'art actuel dans le domaine. Cela nous permet non seulement de situer notre contribution dans le contexte plus large de la recherche sur la stabilité des systèmes, mais aussi de comprendre de quelle manière elle se compare à d'autres méthodes et approches existantes. Cette analyse comparative est essentielle pour démontrer la valeur et l'originalité de notre contribution à la communauté scientifique.

Enfin il convient de noter, que dans le cadre de nos travaux, nous avons utilisé plusieurs types d'implémentations différentes pour notre fonction de coût : une version avec contraintes et sans contrainte externe implémentée par l'algorithme génétique, et une version sans contrainte implémentée par l'optimisation par essaim de particules. Toutefois, il est important de noter que, de base, un algorithme génétique n'opère pas sous contraintes. Dans l'implémentation de MATLAB, ces contraintes ont été ajoutées via des fonctions de pénalité. Chacun de ces algorithmes d'optimisation a ses propres caractéristiques, avantages et inconvénients, il est donc impératif d'analyser leur efficacité dans notre cas d'étude spécifique. Cette analyse nous permet d'identifier les forces et les faiblesses de chaque algorithme dans le contexte de notre méthode et de fournir un aperçu précieux sur les meilleures stratégies d'implémentation pour optimiser les performances de notre approche.

Néanmoins il est important de souligner, que durant la conduite de cette recherche, nous n'avons pas eu l'opportunité d'effectuer des études comparatives exhaustives sur des systèmes complexes. Une telle évaluation aurait assurément enrichi notre analyse en fournissant des données empiriques et des observations pertinentes sur le comportement de notre méthode dans des contextes opérationnels concrets et complexes. Toutefois, bien que cette lacune limite la portée pratique immédiate de nos résultats, les tendances que nous avons observées au cours de nos tests préliminaires sont prometteurs. Ils esquissent des perspectives encourageantes pour l'application future de notre approche à des problématiques industrielles réelles.

4.3.1 Performance de notre approche

4.3.1.1 Fonction de coût

Pour évaluer l'efficacité de notre approche, nous commençons par la base des critères de performance : la forme de la fonction de Lyapunov. Dans le contexte de la stabilité au sens de Lyapunov, la première condition essentielle est la suivante : pour qu'une fonction V soit une fonction de Lyapunov, il faut que $\lambda_{\max} < 0$. Ce critère λ_{\max} est détaillé dans les pseudo-codes de nos divers algorithmes. Si cette condition n'est pas respectée, alors la fonction V n'est pas une fonction de Lyapunov et ne peut pas être utilisée pour déduire la stabilité du système ainsi que son domaine de stabilité associé.

Afin d'affiner nos évaluations, nous rappelons le critère P_2 . Ce critère joue un rôle décisif dans l'évaluation de la qualité des fonctions de Lyapunov obtenues. Plus précisément, P_2 représente le nombre de points dans le maillage qui respecte les conditions du level set. P_2 est calculé comme le nombre de points pour lesquels $G(x) < 0$ et $V(x) \leq c$. Dans le scénario où nous utilisons un maillage de 21 points par 21 points, nous évaluons ces conditions en 441 points distincts. L'objectif optimal est donc d'obtenir $P = 441$, soit $Q = -441$ selon l'Algorithme 3.

Pour faciliter l'interprétation et l'analyse des performances, nous standardisons la valeur de la fonction de coût de manière à ce que le résultat optimal soit toujours -1 . Pour ce faire, nous définissons seulement pour cette section Q comme suit :

$$Q = -\frac{P_2}{N1 \times N2}$$

où $N1 \times N2$ représente le nombre total de points dans le maillage. Cette normalisation garantit que la fonction de coût reste cohérente et facilement interprétable, quel que soit le nombre de points du maillage. Ainsi, une valeur de Q proche de -1 indique un résultat optimal de la fonction de Lyapunov, la rendant plus intuitive à évaluer.

4.3.1.2 Système étudié

Considérons le système donné par :

$$\begin{cases} \dot{x}_1 &= -x_1 + x_2 + 0.5(\exp(x_1) - 1) \\ \dot{x}_2 &= -x_1 - x_2 + x_1x_2 + x_1 \cos x_1 \end{cases}$$

Selon (NAJAFI et al., 2016), ce système est asymptotiquement stable dans l'intervalle $[-1; 1]$ pour les variables x_1 et x_2 . Cet exemple s'est avéré être un défi majeur, mettant fréquemment notre algorithme à l'épreuve. Ces difficultés ont néanmoins été bénéfiques, car elles nous ont guidés dans l'optimisation et le perfectionnement de notre méthode tout au long de cette thèse. L'expérience acquise avec cet exemple a été fondamentale dans l'élaboration de notre méthode 2. Initialement, nous avons employé la méthode 1 pour tester le système, mais avons constaté que la vérification a posteriori du plus grand level set, lié à la maximisation du domaine de définition de la fonction de Lyapunov, D , révélait un domaine trop restreint. Ce constat a été le moteur pour développer la méthode 2.

4.3.1.3 Évaluation des performances

La nature stochastique de notre algorithme d'optimisation nous permet une évaluation statistique approfondie des résultats. Le test du système étudié est soumis à vingt exécutions successives. Pour évaluer les performances de l'algorithme, plusieurs mesures de Q sont considérées à l'issue de ces exécutions :

$$\begin{cases} Q_{min} = \min_{i=1, \dots, n \text{ runs}} Q_i & \text{(Valeur Minimale)} \\ Q_{mean} = \frac{1}{n} \sum_{i=1}^{n \text{ runs}} Q_i & \text{(Valeur Moyenne)} \\ Q_{std} = \sqrt{\frac{1}{n} \sum_{i=1}^{n \text{ runs}} (Q_i - Q_{mean})^2} & \text{(Écart-Type)} \end{cases}$$

Ainsi, pour chaque test, nous notons non seulement la valeur minimale Q_{min} , la valeur moyenne Q_{mean} et son écart-type Q_{std} , mais également le temps de calcul moyen t_{cpu} nécessaire pour effectuer ces vingt exécutions.

Rappelons que notre approche générique permet d'utiliser différentes formes de fonctions de Lyapunov. Pour des systèmes moins complexes, une fonction de Lyapunov quadratique est souvent suffisante. Cependant, pour des systèmes plus complexes ou non linéaires, une fonction de Lyapunov modélisée par un réseau de neurones peut s'avérer plus appropriée. Forts de cette conviction, nous avons décidé de tester l'efficacité de notre algorithme en utilisant ces deux formes de fonctions pour le système en question.

Les performances obtenues, que ce soit avec une fonction de Lyapunov représentée par un réseau de neurones ou sous une forme quadratique, sont détaillées dans la [Table 5.1](#).

	Q_{min}	Q_{mean}	Q_{std}	$t_{cpu}/20runs(mn)$
Forme Quadratique	-0.92	-0.71	0.28	18.76
Forme NN	-1	-0.86	0.19	52.9

TABLE 4.1 – Mesure de performance de l'algorithme pour différentes formes de la fonction de Lyapunov

Comme nous l'avions anticipé, la modélisation de la fonction de Lyapunov à l'aide d'un réseau de neurones démontre une nette supériorité, surtout lorsque la complexité du système s'accroît. Ces résultats confirment la pertinence de notre choix d'intégrer des approches basées sur les réseaux de neurones pour traiter des systèmes plus élaborés. Néanmoins, cette mesure de performance met également en lumière l'un des défis majeurs que nous avons régulièrement abordé tout au long de cette thèse : le temps de calcul. Bien que la forme d'un réseau neuronal offre des avantages indéniables en termes de précision et de robustesse, elle s'accompagne d'une augmentation significative du temps de calcul, soulignant ainsi la nécessité d'une optimisation continue et d'une recherche de compromis entre précision et efficacité calculatoire.

4.3.2 Notre approche face à l'état de l'art

Dans cette section, nous cherchons à situer notre travail par rapport aux méthodes existantes dans la littérature scientifique, en réalisant une analyse comparative. Cette approche est essentielle, non seulement pour démontrer la validité de notre méthode, mais également pour souligner sa contribution au domaine de recherche existant.

Nous procédons à une comparaison de notre approche avec une méthode choisie pour sa pertinence et sa représentativité dans le domaine. La méthode examinée dans cette section vise explicitement à maximiser le domaine de stabilité. L'article ([NAJAFI et al., 2016](#)) propose une technique d'échantillonnage rapide pour estimer en temps réel les différents sous-DOAs de systèmes non linéaires. L'approche d'échantillonnage présentée dans cet article poursuit le même objectif que les approches d'optimisation basées sur la théorie de Lyapunov : trouver le plus grand level set d'une fonction candidate de Lyapunov pour approximer le DOA. Les auteurs évaluent explicitement les conditions énoncées dans le théorème de Lyapunov pour une fonction de Lyapunov donnée par rapport à un état x_i choisi aléatoirement. Les level sets associés à l'échantillon x_i avec une dérivée positive de la fonction de Lyapunov sont écartés. L'article propose deux méthodes d'échantillonnage, l'une sans mémoire et l'autre avec mémoire, conçues pour obtenir des estimations plus précises.

Dans le cadre de nos recherches, nous avons procédé à une analyse comparative de cette méthode sur trois systèmes différents. Nous considérons les trois systèmes dynamiques suivants :

- Système 1 :

$$\begin{cases} \dot{x}_1 &= -x_2 \\ \dot{x}_2 &= x_1 - x_2 + x_1^2 x_2 \end{cases} \quad (4.1)$$

- Système 2 :

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -0.2x_2 + 0.81 \sin(x_1) \cos(x_1) - \sin(x_1) \end{cases} \quad (4.2)$$

- Système 3 :

$$\begin{cases} \dot{x}_1 &= -0.25x_1 + \ln(1 + x_2) \\ \dot{x}_2 &= -0.3750x_1 - 0.2x_1x_2 + (0.125x_1 - x_2) \cos(x_1) \end{cases} \quad (4.3)$$

Cette comparaison est fondée principalement sur le plus grand level set obtenu et le temps de convergence. Les résultats obtenus sont présentés dans la [Table 5.2](#) ci-dessous.

Méthode	Système 1 (4.1)	Système 2 (4.2)	Système 3 (4.3)
Plus grand level set trouvé			
Notre méthode GA sous contraintes	3.42	0.9	0.64
Notre méthode GA sans contrainte	3.47	0.87	0.6
Notre méthode PSO	3.37	0.92	0.57
Sampling Approach NAJAFI et al., 2016	2.318	0.708	0.278
Temps de Convergence (min)			
Notre méthode GA sous contraintes	2	4.5	5.2
Notre méthode GA sans contrainte	1	1.96	3.1
Notre méthode PSO	0.65	1.45	1.78
Sampling Approach NAJAFI et al., 2016	0.01	0.01	0.01

TABLE 4.2 – Comparaison des méthodes sur le plus grand level set et le temps de convergence

Pour cette analyse de comparaison, nous avons réalisé dix essais sur chaque système afin d'obtenir une estimation fiable des résultats. Bien que ces essais n'offrent peut-être pas une représentation optimale de la performance de chaque méthode, ils fournissent un aperçu précieux des capacités et des limites potentielles des différentes techniques.

Afin de réaliser une analyse détaillée, examinons les résultats de cette méthode de manière plus approfondie. Cette approche issue de ([NAJAFI et al., 2016](#)) est particulièrement intéressante en raison de sa vitesse de convergence, qui est manifestement beaucoup plus rapide que dans nos différentes méthodes. Cependant, cette efficacité ne doit pas masquer certaines de ses limites, selon les auteurs. Bien que l'approche par échantillonnage offre des temps de convergence impressionnants, elle souffre d'une absence de garantie formelle de convergence. Dans les scénarios où une preuve rigoureuse de la performance est nécessaire, cette absence pourrait être problématique. De plus, le choix des auteurs de s'en tenir à des fonctions de Lyapunov quadratiques comme candidates peut potentiellement limiter la capacité de cette méthode à estimer le DOA pour des systèmes non linéaires plus élaborés.

Il est pertinent de mentionner que les auteurs de l'article (NAJAFI et al., 2016) ont eux-mêmes réalisé une étude comparative en confrontant leur méthode aux méthodes d'optimisation existantes. Leur analyse a montré qu'en moyenne, ils trouvent de plus grands level sets par rapport aux techniques d'optimisation. Ainsi, en comparant notre méthode avec celle de l'article, nous pouvons indirectement évaluer notre approche par rapport aux autres techniques d'optimisation présentes dans le domaine.

En ce qui concerne notre méthode, la transition d'une implémentation sous contraintes à une implémentation sans contrainte externe s'est avérée judicieuse. En effet, l'amélioration en termes de temps calculatoire est significative, tout en conservant des résultats comparables. Cependant, il convient de noter que nos tests ont été réalisés sur des systèmes théoriques. Dans des scénarios industriels plus complexes, les résultats pourraient varier.

De plus, il convient de souligner que notre méthode met l'accent sur la maximisation du plus grand level set associé au DOA, une notion parfois négligée dans la littérature, comme illustré par (BANKS, 2002). La comparaison a délibérément été basée sur la stabilité asymptotique. Cette spécificité n'est pas anodine : nombre de méthodes évoquées au cours de ce manuscrit ne traitent pas des formes de stabilité plus nuancées comme la stabilité exponentielle ou l'ISS. Or, notre méthode et notre schéma d'optimisation ont été spécifiquement conçus pour examiner ces formes de stabilité plus complexes, sans nécessiter de modifications majeures à notre algorithme.

En conclusion, chaque méthode présente ses avantages et ses inconvénients. Notre méthode, grâce à sa flexibilité, offre une performance compétitive par rapport aux méthodes reconnues. Cependant, notre analyse montre également qu'il existe toujours un espace pour l'amélioration et l'exploration de nouvelles techniques. Dans le futur, il serait intéressant d'intégrer des aspects de différentes méthodes pour créer une approche hybride qui capitalise sur les points forts de chaque technique tout en minimisant leurs faiblesses.

4.4 Enjeux d'une application industrielle concrète

L'analyse de la stabilité des systèmes est un enjeu majeur dans de nombreux domaines d'application, notamment dans l'industrie. Elle garantit la sécurité, la fiabilité et l'efficacité des systèmes en interaction avec l'environnement et les utilisateurs. Dans cette section, nous nous penchons sur un exemple concret d'un système industriel : le manche à retour d'effort. À travers cet exemple, nous explorons les défis liés à l'analyse de la stabilité et les solutions envisagées pour y faire face. Puis, en nous projetant vers l'avenir, nous évoquons les perspectives de recherche.

4.4.1 Modélisation du système

Le système industriel étudié ici est un manche à retour d'effort, utilisé pour fournir un retour au pilote. Ce système s'appuie sur des lois d'impédance adaptative pour moduler cet effort de retour, permettant une expérience utilisateur plus intuitive et précise. La loi d'impédance adaptative, un concept clé dans le domaine du contrôle, joue un rôle fondamental dans notre système de manche à retour d'effort. Cette loi permet de gérer de manière dynamique la réponse du système aux forces externes, ce qui est essentiel dans les interfaces homme-machine. Dans le cadre du manche à retour d'effort, elle ajuste le retour de force basé sur des paramètres tels que l'inertie, la viscosité et la raideur. L'étude de la stabilité est importante pour garantir une interaction sécurisée et efficace entre l'homme et la machine, en particulier dans des applications de haute précision telles que la robotique, l'aviation et les simulateurs de vol. Face à la complexité propre à cet exemple industriel, c'est la stabilité entrée-état qui se révèle être le critère pertinent pour évaluer le système, la stabilité autonome pouvant être abordée à travers la théorie des systèmes de Lure.

Le fonctionnement de ce manche peut être approximé par une modélisation classique masse-ressort. Une illustration de ce modèle est donnée par la Figure 4.14. De plus, pour aider à visualiser l'effort ressenti par le pilote via le manche, une représentation simplifiée de l'axe du système est également fournie à la Figure 4.15.

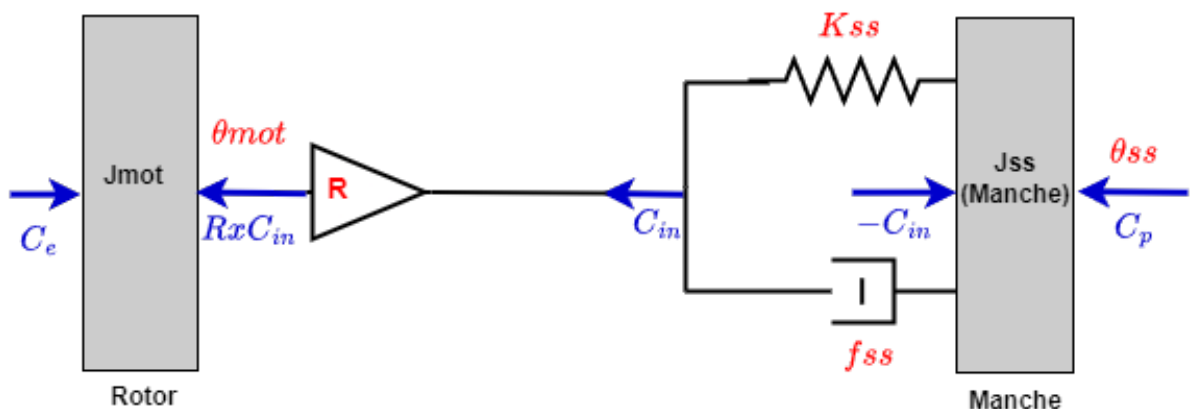


FIGURE 4.14 – Système masse-ressort

La dynamique du système est fondée sur le théorème du moment, appliqué séparément sur les inerties du moteur et du manche. Ces équations fondamentales sont exprimées par :

$$\begin{aligned} \sum C_{\text{ext}} &= J_{\text{mot}} \ddot{\theta}_{\text{mot}} \\ \sum C_{\text{ext}} &= J_{\text{ss}} \ddot{\theta}_{\text{ss}} \end{aligned}$$

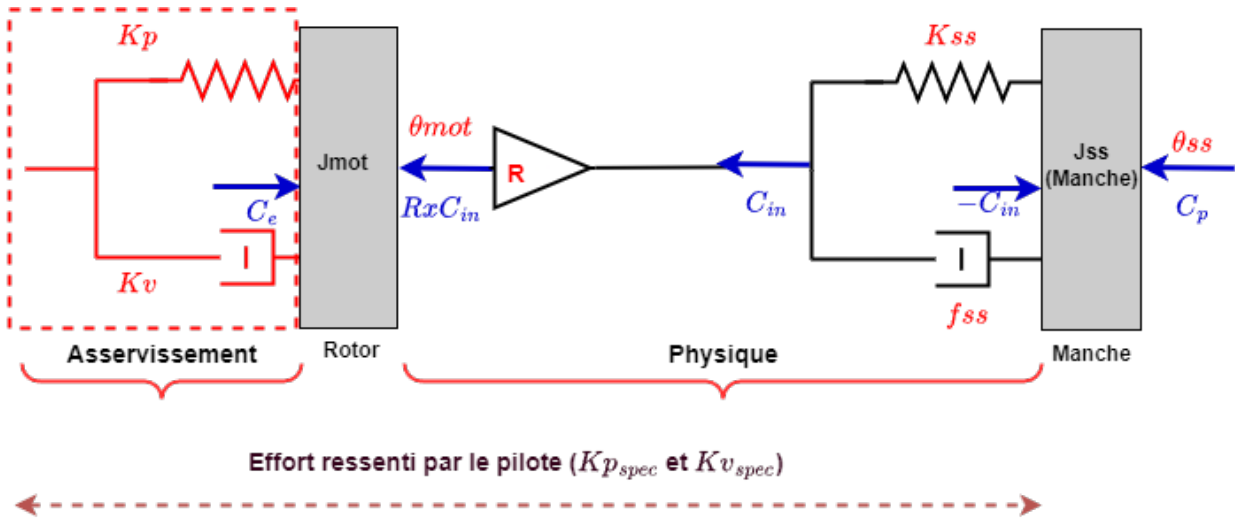


FIGURE 4.15 – Axe simplifié

Un élément clé du modèle est le rapport de réduction R , qui relie les angles du moteur et du manche. Les relations supplémentaires entre les divers paramètres et variables d'état sont décrites par (avec $\theta_{ss} = R\theta_{mot}$) :

$$\begin{aligned} C_e - RC_{in} &= J_{mot}\ddot{\theta}_{mot} \\ C_{in} + C_p &= J_{ss}\ddot{\theta}_{ss} \\ C_p &= -F_p r_p \quad (\text{car } \theta \text{ positive et } x \text{ positive}) \end{aligned}$$

avec les définitions suivantes :

r_p : le point d'appui pilote sur le manche,

r_b : le point du centre d'inertie du manche, servant pour l'effet d'accélération et de pesanteur.

Nous avons les relations :

$$\begin{aligned} C_{in} &= K_{ss}(R.\delta\theta_{mot} - \delta\theta_{ss}) + f_{ss}(R.\delta\dot{\theta}_{mot} - \delta\dot{\theta}_{ss}) \\ \delta\theta_{mot} &= \theta_{mot} - \theta_{mot0'} \\ \delta\theta_{ss} &= \theta_{ss} - \theta_{ss0'}. \end{aligned}$$

Par conséquent, nous pouvons exprimer les dynamiques par :

$$\begin{aligned} J_{mot}\delta\ddot{\theta}_{mot} &= C_e - R.K_{ss}(R.\delta\theta_{mot} - \delta\theta_{ss}) - R.f_{ss}(R.\delta\dot{\theta}_{mot} - \delta\dot{\theta}_{ss}) \\ J_{ss}\delta\ddot{\theta}_{ss} &= K_{ss}(R.\delta\theta_{mot} - \delta\theta_{ss}) + f_{ss}(R.\delta\dot{\theta}_{mot} - \delta\dot{\theta}_{ss}) + C_p. \end{aligned}$$

À partir de ces équations, nous déduisons les équations de base du système :

$$\begin{aligned} J_{mot}\delta\ddot{\theta}_{mot} &= C_e - R.K_{ss}R.\delta\theta_{mot} + R.K_{ss}\delta\theta_{ss} - R.f_{ss}R.\delta\dot{\theta}_{mot} + R.f_{ss}\delta\dot{\theta}_{ss} \\ J_{ss}\delta\ddot{\theta}_{ss} &= K_{ss}R.\delta\theta_{mot} - K_{ss}\delta\theta_{ss} + f_{ss}R.\delta\dot{\theta}_{mot} - f_{ss}\delta\dot{\theta}_{ss} + C_p. \end{aligned}$$

Ces équations peuvent être réarrangées pour fournir une perspective plus claire sur le comportement dynamique du système :

$$\begin{aligned}\delta\ddot{\theta}_{\text{mot}} &= -\frac{K_{\text{ss}}R^2}{J_{\text{mot}}}\delta\theta_{\text{mot}} - \frac{f_{\text{ss}}R^2}{J_{\text{mot}}}\delta\dot{\theta}_{\text{mot}} + \frac{RK_{\text{ss}}}{J_{\text{mot}}}\delta\theta_{\text{ss}} + \frac{R\cdot f_{\text{ss}}}{J_{\text{mot}}}\delta\dot{\theta}_{\text{ss}} + \frac{1}{J_{\text{mot}}}C_e \\ \delta\ddot{\theta}_{\text{ss}} &= \left(\frac{K_{\text{ss}}R}{J_{\text{ss}}}\right)\delta\theta_{\text{mot}} + \left(\frac{f_{\text{ss}}R}{J_{\text{ss}}}\right)\delta\dot{\theta}_{\text{mot}} - \left(\frac{K_{\text{ss}}}{J_{\text{ss}}}\right)\delta\theta_{\text{ss}} - \left(\frac{f_{\text{ss}}}{J_{\text{ss}}}\right)\delta\dot{\theta}_{\text{ss}} - \left(\frac{r_p}{J_{\text{ss}}}\right)F_p.\end{aligned}$$

Nous posons le vecteur d'état comme suit :

$$\begin{bmatrix} \delta\theta_{\text{mot}} \\ \delta\dot{\theta}_{\text{mot}} \\ \delta\theta_{\text{ss}} \\ \delta\dot{\theta}_{\text{ss}} \end{bmatrix}$$

Nous obtenons directement la formulation matricielle qui décrit le comportement dynamique du système :

$$\frac{d}{dt} \begin{bmatrix} \delta\theta_{\text{mot}} \\ \delta\dot{\theta}_{\text{mot}} \\ \delta\theta_{\text{ss}} \\ \delta\dot{\theta}_{\text{ss}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{\text{ss}}R^2}{J_{\text{mot}}} & -\frac{f_{\text{ss}}R^2}{J_{\text{mot}}} & \frac{RK_{\text{ss}}}{J_{\text{mot}}} & \frac{Rf_{\text{ss}}}{J_{\text{mot}}} \\ 0 & 0 & 0 & 1 \\ \frac{K_{\text{ss}}R}{J_{\text{ss}}} & \frac{f_{\text{ss}}R}{J_{\text{ss}}} & -\frac{K_{\text{ss}}}{J_{\text{ss}}} & -\frac{f_{\text{ss}}}{J_{\text{ss}}} \end{bmatrix} \begin{bmatrix} \delta\theta_{\text{mot}} \\ \delta\dot{\theta}_{\text{mot}} \\ \delta\theta_{\text{ss}} \\ \delta\dot{\theta}_{\text{ss}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{J_{\text{mot}}} \\ 0 \\ 0 \end{bmatrix} C_e + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{r_p}{J_{\text{ss}}} \end{bmatrix} F_p.$$

Après avoir défini notre système en termes de variables d'état, nous cherchons maintenant à extraire la sortie, à savoir θ_{mot} . Cette sortie est directement liée à la variation de l'angle du moteur :

$$\theta_{\text{mot}} = [1 \ 0 \ 0 \ 0] \begin{bmatrix} \delta\theta_{\text{mot}} \\ \delta\dot{\theta}_{\text{mot}} \\ \delta\theta_{\text{ss}} \\ \delta\dot{\theta}_{\text{ss}} \end{bmatrix} + \theta_{\text{mot}0} + 0C_e + 0F_p.$$

La sortie θ_{mot} est représentée en fonction des variables d'état. Dans cette étude, nous considérons que la condition initiale $\theta_{\text{mot}0}$ est nulle pour simplifier notre analyse, soit $\theta_{\text{mot}0} = 0$.

Regroupons maintenant les variables d'état sous la notation x_{mot} pour une représentation plus concise :

$$x_{\text{mot}} = \begin{bmatrix} \delta\theta_{\text{mot}} \\ \delta\dot{\theta}_{\text{mot}} \\ \delta\theta_{\text{ss}} \\ \delta\dot{\theta}_{\text{ss}} \end{bmatrix}$$

Le système peut alors être représenté sous la forme standard des équations d'état :

$$\begin{aligned}\dot{x}_{\text{mot}} &= A_{\text{mm}}x_{\text{mot}} + B_{\text{me}}C_e + B_{\text{mp}}F_p \\ \theta_{\text{mot}} &= C_{\text{mm}}x_{\text{mot}} + D_{\text{me}}C_e + D_{\text{mp}}F_p.\end{aligned}$$

Finalement, pour une analyse fréquentielle, nous dérivons les fonctions de transfert du système :

$$H_{mm} = C_{mm}(sI - A_{mm})^{-1}B_{me} + D_{me}$$

$$H_p = C_{mp}(sI - A_{mm})^{-1}B_{mp} + D_{mp}.$$

Cette formalisation nous permet ensuite de déduire les fonctions de transfert, qui sont essentielles pour l'analyse fréquentielle du système. La Figure 4.16 illustre le schéma bloc du système.

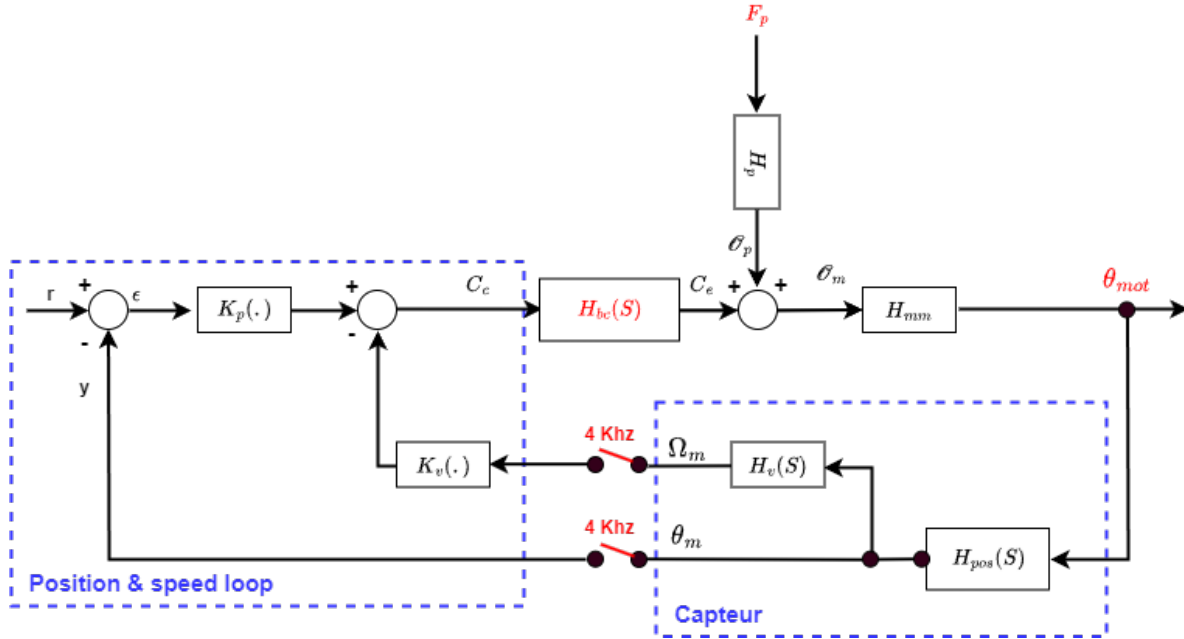


FIGURE 4.16 – Système industriel

Considérons tout d'abord la boucle de courant du système. La dynamique de cette boucle, représentée par H_{bc} , est donnée par :

$$\dot{x}_{bc} = A_{bc}x_{bc} + B_{bc}C_c$$

$$C_e = C_{bc}x_{bc}.$$

Les lois d'impédance, qui sont non-linéaires, nous donnent la relation suivante pour le courant de commande :

$$C_c = -K_p[\theta_m] - K_v[\Omega_m].$$

Ensuite, le capteur de position angulaire, H_{pos} , est décrit par :

$$\dot{x}_{pos} = A_{pos}x_{pos} + B_{pos}\theta_{mot}$$

$$\theta_m = C_{pos}x_{pos}.$$

La dérivée de la position, ou vitesse angulaire, est modélisée comme suit :

$$\dot{x}_v = A_v x_v + B_v \theta_m$$

$$\Omega_m = C_v x_v + D_v \theta_m.$$

Finalement, en intégrant toutes ces sous-parties, nous obtenons la modélisation globale du système. Pour simplifier cette représentation, nous considérons $r = 0$:

$$\begin{cases} \dot{x}_{\text{mot}} = A_{\text{mm}}x_{\text{mot}} + B_{\text{mp}}F_p + B_{\text{me}}C_{\text{bc}}x_{\text{bc}} \\ \dot{x}_{\text{bc}} = A_{\text{bc}}x_{\text{bc}} - B_{\text{bc}}K_p[\theta_m] - B_{\text{bc}}K_v[\Omega_m] \\ \dot{x}_{\text{pos}} = A_{\text{pos}}x_{\text{pos}} + B_{\text{pos}}C_{\text{mm}}x_{\text{mot}} \\ \dot{x}_v = A_vx_v + B_vC_{\text{pos}}x_{\text{pos}} \end{cases}$$

$$\begin{cases} \theta_{\text{mot}} = C_{\text{mm}}x_{\text{mot}} + D_{\text{mp}}F_p + B_{\text{me}}C_{\text{bc}}x_{\text{bc}} \\ C_M = C_{\text{bc}}x_{\text{bc}} \\ \theta_m = C_{\text{pos}}x_{\text{pos}} \\ \Omega_m = C_vx_v + D_vC_{\text{pos}}x_{\text{pos}} \end{cases}$$

La stabilité du système, en l'absence de perturbation, est appelée stabilité autonome. Elle est évaluée en considérant $F_p = 0$. En revanche, la stabilité entrée - état évalue la stabilité du système en présence d'une perturbation F_p , sous l'hypothèse que $\|F_p\| < F_{\text{max}}$.

4.4.2 Perspectives et collaborations

Le système industriel du manche à retour d'effort est caractérisé par sa nature affine par morceaux. Cette particularité non différentiable pose de sérieux défis lorsqu'il s'agit d'analyser et d'entraîner des modèles associés. En effet, les méthodes traditionnelles d'analyse des systèmes sont souvent conçues pour des systèmes différentiables, ce qui n'est pas le cas ici. Pour contourner cet obstacle, nous avons fait appel à des approximateurs de lois d'effort qui sont différentiables, à savoir des réseaux de neurones. Ces approximateurs, grâce à leur capacité à modéliser des fonctions non linéaires complexes, se sont avérés être des outils puissants pour notre étude.

Le système présente des non-linéarités par secteur borné. Si dans la littérature, des méthodes comme les contraintes intégrales quadratiques ([Integral Quadratic Constraints \(IQC\)](#)) ou les systèmes de Lure permettent d'analyser et de démontrer la stabilité des systèmes avec des non-linéarités sectorisées ([AHBE et al., 2020](#); [CUNIS et LEGAT, 2023](#); [IANNELLI et al., 2019](#); [YIN et al., 2020](#)), ces approches ne sont pas directement transposables à notre contexte. En effet, ces méthodes ne traitent pas de la stabilité entrée - état, essentielle pour notre étude, même si elles abordent des non-linéarités sectorisées. À notre connaissance, dans l'état actuel de l'art, des méthodes traitant des non-linéarités sectorisées avec la stabilité ISS demeurent une lacune.

La mise en œuvre de la stabilité entrée - état pour ce système industriel du manche à retour d'effort s'est heurtée à un obstacle significatif : le temps de calcul. En raison de la complexité du système, les processus d'optimisation nécessitent parfois plusieurs jours, avec des risques de plantages informatiques ou de non-convergence des solutions. Ces incidents nous obligent souvent à recommencer les calculs depuis le début. Il est important de noter que ces difficultés ne remettent pas en cause la validité de notre approche, mais soulignent la nécessité d'aborder la problématique du temps de calcul de notre outil tout en conservant la précision des résultats obtenus. Nous avons consacré des efforts considérables à l'exploration de cette problématique et poursuivons activement notre recherche pour surmonter ce défi.

Pour pallier cette contrainte du temps de calcul, nous nous sommes orientés vers deux axes majeurs de recherche, déjà évoqués dans ce mémoire. D'une part, nous avons exploré la structure des systèmes inter-connectés. Ces systèmes, composés de plusieurs sous-systèmes en interaction, offrent une manière modulaire d'aborder des systèmes complexes. En décomposant le système industriel

en ses composants individuels, nous pouvons réduire sa complexité. Ainsi, chaque sous-système peut être analysé séparément en termes de stabilité entrée - état, comme décrit dans la section 1.5.7. L'interconnexion entre ces sous-systèmes est alors modélisée par des équations algébriques supplémentaires, exploitant les propriétés ISS pour établir un théorème de petit-gain pour ces systèmes non linéaires invariants dans le temps.

D'autre part, nous nous sommes penchés sur la robustesse au maillage dans la section 3.4.5. Cette notion, essentielle dans notre étude sur l'approximation du domaine d'attraction en utilisant la théorie de Lyapunov, se concentre sur la capacité de notre méthode à produire des résultats fiables malgré les variations de la qualité du maillage. La question centrale est de garantir que les propriétés vérifiées sur les points du maillage s'étendent aux points intermédiaires, ceux non directement représentés dans le maillage. Alors, en optant pour un maillage contenant moins de points, tout en conservant la précision des résultats, nous pouvons réaliser des économies significatives en termes de temps de calcul.

Cette dualité d'approches, combinant la modularité des systèmes inter-connectés avec la robustesse au maillage, est au cœur de notre stratégie pour optimiser les temps de calcul sans compromettre la précision. Ces travaux de recherche s'inscrivent dans une vision plus large visant à développer des méthodes d'entraînement pour l'analyse de la stabilité des systèmes complexes. Cette thèse représente la première étape vers cette ambition. Dans la continuité de cette démarche, une nouvelle thèse a été initiée il y a près de deux ans à Montréal au Canada. J'ai eu l'opportunité de m'y rendre durant un mois, collaborant étroitement avec l'équipe sur place pour approfondir ces axes de recherche et les adapter au contexte industriel.

Il est à noter que nous continuons activement à travailler sur cette problématique. Ainsi, il est probable que de nouveaux résultats, encore en phase d'analyse au moment de la rédaction de ce manuscrit, puissent être présentés lors de la soutenance.

Conclusion générale

Conclusion générale

Cette thèse nous a permis d'explorer en profondeur les complexités et les subtilités de la stabilité dans les systèmes dynamiques non linéaires. Elle offre ainsi un aperçu précieux et une méthodologie innovante pour analyser et assurer la stabilité dans divers contextes.

Nous avons commencé par un examen approfondi des principes théoriques de la stabilité et des méthodes classiques utilisées. Ces débuts ont mis en lumière la valeur et les limites de ces approches. Cela nous a permis de mettre en évidence l'importance et la pertinence des travaux d'Alexandre Lyapunov. Nous avons introduit et détaillé les définitions de stabilité et la méthode directe de Lyapunov. Cette fondation théorique a fourni un cadre solide pour nos investigations ultérieures.

Nous avons ensuite effectué une revue de l'état de l'art en présentant les méthodes théoriques, numériques et automatiques existantes. Suite à cette étude, nous avons développé et implémenté une méthode d'optimisation robuste et innovante pour identifier des fonctions de Lyapunov efficaces dans divers contextes. Cette approche se distingue non seulement par son caractère générique et adaptable, mais aussi par sa capacité à maximiser le domaine de stabilité estimé garanti de la fonction de Lyapunov, élément central pour garantir la stabilité des systèmes dynamiques.

Enfin, nous avons détaillé une série de tests et d'analyses réalisés à partir de divers scénarios. Nous avons évalué et comparé l'efficacité des méthodes de stabilité introduites. Ces évaluations et comparaisons ont permis d'apporter un éclairage sur les forces et les faiblesses de chaque technique, guidant ainsi les choix futurs d'approches et de méthodologies pour des applications spécifiques dans le domaine industriel.

À travers ce travail de recherche, nous avons fourni une compréhension sur l'étude de la stabilité des systèmes dynamiques grâce aux notions de stabilité, enrichie par une revue des contributions scientifiques actuelles. Cette démarche a conduit à la conception d'une méthode d'optimisation novatrice et polyvalente pour l'analyse de la stabilité. Cette méthode d'optimisation est prête à faire face aux défis complexes et variés du monde industriel d'aujourd'hui.

Perspectives

Au terme de ces travaux, il apparaît pertinent de se tourner vers l'avenir et d'envisager les multiples directions possibles que pourrait prendre la recherche dans le domaine de l'automatique. Tout au long de cette thèse, nous avons examiné diverses perspectives et idées futures, certaines étant brièvement mentionnées, d'autres explorées plus en profondeur. Nous avons mis en lumière des perspectives les plus prometteuses identifiées au cours de nos travaux. Elles pourraient être utiles dans une suite à cette thèse. Nous soulignons plus particulièrement celles qui possèdent une portée plus large et un potentiel d'impact significatif dans la continuation et l'approfondissement de nos recherches.

Dans le cadre de nos travaux, nous avons adopté une méthode spécifique pour déterminer le maillage, bien que diverses autres approches soient disponibles et méritent une exploration future. Une

piste prometteuse à développer réside dans l'utilisation du gradient du candidat actuel de la fonction de Lyapunov durant le processus d'optimisation. Plus précisément, l'analyse du gradient lors de la première exécution de l'algorithme d'optimisation pourrait s'avérer instructive : un gradient plus important pourrait signaler le besoin d'un maillage plus fin dans les exécutions subséquentes. Cette stratégie pourrait améliorer la précision et l'efficacité de l'optimisation.

Il convient de noter que nos expérimentations initiales avec l'algorithme génétique et l'optimisation par essaim de particules ont été menées en utilisant des paramètres de base. Nous avons observé qu'une simple modification de l'algorithme d'optimisation pouvait se traduire par un gain significatif en temps de calcul, sans compromettre l'efficacité au niveau des valeurs obtenues. Ainsi, une perspective intéressante de recherche serait d'explorer d'autres algorithmes d'optimisation et d'examiner minutieusement les paramètres associés à chacun d'eux. Un ajustement fin des paramètres pourrait révéler des configurations optimales plus adaptées à notre contexte spécifique.

Nous avons également initié une application de notre outil sur un système industriel. Les contraintes de temps nous ont empêchés de produire des résultats significatifs à ce stade. À court terme, l'objectif est d'approfondir cet exemple industriel pour obtenir des résultats probants en matière de stabilité autonome et de stabilité entrée - état. Le travail initial implique de modéliser notre système en équations mathématiques et de trouver plusieurs solutions envisageables afin que la mise en oeuvre se réalise avec les meilleurs résultats possibles. Cela constitue une base solide pour des investigations futures dans ce domaine.

Au cours de cette recherche, nous avons intégré la procédure de robustesse au maillage dans des contextes théoriques. Bien que les exemples examinés soient principalement académiques, cette approche montre un potentiel prometteur pour les applications futures. L'utilisation d'un maillage plus grossier a démontré des résultats préliminaires positifs en termes de conservation des domaines de stabilité estimés. Cependant, il est important de noter que, dans nos exemples étudiés, cette approche n'a pas conduit à une amélioration majeure du temps de calcul. Néanmoins, l'intégration de la robustesse au maillage reste une piste de recherche attrayante. Elle offre des perspectives prometteuses pour renforcer la fiabilité et l'efficacité de nos méthodes utilisées dans la gestion des systèmes complexes et de grande dimension. Ces systèmes sont typiques du secteur industriel.

Les contraintes de temps inhérentes à cette thèse n'ont pas permis la réalisation d'essais approfondis sur des systèmes industriels. Ces tests auraient sans doute permis d'enrichir et d'offrir une compréhension pratique et concrète de notre méthode dans des environnements industriels complexes. Cette absence d'évaluation pratique et industrielle limite l'application immédiate de nos résultats, mais les tendances observées et les résultats préliminaires obtenus sont encourageants. Ces éléments laissent entrevoir des possibilités intéressantes pour des applications industrielles futures. Ils suggèrent que notre approche pourrait se révéler efficace dans le cadre de déploiements à plus grande échelle et dans des environnements industriels variés.

Enfin, un objectif à plus long terme est de rendre notre algorithme apte à fonctionner en mode embarqué ou en ligne. Actuellement conçu pour une utilisation hors ligne, où il analyse la stabilité d'un système donné, l'algorithme est à intégrer directement dans les systèmes en cours de fonctionnement. Ce n'est qu'un objectif, mais l'algorithme serait capable de se mettre à jour dynamiquement en réponse aux modifications des paramètres du système en temps réel. Cette transition vers un mode embarqué impliquerait probablement l'adoption d'une approche sans modèle (model-free), qui serait mise en oeuvre une fois le système identifié à l'aide d'un réseau de neurones. Des travaux préliminaires en ce sens sont déjà en cours à Montréal, au Canada, promettant des applications en temps réel et ouvrant des perspectives enthousiasmantes pour l'utilisation pratique et l'implémentation de nos travaux.

Bibliographie

- ABATE, A., AHMED, D., GIACOBBE, M., & PERUFFO, A. (2020). Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3), 773-778.
- AHBE, E., IANNELLI, A., & SMITH, R. S. (2020). Region of attraction analysis of nonlinear stochastic systems using Polynomial Chaos Expansion. *Automatica*, 122, 109187.
- ALOUI, M., HAMIIDI, F., & JERBI, H. (2021). Maximizing the Domain of attraction of nonlinear systems : A PSO optimization approach. *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, 375-380.
- AMBROSINO, R., & GARONE, E. (2012). Robust stability of linear uncertain systems through piecewise quadratic Lyapunov functions defined over conical partitions. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2872-2877.
- ANDERSON, J., & PAPACHRISTODOULOU, A. (2015). ADVANCES IN COMPUTATIONAL LYAPUNOV ANALYSIS USING SUM-OF-SQUARES PROGRAMMING. *Discrete & Continuous Dynamical Systems-Series B*, 20(8).
- ANTSAKLIS, P. J., & MICHEL, A. N. (1997). *Linear systems* (T. 8). Springer.
- APS, M. (2019). Mosek optimization toolbox for matlab. *User's Guide and Reference Manual, Version, 4*, 1.
- BAIER, R., GRÜNE, L., & FREYR HAFSTEIN, S. (2012). Linear programming based Lyapunov function computation for differential inclusions.
- BANKS, C. (2002). Searching for Lyapunov functions using genetic programming. *Virginia Polytech Institute, unpublished*.
- BERKENKAMP, F., TURCHETTA, M., SCHOELLIG, A., & KRAUSE, A. (2017). Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30.
- BHATIA, N. P., & HÁJEK, O. (2006). *Local semi-dynamical systems* (T. 90). Springer.
- BHATIA, N. P., & SZEGÖ, G. P. (2002). *Stability theory of dynamical systems*. Springer Science & Business Media.
- BJÖRNSSON, J., GIESL, P., & HAFSTEIN, S. (2014). Algorithmic verification of approximations to complete Lyapunov functions. *Proceedings of the 21st international symposium on mathematical theory of networks and systems*, 180, 1181-1188.
- BJÖRNSSON, J., GIESL, P., HAFSTEIN, S., KELLETT, C. M., & LI, H. (2014). Computation of continuous and piecewise affine Lyapunov functions by numerical approximations of the Massera construction. *53rd IEEE Conference on Decision and Control*, 5506-5511.
- BJÖRNSSON, J., GUDMUNDSSON, S., & HAFSTEIN, S. (2015). Class library in C++ to compute Lyapunov functions for nonlinear systems. *IFAC-PapersOnLine*, 48(11), 778-783.
- BOCQUILLON, B., FEYEL, P., SANDOU, G., & RODRIGUEZ-AYERBE, P. (2020a). Computation of neural networks lyapunov functions for discrete and continuous time systems with domain of attraction maximization. *12th International Conference on Neural Computation Theory and Applications (NCTA)*.

- BOCQUILLON, B., FEYEL, P., SANDOU, G., & RODRIGUEZ-AYERBE, P. (2020b). Efficient construction of neural networks Lyapunov functions with domain of attraction maximization. *17th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.
- BOCQUILLON, B., FEYEL, P., SANDOU, G., & RODRIGUEZ-AYERBE, P. (2022). A Comprehensive Framework to Determine Lyapunov Functions for a Set of Continuous Time Stability Problems. *IECON 2022—48th Annual Conference of the IEEE Industrial Electronics Society*, 1-6.
- BORCHERS, B. (1999). CSDP, AC library for semidefinite programming. *Optimization methods and Software*, 11(1-4), 613-623.
- BOTTOU, L. (2010). Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010 : 19th International Conference on Computational Statistics Paris France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, 177-186.
- BOTTOU, L., CURTIS, F. E., & NOCEDAL, J. (2018). Optimization methods for large-scale machine learning. *SIAM review*, 60(2), 223-311.
- CAMILLI, F., CESARONI, A., GRÜNE, L., & WIRTH, F. (2006). Stabilization of controlled diffusions and Zubov's method. *Stochastics and Dynamics*, 6(03), 373-393.
- CAMILLI, F., & GRÜNE, L. (2002). Characterizing attraction probabilities via the stochastic Zubov equation.
- CAMILLI, F., GRÜNE, L., & WIRTH, F. (2001). A generalization of Zubov's method to perturbed systems. *SIAM Journal on Control and Optimization*, 40(2), 496-515.
- CAMILLI, F., GRÜNE, L., & WIRTH, F. (2004). Construction of Lyapunov functions on the domain of asymptotic nullcontrollability : numerics. *IFAC Proceedings Volumes*, 37(13), 715-720.
- CAMILLI, F., GRÜNE, L., & WIRTH, F. (2007). A regularization of Zubov's equation for robust domains of attraction. In *Nonlinear control in the year 2000* (p. 277-289). Springer.
- CAMILLI, F., GRÜNE, L., & WIRTH, F. (2008). Control Lyapunov functions and Zubov's method. *SIAM Journal on Control and Optimization*, 47(1), 301-326.
- CHANG, Y.-C., ROOHI, N., & GAO, S. (2019). Neural lyapunov control. *Advances in neural information processing systems*, 32.
- CHEN, G. (2004). Stability of nonlinear systems. *Encyclopedia of RF and Microwave Engineering*, 4881-4896.
- CHEN, Y., SHI, Y., & ZHANG, B. (2018). Optimal control via neural networks : A convex approach. *arXiv preprint arXiv :1805.11835*.
- CHESI, G. (2007a). Estimating the domain of attraction via union of continuous families of Lyapunov estimates. *Systems & control letters*, 56(4), 326-333.
- CHESI, G. (2007b). Estimating the domain of attraction via union of continuous families of Lyapunov estimates. *Systems & control letters*, 56(4), 326-333.
- CHESI, G. (2009). Estimating the domain of attraction for non-polynomial systems via LMI optimizations. *Automatica*, 45(6), 1536-1541.
- CHESI, G., GARULLI, A., TESI, A., & VICINO, A. (2003). Solving quadratic distance problems : an LMI-based approach. *IEEE Transactions on Automatic Control*, 48(2), 200-212.
- CHESI, G., TESI, A., & VICINO, A. (2002). Computing optimal quadratic Lyapunov functions for polynomial nonlinear systems via LMIs. *IFAC Proceedings Volumes*, 35(1), 43-48.
- CHETAEV, N. G., NADLER, M., BABISTER, A., & BURLAK, J. (1961). The stability of motion. (*No Title*).
- CHOI, M.-D., LAM, T. Y., & REZNICK, B. (1995). Sums of squares of real polynomials. *Proceedings of Symposia in Pure mathematics*, 58, 103-126.
- CUNIS, T., & LEGAT, B. (2023). Sequential sum-of-squares programming for analysis of nonlinear systems. *2023 American Control Conference (ACC)*, 756-772.

- DAVISON, E., & KURAK, E. (1971). A computational method for determining quadratic Lyapunov functions for non-linear systems. *Automatica*, 7(5), 627-636.
- DE, R. (1986). Learning representations by back-propagation errors. *nature*, 323, 533-536.
- DE FIGUEIREDO, R. J., & CHEN, G. (1993). *Nonlinear feedback control systems : an operator theory approach*. Academic Press Professional, Inc.
- DE JONG, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. University of Michigan.
- DE SARKAR, A., & RAO, N. D. (1971). Zubov's method and transient-stability problems of power systems. *Proceedings of the Institution of Electrical Engineers*, 118(8), 1035-1040.
- DESOER, C. A., & VIDYASAGAR, M. (2009). *Feedback systems : input-output properties*. SIAM.
- DOHERTY, A. C., PARRILO, P. A., & SPEDALIERI, F. M. (2002). Distinguishing separable and entangled states. *Physical Review Letters*, 88(18), 187904.
- EBERHART, R., & KENNEDY, J. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, 39-43.
- EGHBAL, N., PARIZ, N., & KARIMPOUR, A. (2013). Discontinuous piecewise quadratic Lyapunov functions for planar piecewise affine systems. *Journal of Mathematical Analysis and Applications*, 399(2), 586-593.
- EIBEN, A. E., & SMITH, J. E. (2015). *Introduction to evolutionary computing*. Springer.
- FEYEL, P. (2015). *Optimisation des correcteurs par les métaheuristiques. Application à la stabilisation inertielle de ligne de visée*. [thèse de doct., CentraleSupélec].
- FEYEL, P. (2017). *Robust control optimization with metaheuristics*. John Wiley & Sons.
- FRANKE, C., & SCHABACK, R. (1998). Solving partial differential equations by collocation using radial basis functions. *Applied Mathematics and Computation*, 93(1), 73-82.
- FRASNEDO, S. (2016). *Optimisation des lois de commande d'un imageur sur critère optronique. Application à un imageur à deux étages de stabilisation*. [thèse de doct., Université Paris-Saclay].
- FREUND, R. M. (2004). Introduction to semidefinite programming (SDP). *Massachusetts Institute of Technology*, 8-11.
- GENESIO, R., TARTAGLIA, M., & VICINO, A. (1985). On the estimation of asymptotic stability regions : State of the art and new proposals. *IEEE Transactions on automatic control*, 30(8), 747-755.
- GIESL, P. (2004). Approximation of domains of attraction and Lyapunov functions using radial basis functions. *IFAC Proceedings Volumes*, 37(13), 697-702.
- GIESL, P. (2007). *Construction of global Lyapunov functions using radial basis functions* (T. 1904). Springer.
- GIESL, P. (2008a). Construction of a local and global Lyapunov function for discrete dynamical systems using radial basis functions. *Journal of Approximation Theory*, 153(2), 184-211.
- GIESL, P. (2008b). Construction of a local and global Lyapunov function using radial basis functions. *IMA journal of applied mathematics*, 73(5), 782-802.
- GIESL, P., & HAFSTEIN, S. (2012). Existence of piecewise linear Lyapunov functions in arbitrary dimensions. *Discrete Contin. Dyn. Syst*, 32(10), 3539-3565.
- GIESL, P., & HAFSTEIN, S. (2014). Computation of Lyapunov functions for nonlinear discrete time systems by linear programming. *Journal of Difference Equations and Applications*, 20(4), 610-640.
- GIESL, P., & HAFSTEIN, S. (2015a). Computation and verification of Lyapunov functions. *SIAM Journal on Applied Dynamical Systems*, 14(4), 1663-1698.
- GIESL, P., & HAFSTEIN, S. (2015b). Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems-Series B*, 20(8), 2291-2331.

- GIESL, P. A., & HAFSTEIN, S. F. (2014). Revised CPA method to compute Lyapunov functions for nonlinear systems. *Journal of Mathematical Analysis and Applications*, 410(1), 292-306.
- GROSMAN, B., & LEWIN, D. R. (2005). Automatic generation of Lyapunov functions using genetic programming. *IFAC Proceedings Volumes*, 38(1), 75-80.
- GRUJIC, L. T., et al. (1994). Solutions to Lyapunov stability problems of sets : nonlinear systems with differentiable motions. *International Journal of Mathematics and Mathematical Sciences*, 17, 103-112.
- GRUNE, L., & WURTH, F. (2000). Computing control Lyapunov functions via a Zubov type algorithm. *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, 3, 2129-2134.
- GRÜNE, L. (2004). *Asymptotic behavior of dynamical and control systems under perturbation and discretization*. Springer.
- GRÜNE, L., SEMMLER, W., & BERNARD, L. (2007). Firm value, diversified capital assets, and credit risk : towards a theory of default correlation.
- HACHICHO, O. (2007). A novel LMI-based optimization algorithm for the guaranteed estimation of the domain of attraction using rational Lyapunov functions. *Journal of the Franklin Institute*, 344(5), 535-552.
- HACHICHO, O., & TIBKEN, B. (2002). Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 3, 3150-3155.
- HAFSTEIN, S., KELLETT, C. M., & LI, H. (2014). Continuous and piecewise affine Lyapunov functions using the Yoshizawa construction. *2014 American Control Conference*, 548-553.
- HAFSTEIN, S. F. (2004). A constructive converse Lyapunov theorem on exponential stability. *Discrete & Continuous Dynamical Systems*, 10(3), 657.
- HAFSTEIN, S. F. (2005). A constructive converse Lyapunov theorem on asymptotic stability for nonlinear autonomous ordinary differential equations. *Dynamical Systems*, 20(3), 281-299.
- HAFSTEIN, S. F. (2009). An algorithm for constructing Lyapunov functions. *Electronic Journal of Differential Equations*, 1(Mon. 01-09), 08-101.
- HAGAN, M. T., DEMUTH, H. B., & BEALE, M. (1997). *Neural network design*. PWS Publishing Co.
- HAHN, W., et al. (1967). *Stability of motion* (T. 138). Springer.
- HAN, J., et al. (2016). Deep learning approximation for stochastic control problems. *arXiv preprint arXiv :1611.07422*.
- HENRION, D., & GARULLI, A. (2005). *Positive polynomials in control* (T. 312). Springer Science & Business Media.
- HEWIT, J., & STOREY, C. (1967). Optimisation of the Zubov and Ingwerson methods for constructing Lyapunov functions. *Electronics Letters*, 5(3), 211-213.
- HEWIT, J., & STOREY, C. (1969). Comparison of numerical methods in stability analysis. *International Journal of Control*, 10(6), 687-701.
- HIRSCH, M. W., SMALE, S., & DEVANEY, R. L. (2012). *Differential equations, dynamical systems, and an introduction to chaos*. Academic press.
- HIRWA, S. (2013). *Méthodes de commande avancées appliquées aux viseurs*. [thèse de doct., Supélec].
- HOLLAND, J. H. (1992). *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- HU, T., & LIN, Z. (2003). Composite quadratic Lyapunov functions for constrained control systems. *IEEE Transactions on Automatic Control*, 48(3), 440-450.
- IANNELLI, A., SEILER, P., & MARCOS, A. (2019). Region of attraction analysis with integral quadratic constraints. *Automatica*, 109, 108543.

- JARVIS-WLOSZEK, Z., FEELEY, R., TAN, W., SUN, K., & PACKARD, A. (2003). Some controls applications of sum of squares programming. *42nd IEEE international conference on decision and control (IEEE Cat. No. 03CH37475)*, 5, 4676-4681.
- JIANG, Z.-P., MAREELS, I., & WANG, Y. (1995). A Lyapunov formulation of nonlinear small gain theorem for interconnected systems. *IFAC Proceedings Volumes*, 28(14), 625-630.
- JIANG, Z.-P., MAREELS, I. M., & WANG, Y. (1996). A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems. *Automatica*, 32(8), 1211-1215.
- JOHANSSON, M., & RANTZER, A. (1997). Computation of piecewise quadratic Lyapunov functions for hybrid systems. *1997 European Control Conference (ECC)*, 2005-2010.
- JONES, M., & PEET, M. M. (2023). Existence of Partially Quadratic Lyapunov Functions That Can Certify The Local Asymptotic Stability of Nonlinear Systems. *2023 American Control Conference (ACC)*, 4130-4135.
- KALTOFEN, E., LI, B., YANG, Z., & ZHI, L. (2008). Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, 155-164.
- KALTOFEN, E. L., LI, B., YANG, Z., & ZHI, L. (2012). Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *Journal of Symbolic Computation*, 47(1), 1-15.
- KAMYAR, R., & PEET, M. (2014). Polynomial optimization with applications to stability analysis and control-alternatives to sum of squares. *arXiv preprint arXiv :1408.5119*.
- KENNEDY, J., & EBERHART, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, 4, 1942-1948.
- KERR, C., MARGOLIS, S., & VOGT, W. (1964). Discussion of "Control engineering applications of VI Zubov's construction procedure for Liapunov functions". *IEEE Transactions on Automatic Control*, 9(2), 196-196.
- KHALIL, H. K. (2002). *Nonlinear systems* (T. 3). Prentice Hall Upper Saddle River, NJ.
- KINGMA, D. P., & BA, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.
- KOCVARA, M., STINGL, M., & GBR, P. (2005). PENBMI user's guide (version 2.0). *software manual, PENOPT GbR, Hauptstrasse A, 31*, 91338.
- KOZA, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4, 87-112.
- KOZA, J. R., KEANE, M. A., YU, J., MYDLOWEC, W., & BENNETT III, F. H. (2000). Automatic synthesis of both the topology and parameters for a controller for a three-lag plant with a five-second delay using genetic programming. *Workshops on Real-World Applications of Evolutionary Computation*, 168-177.
- KRISHNA, A. V., SANGAMREDDI, C., & PONNADA, M. R. (2019). Optimal Design of Roof-Truss using GA in MATLAB. *i-Manager's Journal on Structural Engineering*, 8(1), 39.
- LA SALLE, J. P. (1976). *The stability of dynamical systems*. SIAM.
- LASALLE, J. (1960). Some extensions of Liapunov's second method. *IRE Transactions on circuit theory*, 7(4), 520-527.
- LEFFERTS, E. J. (1965). *A guide of the application of the Liapunov direct method to flight control systems* (rapp. tech.).
- LI, H., BAIER, R., GRÜNE, L., HAFSTEIN, S. F., & WIRTH, F. (2014). Computation of local ISS Lyapunov functions via linear programming. *MTNS 2014*.
- LI, H., BAIER, R., GRÜNE, L., HAFSTEIN, S. F., & WIRTH, F. (2015). Computation of local ISS Lyapunov functions with low gains via linear programming.

- LI, S., SHAO, Z., & GUAN, Y. (2017). A dynamic neural network approach for efficient control of manipulators. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, 49(5), 932-941.
- LI, S., ZHANG, Y., & JIN, L. (2016). Kinematic control of redundant manipulators using neural networks. *IEEE transactions on neural networks and learning systems*, 28(10), 2243-2254.
- LI, Z., XIA, Y., SU, C.-Y., DENG, J., FU, J., & HE, W. (2014). Missile guidance law based on robust model predictive control using neural-network optimization. *IEEE transactions on neural networks and learning systems*, 26(8), 1803-1809.
- LIU, J. (2013). *Radial Basis Function (RBF) neural network control for mechanical systems : design, analysis and Matlab simulation*. Springer Science & Business Media.
- LIU, W.-C., & ZHANG, S.-G. (2017). Variable gradient approach to construct Lyapunov function for judging stability of nonLinear systems. *International Conference on Electrical Engineering and Automation Control (ICEEAC 2017)*.
- LOFBERG, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, 284-289.
- LYAPUNOV, A. M. (1892). The general problem of the stability of motion. *International journal of control*, 55(3), 531-534.
- MALISOFF, M. (2003). Further results on Lyapunov functions and domains of attraction for perturbed asymptotically stable systems. *arXiv preprint math/0311267*.
- MARGOLIS, S., & VOGT, W. (1963). Control engineering applications of VI Zubov's construction procedure for Lyapunov functions. *IEEE Transactions on Automatic Control*, 8(2), 104-113.
- MASSERA, J. L. (1949). On Liapounoff's conditions of stability. *Annals of Mathematics*, 705-721.
- MCCULLOCH, W. S., & PITTS, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.
- MCGOUGH, J. S., CHRISTIANSON, A. W., & HOOVER, R. C. (2010). Symbolic computation of Lyapunov functions using evolutionary algorithms. *Proceedings of the 12th IASTED international conference*, 15, 508-515.
- MOHAMMED, N., & GIESL, P. (2015). Grid refinement in the construction of Lyapunov functions using radial basis functions. *Discrete Contin. Dyn. Syst. Ser. B*, 20(8), 2453-2476.
- MOHAMMEDALI, K. H., FADHEL, F., & AHMAD, N. A. (2015). Modified variable gradient method for constructing the Lyapunov function using nonlinear programming method. *2015 10th Asian Control Conference (ASCC)*, 1-6.
- MOHLER, R. R. (1991). *Nonlinear systems (vol. 2) applications to bilinear control*. Prentice-Hall, Inc.
- NAJAFI, E., BABUŠKA, R., & LOPES, G. A. (2016). A fast sampling method for estimating the domain of attraction. *Nonlinear dynamics*, 86, 823-834.
- NAKAMURA-ZIMMERER, T., GONG, Q., & KANG, W. (2021). Adaptive deep learning for high-dimensional Hamilton–Jacobi–Bellman equations. *SIAM Journal on Scientific Computing*, 43(2), A1221-A1247.
- NARENDRA, K. S. (2014). *Frequency domain criteria for absolute stability*. Elsevier.
- NIU, B., LIU, Y., ZHOU, W., LI, H., DUAN, P., & LI, J. (2019). Multiple Lyapunov functions for adaptive neural tracking control of switched nonlinear nonlower-triangular systems. *IEEE Transactions on Cybernetics*, 50(5), 1877-1886.
- NOROOZI, N., KARIMAGHAEI, P., SAFAEI, F., & JAVADI, H. (2008). Generation of lyapunov functions by neural networks. *Proceedings of the World Congress on Engineering, 2008*.
- O'NEILL, M., RYAN, C., KEIJZER, M., & CATTOLICO, M. (2001). Crossover in grammatical evolution : The search continues. *Genetic Programming : 4th European Conference, EuroGP 2001 Lake Como, Italy, April 18–20, 2001 Proceedings 4*, 337-347.
- PAPACHRISTODOULOU, A. (2004). Analysis of nonlinear time-delay systems using the sum of squares decomposition. *Proceedings of the 2004 American Control Conference*, 5, 4153-4158.

- PAPACHRISTODOULOU, A., ANDERSON, J., VALMORBIDA, G., PRAJNA, S., SEILER, P., & PARRILO, P. A. (2013). Sum of squares optimization toolbox for MATLAB user's guide. *arXiv preprint arXiv :1310.4716*.
- PAPACHRISTODOULOU, A., & PRAJNA, S. (2002). On the construction of Lyapunov functions using the sum of squares decomposition. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 3, 3482-3487.
- PARKS, P. C., & HAHN, V. (1993). *Stability theory*. Prentice-Hall, Inc.
- PARRILO, P. A. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology.
- PARRILO, P. A., & PERETZ, R. (2004). An inequality for circle packings proved by semidefinite programming. *Discrete & Computational Geometry*, 31, 357-367.
- PERKO, L. (2013). *Differential equations and dynamical systems* (T. 7). Springer Science & Business Media.
- PETRIDIS, V., & PETRIDIS, S. (2006). Construction of neural network based Lyapunov functions. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 5059-5065.
- PETTERSEN, K. Y. (2017). Lyapunov sufficient conditions for uniform semiglobal exponential stability. *Automatica*, 78, 97-102.
- PEUTEMAN, J., & AEYELS, D. (2002). Exponential stability of nonlinear time-varying differential equations and partial averaging. *Mathematics of Control, Signals and Systems*, 15(1), 42-70.
- PINGUET, J. (2023). *Contribution à la synthèse de contrôleurs neuronaux robustes par imitation* [thèse de doct., université Paris-Saclay].
- POUILLY-CATHELAIN, M. (2020). *Synthèse de correcteurs s'adaptant à des critères multiples de haut niveau par la commande prédictive et les réseaux de neurones* [thèse de doct., université Paris-Saclay].
- POWERS, V., & WÖRMANN, T. (1998). An algorithm for sums of squares of real polynomials. *Journal of pure and applied algebra*, 127(1), 99-104.
- PRAJNA, S. (2006). Barrier certificates for nonlinear model validation. *Automatica*, 42(1), 117-126.
- PRAJNA, S., & JADBABAIE, A. (2004). Safety verification of hybrid systems using barrier certificates. *International Workshop on Hybrid Systems : Computation and Control*, 477-492.
- PRAJNA, S., & PAPACHRISTODOULOU, A. (2003). Analysis of switched and hybrid systems beyond piecewise quadratic methods. *Proceedings of the 2003 American Control Conference, 2003.*, 4, 2779-2784.
- PRAJNA, S., PAPACHRISTODOULOU, A., & PARRILO, P. A. (2002). Introducing SOSTOOLS : A general purpose sum of squares programming solver. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 1, 741-746.
- PRAJNA, S., PAPACHRISTODOULOU, A., SEILER, P., & PARRILO, P. A. (2004). SOSTOOLS : Sum of squares optimization toolbox for MATLAB.
- PRAJNA, S., PAPACHRISTODOULOU, A., & WU, F. (2004). Nonlinear control synthesis by sum of squares optimization : A Lyapunov-based approach. *2004 5th Asian control conference (IEEE Cat. No. 04EX904)*, 1, 157-165.
- PRAJNA, S., PARRILO, P. A., & RANTZER, A. (2004). Nonlinear control synthesis by convex optimization. *IEEE Transactions on Automatic Control*, 49(2), 310-314.
- PROKHOROV, D. V. (1994). A Lyapunov machine for stability analysis of nonlinear systems. *Proceedings of 1994 IEEE International Conference on Neural Networks*, 2, 1028-1031.
- RAISSI, M., PERDIKARIS, P., & KARNIADAKIS, G. E. (2019). Physics-informed neural networks : A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378, 686-707.

- REICHENS DÖRFER, E., ODENTHAL, D., & WOLLHERR, D. (2017). Grammatical evolution of robust controller structures using wilson scoring and criticality ranking. *Genetic Programming : 20th European Conference, EuroGP 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings 20*, 194-209.
- RICHARDS, S. M., BERKENKAMP, F., & KRAUSE, A. (2018). The lyapunov neural network : Adaptive stability certification for safe learning of dynamical systems. *Conference on Robot Learning*, 466-476.
- RODDEN, J. (1964). Numerical applications of Lyapunov stability theory. *Joint Automatic Control Conference*, (2), 261-268.
- ROSSO, M. M., CUCUZZA, R., DI TRAPANI, F., & MARANO, G. C. (2021). Nonpenalty machine learning constraint handling using PSO-SVM for structural optimization. *Advances in Civil Engineering*, 2021, 1-17.
- SABOURI, M., SETOODEH, P., & ASEMANI, M. H. (2020). Construction of lyapunov functions using multi-objective genetic algorithm. *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 1-5.
- SANDBERG, I. (1963). On the properties of some systems that distort signals—I. *Bell System Technical Journal*, 42(5), 2033-2046.
- SCHULTZ, D., & GIBSON, u. J. (1962). The variable gradient method for generating Liapunov functions. *Transactions of the American Institute of Electrical Engineers, Part II : Applications and Industry*, 81(4), 203-210.
- SEARSON, D. P., LEAHY, D. E., & WILLIS, M. J. (2010). GPTIPS : an open source genetic programming toolbox for multigene symbolic regression. *Proceedings of the International multiconference of engineers and computer scientists*, 1, 77-80.
- SEILER, P. (2003). Stability region estimates for SDRE controlled systems using sum of squares optimization. *Proceedings of the 2003 American Control Conference, 2003.*, 3, 1867-1872.
- SERPEN, G. (2005). Empirical approximation for Lyapunov functions with artificial neural nets. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2, 735-740.
- SHIMOOKA, H., & FUJIMOTO, Y. (2000). Generating robust control equations with genetic programming for control of a rolling inverted pendulum. *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, 491-495.
- SLOTINE, J.-J. E., LI, W., et al. (1991). *Applied nonlinear control* (T. 199). Prentice hall Englewood Cliffs, NJ.
- SONTAG, E. D. (1983). A Lyapunov-like characterization of asymptotic controllability. *SIAM journal on control and optimization*, 21(3), 462-471.
- STROGATZ, S. H. (2018). *Nonlinear dynamics and chaos with student solutions manual : With applications to physics, biology, chemistry, and engineering*. CRC press.
- STURM, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4), 625-653.
- TALBI, E.-G. (2009). *Metaheuristics : from design to implementation*. John Wiley & Sons.
- TAN, W., & PACKARD, A. (2006). Stability region analysis using sum of squares programming. *2006 American Control Conference*, 6-pp.
- TESI, A., VILLORESI, F., & GENESIO, R. (1996). On the stability domain estimation via a quadratic Lyapunov function : convexity and optimality properties for polynomial systems. *IEEE Transactions on Automatic Control*, 41(11), 1650-1657.
- TOH, K.-C., TODD, M. J., & TÜTÜNCÜ, R. H. (1999). SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4), 545-581.

- VAN DEN BERGH, F., et al. (2007). *An analysis of particle swarm optimizers* [thèse de doct., University of Pretoria].
- VANNELLI, A., & VIDYASAGAR, M. (1985). Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, 21(1), 69-80.
- VENKATARAMAN, P. (2009). *Applied optimization with MATLAB programming*. John Wiley & Sons.
- WANG, D., TAN, D., & LIU, L. (2018). Particle swarm optimization algorithm : an overview. *Soft computing*, 22, 387-408.
- WANG, T., GAO, H., & QIU, J. (2015). A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. *IEEE Transactions on Neural Networks and Learning Systems*, 27(2), 416-425.
- WANG, X.-M., NIU, B., WU, G.-q., LI, J.-Q., DUAN, P.-y., & YANG, D. (2018). Adaptive RBF neural-network-based design strategy for non-strict-feedback nonlinear systems by using integral Lyapunov functions. *IEEE Access*, 6, 75076-75085.
- WEN, G., YU, W., HU, G., CAO, J., & YU, X. (2015). Pinning synchronization of directed networks with switching topologies : A multiple Lyapunov functions approach. *IEEE Transactions on Neural Networks and Learning Systems*, 26(12), 3239-3250.
- WENDLAND, H. (1998). Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *Journal of approximation theory*, 93(2), 258-272.
- WENDLAND, H. (2004). *Scattered data approximation* (T. 17). Cambridge university press.
- WOOD, G., & ZHANG, B. (1996). Estimation of the Lipschitz constant of a function. *Journal of Global Optimization*, 8, 91-103.
- WU, F., & PRAJNA, S. (2004). A new solution approach to polynomial LPV system analysis and synthesis. *Proceedings of the 2004 American Control Conference*, 2, 1362-1367.
- YAMASHITA, M., FUJISAWA, K., & KOJIMA, M. (2003). Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0). *Optimization Methods and Software*, 18(4), 491-505.
- YANG, L., SUN, D., & TOH, K.-C. (2015). SDPNAL+ : a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation*, 7(3), 331-366.
- YIN, H., PACKARD, A., ARCAK, M., & SEILER, P. (2020). Reachability analysis using dissipation inequalities for uncertain nonlinear systems. *Systems & Control Letters*, 142, 104736.
- YOSHIZAWA, T. (1966). Stability theory by Liapunov's second method. *Publications of the Mathematical Society of Japan*.
- YU, Y.-n., & VONGSURIYA, K. (1967). Nonlinear Power System Stability Study by Liapunov Function and Zubov's Method. *IEEE Transactions on Power Apparatus and Systems*, PAS-86(12), 1480-1485. <https://doi.org/10.1109/TPAS.1967.291911>
- ZAMES, G. (1959). Conservation of bandwidth in nonlinear operations, Quarterly Progress Report.
- ZAMES, G. (1966a). On the input-output stability of time-varying nonlinear feedback systems part one : Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE transactions on automatic control*, 11(2), 228-238.
- ZAMES, G. (1966b). On the input-output stability of time-varying nonlinear feedback systems—Part II : Conditions involving circles in the frequency plane and sector nonlinearities. *IEEE transactions on automatic control*, 11(3), 465-476.
- ZHANG, B., LAM, J., & XU, S. (2014). Stability analysis of distributed delay neural networks based on relaxed Lyapunov–Krasovskii functionals. *IEEE Transactions on Neural Networks and Learning Systems*, 26(7), 1480-1492.
- ZHANG, G., YAO, M., XU, J., & ZHANG, W. (2020). Robust neural event-triggered control for dynamic positioning ships with actuator faults. *Ocean Engineering*, 207, 107292.

- ZHAO, X.-Y., SUN, D., & TOH, K.-C. (2010). A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4), 1737-1765.
- ZONGMIN, W. (1992). Hermite-Birkhoff interpolation of scattered data by radial basis functions. *Approximation Theory and its Applications*, 8(2), 1-10.
- ZUBOV, V. I. (1964). Methods of AM Lyapunov and their application. (*No Title*).

Titre : Méthodes d'entraînement pour l'analyse de la stabilité d'un système complexe

Mots clés : Théorie de Lyapunov, système non linéaire, optimisation, domaine d'attraction, stabilité

Résumé : Ces dernières années, le domaine de la recherche en automatique a été transformé par l'émergence de technologies avancées telles que l'Intelligence Artificielle. Cette thèse se concentre sur l'utilisation de techniques d'entraînement et d'optimisation pour démontrer la stabilité des contrôleurs intelligents, un élément clé dans le processus de certification. Elle débute par une exploration des principes de la stabilité de Lyapunov, une théorie introduite à la fin du XIXe siècle, qui permet d'étudier la stabilité d'un système dynamique sans hypothèse structurelle. Elle passe en revue certaines méthodes théoriques, numériques et automatiques pour identifier une fonction de Lyapunov, soulignant les limites

de certaines approches existantes. Cette recherche aboutit au développement de méthodes d'optimisation innovantes pour déterminer des fonctions de Lyapunov, qui se distinguent par leur adaptabilité et leur efficacité à maximiser le domaine de stabilité estimé. L'algorithme proposé est testé rigoureusement, démontrant son adaptabilité à divers scénarios industriels tout en le comparant à certaines méthodes reconnues de l'état de l'art. En conclusion, cette thèse établit les bases pour l'application pratique d'un algorithme dans des systèmes industriels complexes, ouvrant la voie à des applications futures plus larges, détaillées en fin de manuscrit.

Title : Training methods for analyzing the stability of a complex system

Keywords : Lyapunov theory, nonlinear system, optimization, domain of attraction, stability

Abstract : In recent years, the field of automation research has been transformed by the emergence of advanced technologies such as Artificial Intelligence. This thesis focuses on the use of training and optimization techniques to demonstrate the stability of intelligent controllers, a key element in the certification process. It begins by exploring the principles of Lyapunov stability, a theory introduced in the late 19th century, which allows the stability of a dynamic system to be studied without structural assumption. It reviews certain theoretical, numerical, and automatic methods to identify a Lyapunov function, highlighting the limitations of certain existing approaches. This re-

search leads to the development of innovative optimization methods for determining Lyapunov functions, which are distinguished by their adaptability and effectiveness in maximizing the estimated stability domain. The proposed algorithm is rigorously tested, demonstrating its adaptability to various industrial scenarios while comparing it to certain recognized methods of the state of the art. In conclusion, this thesis lays the groundwork for the practical application of an algorithm in complex industrial systems, paving the way for broader future applications, detailed at the end of the manuscript.