Efficient Algorithms for Control and Reinforcement Learning

Eloïse Berthier

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École normale supérieure

Algorithmes efficaces pour le contrôle et l'apprentissage par renforcement

Efficient Algorithms for Control and Reinforcement Learning

Soutenue par
Eloïse BERTHIER
Le 27 octobre 2022

École doctorale n°386
Sciences Mathématiques de Paris Centre

Spécialité
Informatique

Composition du jury :

Emmanuel TRÉLAT Professeur, Sorbonne Université	*Président*
Marianne AKIAN Directeur de recherche, Inria Saclay	*Rapporteur*
Florence d'ALCHÉ-BUC Professeur, Télécom Paris	*Rapporteur*
Justin CARPENTIER Chargé de recherche, Inria Paris, ENS	*Examinateur*
Colin JONES Associate professor, EPFL	*Examinateur*
Francis BACH Directeur de recherche, Inria Paris, ENS	*Directeur de thèse*

1794
ENS
ÉCOLE NORMALE SUPÉRIEURE

PSL

À ma grand-mère
Nicole Berthier,

version du 10/10/2022

Contents

Remerciements

Je souhaite d'abord exprimer toute ma reconnaissance à mon directeur de thèse, Francis Bach. Je mesure l'opportunité que tu m'as offerte en me permettant de réfléchir et de progresser à tes côtés au cours de ces trois dernières années. Tu m'as fourni un encadrement sans faille, à la fois scientifiquement et humainement. J'espère continuer à m'inspirer des qualités que j'ai admirées en travaillant avec toi, parmi lesquelles une certaine sérénité pour aborder les problèmes, toujours mêlée d'humilité.

Je remercie particulièrement les chercheurs avec qui j'ai eu la chance de collaborer, d'abord Justin Carpentier pour son indispensable initiation au contrôle et à la robotique, puis Alessandro Rudi pour sa maîtrise infaillible des noyaux, et Ziad Kobeissi pour sa remarquable attention aux détails et aux difficultés cachées.

J'adresse mes remerciements à Marianne Akian et à Florence d'Alché-Buc pour avoir accepté de rapporter ce manuscrit qui touche à différents domaines, et à Colin Jones et Emmanuel Trélat qui me font l'honneur de l'examiner. Au cours de ma thèse, j'ai eu la chance de pouvoir présenter et obtenir de précieux retours sur mon travail, grâce aux invitations de Stéphane Gaubert et d'Olivier Bokanowski. Qu'ils en soient ici remerciés. Merci également à Didier Henrion pour ses échanges sur les SOS, et à Ana Busic d'avoir pris part à mon comité de suivi doctoral.

Je suis reconnaissante envers la Direction Générale de l'Armement qui m'a permis de réaliser ma thèse dans les meilleures conditions matérielles, en tant qu'ingénieure de l'armement. Je remercie Alain Droniou pour son suivi et son conseils, ainsi que David Filliat de m'accueillir à l'ENSTA Paris à l'issue de cette thèse.

Si je suis venue travailler chaque jour avec le sourire, c'est grâce aux membres de l'équipe Sierra, sans oublier, évidemment, l'inséparable équipe Willow. Je pense d'abord à Oumayma, Etienne et Zerui du bureau C409, le seul bureau au complet avant 9h, qui ont su supporter des températures tantôt polaires, tantôt tropicales, en plus du doux ronronnement des trains. Ces trois années ont aussi été l'occasion de participer à des projets parallèles à ma thèse. Je remercie Denis Merigoux avec qui nous avons relancé le Junior Seminar malgré l'adversité des circonstances, et Gaspard, Théophile et Clémence d'avoir brillamment pris la relève. Un grand merci à Clémentine Fourrier pour toute son énergie dans la co-organisation de deux éditions des Rendez-vous des Jeunes Mathématiciennes et Informaticiennes, une cause qui nous tenait à toutes les deux particulièrement à cœur. À Antoine Bambade, avec qui nous avons animé pendant deux ans un groupe de lecture sur le contrôle : bravo, nous sommes arrivés au dernier chapitre ! J'ai eu la chance d'avoir pu

échanger avec les membres de l'équipe s'intéressant au contrôle, en particulier Pierre-Cyril Aubin, Marc Lambert et Philippe Rigollet. Nos longues discussions ont bien sûr influencé mon travail. Merci également à la team robotique : Yann, Wilson, Thomas, Louis, Fabian, Armand, qui m'ont permis de mieux en appréhender les nombreux aspects. J'ai une pensée particulière pour Jean-Paul Laumond qui a su insuffler sa vision inspirante de la robotique dans nos équipes.

Ces trois années ont aussi été rythmées par les discussions (pluri)quotidiennes autour de la machine à café, indispensables pour trouver l'inspiration et garder la motivation. Parmi ceux que je n'ai pas déjà cités, j'ai eu le plaisir d'y croiser régulièrement, entre autres et dans le désordre : Bertille, Céline, Grégoire, Loucas, Radu, Raphaël, Rémi, Thomas, Ulysse, Vivien, Yana, Yann... Elles ont aussi été ponctuées par des Ground Control et autres Friday Beers, ainsi qu'un inoubliable séminaire d'équipe à Avignon. Ces moments, s'ils ont été un temps trop rares, n'en sont que plus précieux. J'en profite pour remercier le personnel de l'Inria, qui fait tout le nécessaire et le suffisant pour que le 2 rue Simone Iff soit un environnement si agréable au quotidien. En particulier, un grand merci à Hélène Bessin-Rousseau pour sa gentillesse et sa disponibilité, et un clin d'œil à la team piano de l'AGOS et à Marion, grâce à qui cette dernière année a résonné au son de Ravel, ainsi qu'à Fouzia Bouzid pour son efficacité côté ENS.

J'adresse mes sincères remerciements à mes chers amis Guillaume, Clément, Pierre, Maxime et Marine, qui ont été particulièrement présents pendant ces trois années. En plus des expos, restos, cinés, dîners, jeux, et interminables discussions sur le statut des fonctionnaires, j'ai trouvé en vous une oreille attentive et un soutien indéfectible. Vous n'êtes pas étrangers au mystérieux mathématicien Godalle Marmanthier, toujours disponible pour chiner des constantes universelles ou des noms d'algorithmes absurdes, optimiser du code (même écrit depuis gedit), prouver un lemme retors, trouver LA bonne référence, ou relire quelques dizaines de pages (même de contrôle !) en un week-end.

Je ne saurais terminer sans remercier ma famille. Je pense à mes grands-parents de Dole, qui ont toujours su me faire découvrir d'autres horizons. À mon frère Antoine et à Manon, qui ont été une présence indispensable à Paris et dont la porte était toujours ouverte (en même temps j'avais la clef...). Merci à mon père pour m'avoir appris il y a bien longtemps les principaux prérequis pour cette thèse, à savoir les boucles for en Basic et la formule de Taylor avec reste intégral. Je n'oublie pas, bien sûr, le rôle des quelques enseignants de toutes disciplines, en particulier au lycée Gay-Lussac, qui ont su éveiller ma curiosité. Merci à ma mère, d'avoir toujours répondu présente. Enfin, je dédie ce travail à ma grand-mère Nicole, qui était si fière de moi.

Introduction et résumé des contributions

Contrôle et apprentissage

La théorie du contrôle optimal – aussi appelée commande optimale en français – trouve ses origines dans le calcul des variations (van Brunt, 2004). Il s'agit d'un problème ancien, formulé dès le XVII$^{\text{ème}}$ siècle, et que l'on peut résumer ainsi : quel chemin une onde doit-elle emprunter pour aller d'un point à un autre en temps minimal ? Le problème du contrôle optimal, qui en est une généralisation, apparaît dans les années 1950 sous l'impulsion de la cybernétique – l'étude des systèmes complexes (Wiener, 1948) – qui introduit notamment la notion de rétroaction. Il s'agit de trouver une commande (ou un contrôle) u qui doit être exercé à tout instant sur un système dynamique x évoluant dans le temps, afin de minimiser un coût L. Formellement, il s'agit d'un problème d'optimisation :

$$\begin{aligned} &\inf_{u(\cdot)} \int_0^T L(x(t), u(t))\mathrm{d}t \\ \text{s.t. } &\forall t \in [0, T], \;\; \dot{x}(t) = f(x(t), u(t)). \end{aligned}$$

Ce formalisme est flexible et peut s'adapter à des problèmes discrets ou continus, déterministes ou stochastiques, lui permettant ainsi de modéliser un grand nombre de problèmes, comme la recherche d'une trajectoire spatiale, la commande d'une machine-outil, le déplacement d'un robot anthropomorphe ou la conduite autonome. Les années 1960 à 1980 connaissent l'avènement de l'automatique, discipline dédiée à la modélisation et à la commande des systèmes linéaires (Bourlès and Kwan, 2013). Au-delà de l'aspect scientifique, le contrôle des systèmes linéaires est structurant en ingénierie, qui se consacre largement à la conception de système asservis – c'est-à-dire régulés par une boucle de rétroaction –, et en particulier à l'étude de leur stabilité. À partir des années 1980, de nombreux travaux sont consacrés à l'étude des systèmes non-linéaires et au contrôle robuste (Zames, 1981), c'est-à-dire un ensemble de méthodes pouvant tolérer une spécification inexacte du modèle.

L'apprentissage par renforcement est un sous-domaine de l'apprentissage automatique. Il s'agit, pour un agent, d'apprendre à agir dans un environnement, de façon à maximiser les récompenses reçues au cours du temps. L'environnement réagit de façon stochastique à l'état s de l'agent et à l'action a qu'il vient d'effectuer, en lui envoyant une récompense r et en modifiant son état. Une formalisation de ce problème

sous forme de processus de décision Markovien a été introduite par Bellman en 1957 (Bellman, 1957b). Il s'agit d'un problème d'optimisation stochastique :

$$\max_{\pi:\mathcal{S}\to\mathcal{A}} \mathbb{E}_\pi \sum_{t=0}^{+\infty} r_t \ .$$

Une spécificité de l'apprentissage par renforcement est que la façon dont l'environnement réagit est *inconnue* pour l'agent. Il ne dispose pas d'un modèle de cet environnement, et se voit contraint d'*apprendre* à agir de façon optimale. La formulation de ce problème s'inspire des neurosciences et de la psychologie. En particulier, la métaphore de la récompense n'est pas sans rappeler les processus de renforcement induits par les neurotransmetteurs, ou les expériences de conditionnement chez les animaux. En pratique, les méthodes d'apprentissage par renforcement ont souvent été appliquées pour la résolution des jeux, avec succès pour le backgammon et les échecs dans les années 1990, puis les jeux vidéos Atari, le jeu de Go, et plus récemment les jeux en temps réel comme Starcraft. Depuis les années 2010, ces succès s'expliquent en grande partie par des progrès dans la mise en œuvre de techniques d'apprentissage automatique plus efficaces comme les réseaux de neurones.

En examinant les problèmes de contrôle optimal et d'apprentissage par renforcement, on notera de nombreuses similitudes (Bertsekas, 2019; Meyn, 2022). L'état d'un système – ou d'un agent – évolue dans le temps, en suivant une certaine dynamique, contrôlée en partie par sa commande – ou son action. Le but est de minimiser un coût – ou de maximiser une récompense, l'opposé d'un coût – au cours du temps. Néanmoins, trois différences notables subsistent :

1. le caractère déterministe ou stochastique ;
2. le caractère discret ou continu ;
3. le fait qu'un modèle d'évolution soit connu ou non.

Le formalisme des processus de décision Markoviens est naturellement stochastique, alors que les problèmes de contrôle optimal sont généralement écrits de façon déterministe, même s'il existe des problèmes de contrôle stochastique. Ce n'est donc pas une caractéristique qui les distingue fondamentalement, mais la stochasticité génère souvent des difficultés théoriques et pratiques supplémentaires.

D'autre part, le problème de contrôle optimal est généralement formulé en temps continu et en état continu, face à l'apprentissage par renforcement en temps discret, et souvent en état discret. Cette différence entre discret et continu n'est pas aussi bénigne qu'elle n'y paraît. Les problèmes discrets sont par nature combinatoires : il n'est pas possible de prévoir localement l'état d'un système au temps suivant en l'observant à un temps donné. Cela peut rendre leur étude particulièrement complexe quand le nombre d'états augmente. À l'inverse, une évolution continue, comme celle d'une équation différentielle, présente une certaine régularité : entre deux pas de temps proches, l'état du système est peu modifié. Cette différence explique par exemple le fait qu'un objet central en apprentissage par renforcement comme la fonction Q, qui sert à estimer la valeur d'un couple action-état, n'a pas de sens pour des problèmes à temps continu, car l'impact d'une action individuelle est négligeable. En temps continu, c'est le Hamiltonien qui joue un rôle similaire. Dans le Chapitre 3, nous explorons la notion de discrétisation en espace d'un problème continu, en étendant une méthode dédiée aux problèmes à temps continu vers des problèmes à temps discret.

Enfin, pour le contrôle optimal, la dynamique et le coût, qui constituent le *modèle*, sont des fonctions connues, qui peuvent donc être utilisées pour construire une solution optimale. Ce n'est pas le cas en apprentissage par renforcement : le processus qui génère la dynamique et les récompenses est caché, et n'est observé

qu'à travers un nombre fini d'observations. C'est là la principale difficulté supplémentaire de l'apprentissage par renforcement par rapport au contrôle optimal, qui justifie l'utilisation de techniques d'estimation. En somme, on pourrait considérer que l'apprentissage par renforcement est un problème de contrôle, couplé à un problème d'apprentissage de modèle. Dans la suite de cette thèse, nous relâcherons progressivement l'hypothèse du modèle connu, pour aller progressivement du paradigme *contrôle* au paradigme *contrôle + apprentissage*. En effet, si dans le Chapitre 3, nous supposons le modèle parfaitement connu, dans le Chapitre 4, nous supposons seulement qu'il appartient à un certain ensemble de modèles, *i.e.*, qu'il est connu jusqu'à un certain ordre, et nous développons des méthodes robustes qui fonctionnent sur toute cette classe de modèles. Dans le Chapitre 5, le modèle n'est connu qu'à travers un lot de n observations, et enfin, dans le Chapitre 6, ces observations sont reçues de façon incrémentale.

Vers des algorithmes efficaces

La plupart des problèmes de contrôle et d'apprentissage par renforcement ne peuvent être résolus analytiquement, il faut donc faire appel à des méthodes numériques. Si ces méthodes peuvent résoudre certains problèmes bien spécifiés et de faible dimension, elles sont d'une efficacité limitée pour des applications plus ambitieuses comme la robotique. En effet, de telles applications entraînent certaines contraintes :

- les dimensions du système, quoique modérées du point de vue de l'apprentissage automatique ($d \simeq 10$ ou 20), sont prohibitives pour la plupart des méthodes numériques pour le contrôle ;
- la dynamique est non-linéaire, empêchant l'utilisation directe des méthodes de contrôle linéaire ;
- le modèle n'est pas connu de façon exacte, rendant inutile la résolution exacte du problème ;
- certains calculs doivent se faire en temps réel, et/ou sur des systèmes embarqués, limitant ainsi l'accès aux ressources et temps de calculs.

Dans cette thèse, nous chercherons à développer et analyser des méthodes numériques qui tiennent compte de ces contraintes, et qui seront donc suspectibles d'être appliquées à des problèmes de robotique. De plus, un certain niveau de certification est souvent requis pour le déploiement d'algorithmes sur des systèmes physiques. C'est pourquoi nous chercherons si possible à développer des garanties théoriques pour ces algorithmes, sous forme de certificats ou de taux de convergence.

Résumé des contributions

Dans le Chapitre 1, nous introduisons formellement les problèmes de contrôle optimal et d'apprentissage par renforcement. Nous présentons d'abord les principaux résultats théoriques : le principe du maximum de Pontryagin et le principe d'optimalité de Bellman. Puis nous nous intéressons aux méthodes numériques existantes, dont certaines sont communes au contrôle et à l'apprentissage par renforcement comme la formulation par programmation linéaire. D'autres sont spécifiques au contrôle, comme les méthodes de tir indirectes, ou à l'apprentissage par renforcement, comme les méthodes de différences temporelles. Nous insistons sur les atouts et les limites de ces méthodes, dans le cadre d'applications à des problèmes de grande dimension.

Dans le Chapitre 2, nous présentons successivement plusieurs outils qui seront utilisés dans la suite de la thèse. Nous décrivons d'abord succinctement la dynamique d'un système robotique articulé et présentons deux algorithmes permettant de résoudre numériquement les problèmes de dynamique directe et inverse. Ces algorithmes seront utilisés dans un exemple numérique du Chapitre 4. Puis nous présentons plus en détail le domaine de l'optimisation polynomiale, ainsi que ses applications au problème de certification de stabilité (qui servira de méthode de référence dans le Chapitre 4), et à la résolution numérique du problème de contrôle optimal polynomial (dont nous proposerons une extension aux systèmes lisses non-polynomiaux dans le Chapitre 5). Nous introduisons ensuite les méthodes à noyaux, une classe d'algorithmes largement utilisés en apprentissage automatique. Ces méthodes présentent le double avantage d'être à la fois applicables sur des problèmes de grande dimension, et d'être bien comprises d'un point de vue théorique. Nous utilisons les méthodes à noyaux dans le Chapitre 5 pour représenter des fonctions positives, et dans le Chapitre 6 nous analyserons une version adaptée aux méthodes à noyaux de l'algorithme des différences temporelles. Enfin, nous présenterons brièvement les principes de l'algèbre max-plus, ainsi que son intérêt pour les problèmes de contrôle, que nous appliquerons ensuite à un processus Markovien déterministe dans le Chapitre 3.

Les deux premiers chapitres sont un état de l'art et ne présentent pas de contribution nouvelle. Les quatre chapitres suivants sont issus de travaux originaux qui constituent les contributions de cette thèse.

Contribution 1 : Discrétisation max-plus de processus de décision Markoviens déterministes.

Le Chapitre 3 est consacré à l'étude d'une méthode d'approximation max-plus pour les problèmes de contrôle à temps et espace continus, d'abord proposée par McEneaney (2003) puis étendue par Akian et al. (2008). Nous proposons d'adapter cette méthode aux processus de décision Markoviens déterministes. Elle permet d'approximer la fonction valeur comme une combinaison max-plus linéaire dans une dictionnaire de fonctions de base. Un algorithme naturel pour obtenir cette approximation est une variante de l'algorithme d'itération par valeurs : il s'agit d'appliquer de façon alternée l'opérateur de Bellman et un opérateur de projection max-plus sur l'espace généré par la base de fonctions. Une variante variationnelle de cette méthode permet de tirer parti de la max-plus linéarité de l'opérateur de Bellman. L'algorithme obtenu peut alors être décomposé en deux parties : une première étape de calculs préliminaires dont la complexité ne dépend pas de l'horizon temporel du processus Markovien, suivi d'une version réduite de l'algorithme d'itération par valeurs dont la complexité ne dépend que de l'horizon et du nombre de fonctions de base.

L'étape de calculs préliminaires est un problème d'optimisation. Dans le cas des processus de décision déterministes à état continu, nous proposons de le résoudre de façon approchée par une méthode de descente de gradient. D'autre part, nous proposons un dictionnaire de fonctions de bases qui permet de produire une discrétisation en espace du processus de décision. Nous analysons les erreurs produites par cette méthode d'approximation max-plus, pour deux dictionnaires de fonctions, en fonction de la régularité Lipschitz de la fonction valeur. Nous proposons ensuite une stratégie simple pour choisir de manière adaptative les dictionnaires de fonctions, en atténuant ainsi la malédiction de la dimension. Enfin, nous montrons empiriquement sur deux exemples de faible dimension que cette discrétisation max-plus est plus compacte, en termes de nombre de paramètres, qu'une discrétisation naïve du problème à état continu.

Ce chapitre est publié dans l'article de journal :

> E. Berthier and F. Bach, "Max-Plus Linear Approximations for Deterministic Continuous-State Markov Decision Processes," in *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 767-772, July 2020, doi:10.1109/LCSYS.2020.2973199.

Contribution 2 : Estimation de régions de stabilité pour les systèmes dynamiques non-linéaires.

Dans le Chapitre 4, on cherche à stabiliser un système dynamique autour d'un de ses points d'équilibre. Cela peut être effectué à l'aide d'un régulateur linéaire-quadratique (LQR). Il s'agit de linéariser localement le système, et de construire un contrôle stabilisant en boucle fermée autour de ce point. Si le système dynamique est linéaire, le contrôleur LQR calculé au point d'équilibre est valide sur tout l'espace, et le système ainsi contrôlé sera globalement asymptotiquement stable. Si le système est non-linéaire, cela n'est vrai que localement, sur une zone appelée région d'attraction ou de stabilité. Un problème important en pratique est d'estimer la taille de cette région d'attraction, surtout pour les systèmes très non-linéaires. Il s'agit concrètement d'exhiber une fonction de Lyapunov valide sur cette région. Pour des systèmes polynomiaux, il est possible de certifier la stabilité d'une région en faisant appel à l'optimisation polynomiale. Néanmoins, il en résulte un problème difficile à résoudre numériquement pour des systèmes de grande dimension. En effet, l'estimation de régions de stabilité n'étant souvent qu'un sous-problème d'un algorithme plus large qui l'appelle de façon répétée, elle doit pouvoir être effectuée rapidement.

Nous proposons deux certificats de stabilité qui peuvent être calculés efficacement pour des problèmes de grande dimension. Ils s'appliquent de façon robuste à une classe de systèmes dont les dérivées premières ou secondes sont bornées. En associant ces certificats à un oracle permettant de calculer des bornes sur les dérivées, nous proposons un algorithme simple d'estimation de régions de stabilité. Nous l'étendons ensuite au problème de suivi de trajectoire qui généralise la stabilisation autour d'un point d'équilibre. Enfin, nous validons expérimentalement cette approche sur des systèmes polynomiaux et non-polynomiaux de dimensions variées, dont un système robotique qui ne peut être traité par la méthode d'optimisation polynomiale.

Ce chapitre est publié dans l'article de conférence :

> E. Berthier, J. Carpentier and F. Bach, "Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems," *2021 European Control Conference (ECC)*, 2021, pp. 1412-1419, doi:10.23919/ECC54610.2021.9655071.

Contribution 3 : Sommes de carrés en dimension infinie pour le contrôle optimal.

Dans le Chapitre 5, nous proposons une nouvelle méthode d'approximation numérique pour les problèmes de contrôle optimal. Cette méthode s'applique à des problèmes dont la dynamique est inconnue, et n'est observée qu'à travers un nombre fini d'échantillons. En ce sens, nous nous plaçons dans le paradigme sans modèle de l'apprentissage par renforcement. Nous considérons la méthode d'optimisation polynomiale développée par Lasserre et al. (2008). Il s'agit d'utiliser le dual Lagrangien de la formulation faible du problème de contrôle, qui est un programme linéaire en dimension infinie. En particulier, il est nécessaire de représenter le Hamiltonien comme une fonction positive, c'est-à-dire, dans le cas polynomial, un polynôme positif. La hiérarchie moment - sommes de carrés fournit une méthode numérique basée sur la programmation semi-définie positive. Cependant, cette méthode est coûteuse en termes de temps de calcul, et ne peut pas être appliquée à des systèmes de grande dimension.

Nous proposons d'étendre cette méthode à des systèmes non-nécessairement polynomiaux, et connus uniquement à partir d'échantillons. Pour cela, nous utilisons une représentation des fonctions positives lisses issue des méthodes à noyaux, récemment proposée par Marteau-Ferey et al. (2020). Cette représentation est de dimension infinie, et s'appuie sur l'espace de fonctions des méthodes à noyaux, appelé espace de Hilbert à noyau reproduisant. Nous prouvons que sous des conditions de régularité, pour les systèmes contrôle-affine, la représentation du Hamiltonien comme somme de carrés dans un espace de fonctions lisses est exacte. Après sous-échantillonnage, cette méthode conduit également à un programme semi-défini positif,

pour lequel nous proposons une résolution par méthode de Newton. Nous illustrons cette approche sur un problème de contrôle élémentaire. Nous montrons en particulier comment choisir un espace de fonctions adapté à la structure du problème considéré.

Ce chapitre a été accepté pour publication dans la conférence :

> E. Berthier, J. Carpentier, A. Rudi and F. Bach, "Infinite-dimensional Sums-of-Squares for Optimal Control," *Conference on Decision and Control*, 2022, arXiv:2110.07396.

Contribution 4 : Analyse de l'algorithme non-paramétrique des différences temporelles.

L'algorithme des différences temporelles est un algorithme classique en apprentissage par renforcement qui permet d'évaluer une politique donnée, de façon incrémentale à partir d'observations. Dans le Chapitre 6, nous proposons une analyse non-asymptotique de l'algorithme des différences temporelles, dans sa version non-paramétrique et régularisée. Il s'agit d'une généralisation en dimension infinie de l'algorithme des différences temporelles avec approximation linéaire. Il a été prouvé que cet algorithme avec approximation linéaire converge, non pas vers la fonction valeur, mais vers le point fixe de l'opérateur de Bellman projeté, qui en est général une fonction différente. Cela s'explique par les capacités d'approximation limitées d'une approximation linéaire en dimension finie.

Nous montrons que si l'algorithme non-paramétrique est utilisé dans un espace de Hilbert à noyau reproduisant universel, c'est-à-dire dense dans l'espace des fonctions de carré intégrable, alors les itérés moyennés convergent vers la fonction valeur, et ce même si elle n'appartient pas à l'espace de Hilbert. Nous fournissons des taux de convergence explicites qui dépendent de la régularité relative de la fonction valeur par rapport à l'espace de fonctions. Nous traitons à la fois le cas où les observations sont indépendantes et identiquement distribuées, et le cas où elles sont issues d'une chaîne de Markov. Nous illustrons cette convergence sur un exemple numérique de processus Markovien à état continu.

Ce chapitre a été accepté pour publication dans la conférence :

> E. Berthier, Z. Kobeissi and F. Bach, "A Non-asymptotic Analysis of Non-parametric Temporal-Difference Learning," *Advances in Neural Information Processing Systems*, 2022, arXiv:2205.11831.

Introduction and Summary of the Contributions

Control and Learning

The theory of optimal control has its origins in the calculus of variations (van Brunt, 2004). It is an old problem, formulated as early as the XVIIth century, and which can be summarized as follows: what path should a wave take to get from one point to another in minimal time? The problem of optimal control, which is a generalization of this problem, appeared in the 1950s under the impulse of cybernetics – the study of complex systems (Wiener, 1948) – which introduced the notion of feedback. It is a question of finding a command (or control) u which must be applied at any moment to a dynamical system x evolving in time, in order to minimize a cost L. Formally, it is an optimization problem:

$$\begin{aligned} &\inf_{u(\cdot)} \int_0^T L(x(t), u(t))\mathrm{d}t \\ \text{s.t. } &\forall t \in [0, T], \;\; \dot{x}(t) = f(x(t), u(t)). \end{aligned}$$

This formalism is flexible and can be adapted to discrete or continuous, deterministic or stochastic problems, thus allowing it to model a large number of problems, such as the search for a spatial trajectory, the control of an industrial machine, the movement of an anthropomorphic robot or autonomous driving. The 1960s to 1980s saw the advent of automation, or systems theory, a discipline dedicated to the modeling and control of linear systems (Bourlès and Kwan, 2013). Beyond the scientific aspect, the control of linear systems is structuring in engineering, which is largely devoted to the design of servo systems – that is, systems regulated by a feedback loop – and in particular to the study of their stability. From the 1980s onwards, a lot of work has been devoted to the study of non-linear systems and robust control, *i.e.*, a set of methods that can tolerate model misspecification.

Reinforcement learning is a sub-field of machine learning. It consists, for an agent, in learning to act in an environment in order to maximize the rewards received over time. The environment reacts stochastically to the agent's state s and to the action a it has just performed, by sending it a reward r and by modifying its state. A formalization of this problem in the form of a Markov decision process was introduced by Bellman

in 1957 (Bellman, 1957b). It is a stochastic optimization problem:

$$\max_{\pi:\mathcal{S}\to\mathcal{A}} \mathbb{E}_\pi \sum_{t=0}^{+\infty} r_t \ .$$

A specificity of reinforcement learning is that the way the environment reacts is *unknown* to the agent. It does not have a model of this environment, and is forced to *learn* how to act in an optimal way. The formulation of this problem is inspired by neuroscience and psychology. In particular, the reward metaphor is reminiscent of reinforcement processes induced by neurotransmitters, or conditioning experiments in animals. In practice, reinforcement learning methods have often been applied to solve games, with success for backgammon and chess in the 1990s, then for Atari video games, the game of Go, and more recently for real-time games like Starcraft. Since the 2010s, much of this success has been due to advances in the implementation of more efficient machine learning techniques such as neural networks.

Looking at optimal control and reinforcement learning problems, one will note many similarities (Bertsekas, 2019; Meyn, 2022). The state of a system – or of an agent – evolves over time, following a certain dynamics, controlled in part by its control – or action. The goal is to minimize a cost – or maximize a reward, the opposite of a cost – over time. Nevertheless, three notable differences remain:

1. the deterministic or stochastic character ;
2. the discrete or continuous character;
3. whether a model of evolution is known or not.

The formalism of Markov decision processes is naturally stochastic, whereas optimal control problems are usually written deterministically, even if stochastic control problems exist. It is therefore not a characteristic that fundamentally distinguishes the two problems, but stochasticity often generates additional theoretical and practical difficulties.

On the other hand, the optimal control problem is usually formulated in continuous time and continuous state, while reinforcement learning is formulated in discrete time and often in discrete state. This difference between discrete and continuous is not as benign as it seems. Discrete problems are by nature combinatorial: it is not possible to predict locally the state of a system at the next time by observing it at a given time. This can make their study particularly complex when the number of states increases. On the other hand, a continuous evolution, such as that of a differential equation, presents a certain regularity: between two close time steps, the state of the system is only slightly modified. This difference explains, for example, the fact that a central object in reinforcement learning, such as the Q function, which is used to estimate the value of a state-action pair, is meaningless for continuous time problems, because the impact of an individual action is negligible. In continuous time, it is the Hamiltonian that plays a similar role. In Chapter 3, we explore the notion of space discretization of a continuous problem, by extending a method dedicated to continuous time problems to discrete time problems.

Finally, for optimal control, the dynamics and the cost, which constitute the *model*, are known functions, which can therefore be used to construct an optimal solution. This is not the case in reinforcement learning: the process that generates the dynamics and the rewards is hidden, and is only observed through a finite number of samples. This is the main additional difficulty of reinforcement learning compared to optimal control, which justifies the use of estimation techniques. In short, reinforcement learning could be considered as a control problem, coupled with the problem of learning the model. In the rest of this thesis, we will

progressively relax the assumption of the known model, to go progressively from the *control* paradigm to the *control + learning* paradigm. Indeed, while in Chapter 3, we assume the model is perfectly known, in Chapter 4, we only assume that it belongs to a certain set of models, *i.e.*, that it is known up to a certain order, and we develop robust methods that work over this whole class of models. In Chapter 5, the model is known only through a batch of n observations, and finally, in Chapter 6, these observations are received incrementally.

Towards Efficient Algorithms

Most control and reinforcement learning problems cannot be solved analytically, so numerical methods must be used. While these methods can solve some low-dimensional, well-specified problems, they are of limited effectiveness for more ambitious applications such as robotics. Indeed, such applications entail certain constraints:

- the dimensions of the system, although moderate from the viewpoint of machine learning ($d \simeq 10$ or 20), are prohibitive for most numerical methods designed for control;
- the dynamics is non-linear, preventing the direct use of linear control methods;
- the model is not known exactly, making it useless to solve the problem exactly;
- some computations must be done in real time, and/or on embedded systems, thus limiting access to computational resources and time.

In this thesis, we will try to develop and analyze numerical methods that take these constraints into account, and that can therefore possibly be applied to robotics problems. Moreover, a certain level of certification is often required for the deployment of algorithms on physical systems. This is why we will try if possible to develop theoretical guarantees for these algorithms, in the form of certificates or convergence rates.

Summary of the Contributions

In Chapter 1, we formally introduce the optimal control and reinforcement learning problems. We first present the main theoretical results: Pontryagin's maximum principle and Bellman's optimality principle. Then we focus on existing numerical methods, some of which are common to control and reinforcement learning such as the linear programming formulation. Others are specific to control, such as indirect shooting methods, or to reinforcement learning, such as temporal difference methods. We emphasize the strengths and limitations of these methods in the context of applications to high-dimensional problems.

In Chapter 2, we present successively several tools that will be used in the rest of the thesis. We first briefly describe the dynamics of an articulated robotic system and present two algorithms to numerically solve direct and inverse dynamics problems. These algorithms will be used in a numerical example of the Chapter 4. Then we present in more details the field of polynomial optimization, as well as its applications to the stability certification problem (which will be used as a reference method in Chapter 4), and to the numerical solution of the polynomial optimal control problem (of which we will propose an extension to non-polynomial smooth systems in Chapter 5). We then introduce kernel methods, a class of algorithms

widely used in machine learning. These methods have the dual advantage of being both applicable to high-dimensional problems and well understood from a theoretical point of view. We use kernel methods in Chapter 5 to represent non-negative functions, and in Chapter 6 we will analyze a version of the temporal difference algorithm adapted to kernel methods. Finally, we will briefly present the principles of max-plus algebra, as well as its interest for control problems, which we will then apply to a deterministic Markov decision process in Chapter 3.

The first two chapters are a state of the art and do not present any new contribution. The next four chapters are original pieces of work that constitute the contributions of this thesis.

Contribution 1: Max-plus discretization of deterministic Markov decision processes.

Chapter 3 is devoted to the study of a max-plus approximation method for control problems in continuous time and space, first proposed by McEneaney (2003) and extended by Akian et al. (2008). We propose to adapt this method to deterministic Markov decision processes. It approximates the value function as a linear max-plus combination in a dictionary of basis functions. A natural algorithm to obtain this approximation is a variant of the value iteration algorithm: the Bellman operator and a max-plus projection operator on the space generated by the basis functions are alternately applied. A variational variant of this method allows to take advantage of the max-plus linearity of the Bellman operator. The resulting algorithm can then be decomposed into two parts: a first step of preliminary computations whose complexity does not depend on the time horizon of the Markov process, followed by a reduced version of the value iteration algorithm whose complexity only depends on the horizon and the number of basis functions.

The preliminary computation step is an optimization problem. In the case of deterministic continuous state Markov decision processes, we propose to solve it in an approximate way by a gradient descent method. On the other hand, we propose a dictionary of basis functions that allows to produce a discretization in space of the decision process. We analyze the errors produced by this max-plus approximation method, for two function dictionaries, depending on the Lipschitz regularity of the value function. We then propose a simple strategy to adaptively choose function dictionaries, thus mitigating the curse of dimensionality. Finally, we show empirically on two low-dimensional examples that this max-plus discretization is more compact, in terms of the number of parameters, than a naive discretization of the continuous state problem.

This chapter has been published in the journal article:

> E. Berthier and F. Bach, "Max-Plus Linear Approximations for Deterministic Continuous-State Markov Decision Processes," in *IEEE Control Systems Letters*, vol. 4, no. 3, pp. 767-772, July 2020, doi:10.1109/LCSYS.2020.2973199.

Contribution 2: Estimation of stability regions for nonlinear dynamical systems.

In Chapter 4, we want to stabilize a dynamical system around one of its equilibrium points. This can be done using a linear-quadratic regulator (LQR). It is a matter of linearizing the system locally, and building a stabilizing closed-loop controller around this point. If the dynamical system is linear, the LQR controller computed at the equilibrium point is valid over the whole state space, and the system thus controlled will be globally asymptotically stable. If the system is nonlinear, this is only true locally, over a region called the region of attraction or stability region. An important problem in practice is to estimate the size of this region of attraction, especially for highly nonlinear systems. Concretely, this is done by finding a valid Lyapunov

function over this region. For polynomial systems, it is possible to certify the stability of a region by using polynomial optimization. Nevertheless, this results in a problem that is difficult to solve numerically for high dimensional systems. Indeed, since the estimation of stability regions is often only a sub-problem of a larger algorithm that calls it repeatedly, it must be performed quickly.

We propose two stability certificates that can be computed efficiently for high dimensional problems. They apply robustly to a class of systems whose first or second derivatives are bounded. By associating these certificates with an oracle allowing to compute bounds on the derivatives, we propose a simple algorithm for the estimation of stability regions. We then extend it to the trajectory tracking problem which generalizes stabilization around an equilibrium point. Finally, we experimentally validate this approach on polynomial and non-polynomial systems of various dimensions, including a robotic system that cannot be treated by the polynomial optimization method.

This chapter has been published in the conference article:

> E. Berthier, J. Carpentier and F. Bach, "Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems," *2021 European Control Conference (ECC)*, 2021, pp. 1412-1419, doi:10.23919/ECC54610.2021.9655071.

Contribution 3: Infinite dimensional sum-of-squares for optimal control.

In Chapter 5, we propose a new numerical approximation method for optimal control problems. This method applies to problems whose dynamics is unknown, and is observed only through a finite number of samples. In this sense, we place ourselves in the model-free paradigm of reinforcement learning. We consider the polynomial optimization method developed by Lasserre et al. (2008). This involves using the Lagrangian dual of the weak formulation of the control problem, which is a linear program in infinite dimension. In particular, it is necessary to represent the Hamiltonian as a non-negative function, *i.e.*, in the polynomial case, a non-negative polynomial. The moment-sum-of-squares hierarchy provides a numerical method based on semidefinite programming. However, this method is expensive in terms of computation time, and cannot be applied to high dimensional systems.

We propose to extend this method to systems that are not necessarily polynomial and are known only from samples. For this purpose, we use a representation of smooth positive functions derived from kernel methods, recently proposed by Marteau-Ferey et al. (2020). This representation is of infinite dimension, and is based on the function space of kernel methods, called a reproducing Hilbert kernel space. We prove that under regularity conditions, for control-affine systems, the representation of the Hamiltonian as a sum of squares in a space of smooth functions is exact. After subsampling, this method also leads to a semi-definite program, which we propose to solve with Newton's method. We illustrate this approach on an elementary control problem. We show in particular how to choose a function space adapted to the structure of the problem considered.

This chapter has been accepted for publication in the conference:

> E. Berthier, J. Carpentier, A. Rudi and F. Bach, "Infinite-dimensional Sums-of-Squares for Optimal Control," *Conference on Decision and Control*, 2022, arXiv:2110.07396.

Contribution 4: Analysis of the non-parametric temporal difference learning algorithm.

The temporal difference learning algorithm is a classical algorithm in reinforcement learning that allows to evaluate a given policy, incrementally from observations. In Chapter 6, we propose a non-asymptotic

analysis of the temporal difference algorithm, in its non-parametric and regularized version. It is an infinite dimensional generalization of the temporal difference algorithm with linear function approximation. It has been proved that this algorithm with linear function approximation converges, not to the value function, but to the fixed point of the projected Bellman operator, which is in general a different function. This is due to the limited approximation power of a linear approximation in finite dimension.

We show that if the non-parametric algorithm is used in a universal reproducing kernel Hilbert space, *i.e.*, a function space that is dense in the space of squared integrable functions, then the averaged iterates converge to the value function, even if it does not belong to the Hilbert space. We provide explicit convergence rates that depend on the relative regularity of the value function with respect to the function space. We treat both the case where the observations are independent and identically distributed, and the case where they come from a Markov chain. We illustrate this convergence on a numerical example of a continuous state Markov decision process.

This chapter has been accepted for publication in the conference:

> E. Berthier, Z. Kobeissi and F. Bach, "A Non-asymptotic Analysis of Non-parametric Temporal-Difference Learning," *Advances in Neural Information Processing Systems*, 2022, arXiv:2205.11831.

Chapter 1

Optimal Control & Reinforcement Learning

Abstract. *In this chapter, we give a concise introduction to optimal control and reinforcement learning. We present the main theoretical tools and numerical methods, and insist on some computational challenges that will be further discussed in subsequent chapters. This introduction is based on the reference books by Trélat (2005) and Liberzon (2011) for optimal control, and by Sutton and Barto (2018) for reinforcement learning.*

Contents

1.1 Optimal Control

1.1.1 Setting of the Problem

Let $\mathcal{X}$ and $\mathcal{U}$ be two sets, respectively called the state set and the control set. The state variable is denoted by x, and the control variable by u (coming from the Russian word for "control": управление, pronounced

upravlenie). We define a running cost function $L : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ (also called the Lagrangian) and a terminal cost function $M : \mathcal{X} \to \mathbb{R}$. Let T a positive number called the time-horizon. If the problem is well-defined (see below), then starting from $x_0 \in \mathcal{X}$, and using an input $(u(t))_{t\in[0,T]}$ in $\mathcal{U}$, we can define a unique trajectory $(x(t))_{t\in[0,T]}$ in $\mathcal{X}$, as a solution of the following ordinary differential equation (ODE):

$$x(0) = x_0, \text{ and } \forall t \in [0, T], \ \frac{\mathrm{d}x}{\mathrm{d}t}(t) = \dot{x}(t) = f(x(t), u(t)), \tag{1.1}$$

where $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is called the dynamics. There are different possible sets of assumptions on f and u such that the trajectory is well-defined, coming from the Cauchy-Lipschitz theorem. A sufficient condition is that f be continuous in u and continuously differentiable in x, $\frac{\partial f}{\partial x}$ continuous in u, and u piecewise continuous in t (Liberzon, 2011).

Optimal control is an infinite-dimensional optimization problem, namely the problem of finding a piecewise continuous input function $u : [0, T] \to \mathcal{U}$ such that, along with the generated trajectory, it minimizes a cost criterion over time. More precisely, the optimal control problem (OCP) is defined by:

$$\begin{aligned} &\inf_{u(\cdot)} \int_0^T L(x(t), u(t))\mathrm{d}t + M(x(T)) \\ &\text{s.t. } \forall t \in [0, T], \ \dot{x}(t) = f(x(t), u(t)) \\ &\quad x(0) = x_0. \end{aligned} \tag{1.2}$$

There are many variants of this formulation. If $M = 0$, the problem is said to be in Lagrange form, whereas if $L = 0$, it is in Mayer form. In principle, it is possible to rewrite any OCP in Lagrange or Mayer form at the expense of simple transformations. Furthermore, the cost functions and the dynamics may have an extra dependence on t, which can be simply recast in the previous form, by replacing the state variable x by (t, x) with dynamics $\dot{t} = 1$. Furthermore, the time-horizon T can be either fixed or free (in which case it is part of the optimization variables), and finite or infinite (hence requiring extra assumptions to ensure convergence of the integral cost). Terminal constraints of the form $h_T(x_T) \leq 0$ can be added to specify a fixed target point or set to the trajectory.

Finally, path constraints play an important role in practical applications, such as modelling collisions in robotics (Jallet et al., 2022). Each path constraint is typically expressed by:

$$\forall t \in [0, T], \quad h(t, x(t), u(t)) \leq 0. \tag{1.3}$$

When the constraint is simple and applies to the state or the control only, it can be encoded directly in the state or control set. For instance, if $h(t, x(t), u(t)) = \|u(t)\|^2 - 1$, then one can define $\mathcal{U} = \{\|u\| \leq 1\}$ without further constraint. However, some constraints, especially those which couple x and u, can be much more difficult to deal with, and often require specific attention.

Example 1.1: The double integrator

We model a car whose position x moves along a one-dimensional axis, with a controlled acceleration $\ddot{x} = u$, hence the *double integrator* name. Although the dynamics is a second-order ODE, it can be modeled by a first-order ODE. Indeed, let $x \in \mathcal{X} = \mathbb{R}^2$ and $u \in \mathcal{U} = [-1, 1]$, which means that the

car's acceleration is bounded. The dynamics can be equivalently defined by:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = f(x,u) = \begin{pmatrix} x_2 \\ u \end{pmatrix}. \tag{1.4}$$

We want to find a controller that stops the car at the origin in minimal time. This can be modeled by $L(x,u) = 1$, $M(x) = 0$, with a free time-horizon t_f but fixed terminal constraint $x(t_f) = (0,0)^\top$. The time-variable plays the role of a timer that stops at t_f as soon as $x(t_f) = 0$. In particular, the optimal value of the OCP is the optimal time required to stop the car at the origin. Such problems are commonly encountered and are called *minimal time problems*.

Sometimes, the state $x(t)$ is not easily observed. For instance, in the above example, we might be able to measure only the position of the car $x_1(t)$, but not its speed $x_2(t)$. In this case, the observer variable $y(t) = g(x(t), u(t))$ is observed instead of $x(t)$. The behavior of linear systems with linear observations, of the form:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t), \end{cases} \tag{1.5}$$

has been extensively studied in systems theory (Hespanha, 2018; Kalman, 1960). In particular, observability, *i.e.*, whether x can be reconstructed from observations of y, is characterized by a condition on the matrices (A, C). Another property of the system is its controllability, *i.e.*, whether there exists a u which can bring x to any target point in finite time. It can be evaluated by a similar condition on (A, B), which is why observability and controllability are sometimes seen as dual notions (Kwakernaak and Sivan, 1969, Chapter 1) (see also Kailath et al., 2000, Chapter 15). Note that there are extensions of such notions to linear systems with Gaussian noise, tackled by the field of Kalman filtering (Chui and Chen, 1987). We will present in more detail how to control fully-observed linear systems in Section 1.1.4. However, for non-linear systems, controllability and observability are much more challenging problems.

1.1.2 The Maximum Principle

We now assume that $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{U} \subset \mathbb{R}^p$. The Hamiltonian H is defined, for $x \in \mathcal{X}$, $u \in \mathcal{U}$, $p \in \mathbb{R}^d$ and $p_0 \in \mathbb{R}$, by:

$$H(x,u,p,p_0) = p^\top f(x,u) + p_0 L(x,u), \tag{1.6}$$

where p is called the co-state, and has the same dimension as x.

The maximum principle (Boltyanski et al., 1960) was first established in 1955 (Gamkrelidze, 1999) by Pontryagin and colleagues, hence the name Pontryagin's Maximum Principle (PMP). The PMP gives a necessary condition for the global optimality of a trajectory. In short, it states that, along optimal trajectories, the Hamiltonian H is maximized by the optimal control (see (1.9) below). More precisely, the optimal trajectory is characterized by three objects: the state $x(t)$, the control $u(t)$, and the co-state $p(t)$, which plays the role of a dual variable. In this sense, the PMP can be related to the Karush-Kuhn-Tucker (KKT) conditions for Lagrangian duality (Boyd and Vandenberghe, 2004), but it applies to the more complex, infinite-dimensional optimization problem of optimal control.

For conciseness, we state the maximum principle in its simplest form, for a free time-horizon, fixed-endpoint problem, with no terminal cost, as follows:

$$\begin{aligned} \inf_{u(\cdot),\, t_f} \int_0^{t_f} L(x(t), u(t))\mathrm{d}t \\ \text{s.t. } \forall t \in [0, t_f],\ \ \dot{x}(t) = f(x(t), u(t)) \\ x(0) = x_0, \quad x(t_f) = x_f. \end{aligned} \tag{1.7}$$

Theorem 1 (Maximum Principle (Liberzon, 2011))**.** *Let $u^* : [0, t_f] \to \mathcal{U}$ be an optimal control and $x^* : [0, t_f] \to \mathbb{R}^d$ the corresponding optimal state trajectory in (1.7). Then there exists a function $p^* : [0, t_f] \to \mathbb{R}^d$ and a number $p_0^* \leq 0$, such that $\forall t \in [0, t_f]$, $(p_0^*(t), p^*(t)) \neq (0, 0)$ and the following properties hold:*

(a) x^* and p^* satisfy the coupled canonical Hamiltonian equations:

$$\begin{aligned} \dot{x}^* &= \frac{\partial H}{\partial p}(x^*, u^*, p^*, p_0^*) \\ \dot{p}^* &= -\frac{\partial H}{\partial x}(x^*, u^*, p^*, p_0^*), \end{aligned} \tag{1.8}$$

with boundary conditions $x^*(t_0) = x_0$ and $x^*(t_f) = x_f$.

(b) For each fixed $t \in [0, t_f]$:

$$\forall u \in \mathcal{U}, \quad H(x^*(t), u^*(t), p^*(t), p_0^*) \geq H(x^*(t), u, p^*(t), p_0^*), \tag{1.9}$$

$$\text{and} \qquad H(x^*(t), u^*(t), p^*(t), p_0^*) = 0. \tag{1.10}$$

Note that for non-degenerate problems ($p_0^* \neq 0$), the abnormal multiplier p_0^* can be set to -1 because the above properties are invariant to the scaling of the Hamiltonian. Furthermore, if, instead of being fixed, the terminal state is constrained to lie on a parametric surface, the maximum principle is completed by *transversality* conditions, which constrain $p^*(t_f)$.

Even though the proof of the maximum principle in its full generality is not straightforward (Pontryagin et al., 1974; Trélat, 2005), an intuition can be obtained by the following – purely informal – derivation. In (1.7), let us introduce a dual variable $p(t)$ associated to the constraint $\dot{x}(t) = f(x(t), u(t))$. We can write the Lagrange dual problem. Assuming $t_f = T$ is fixed and that strong duality holds, and ignoring the boundary conditions, the problem is then equivalent to:

$$\sup_{p(\cdot)} \inf_{x(\cdot),\, u(\cdot)} \int_0^T L(x(t), u(t))\mathrm{d}t + \int_0^T p(t)^\top \Big(\dot{x}(t) - f(x(t), u(t))\Big)\mathrm{d}t. \tag{1.11}$$

We recognize the Hamiltonian $H(x, u, p) = p^\top f(x, u) - L(x, u)$ (with abnormal multiplier $p_0 = -1$), so that the problem is equivalent to:

$$\sup_{p(\cdot)} \inf_{x(\cdot),\, u(\cdot)} \int_0^T \Big(p(t)^\top \dot{x}(t) - H(x(t), u(t), p(t))\Big)\mathrm{d}t. \tag{1.12}$$

Hence we directly see that the optimal u maximizes the Hamiltonian, as in (1.9). Besides, the KKT conditions on p and x are respectively:

$$\dot{x}^* - \frac{\partial H}{\partial p}(x, u, p) = 0 \tag{1.13}$$

$$-\dot{p}^* - \frac{\partial H}{\partial x}(x, u, p) = 0, \tag{1.14}$$

using an integration by parts on the first term of (1.12). This recovers the canonical equations (1.8). Note that the optimality condition on u^* (1.9) is not written as a first order condition because of the generic form of the set $\mathcal{U}$. Indeed, the maximum in condition (1.9) can be reached on the boundary of $\mathcal{U}$.

Importantly, the maximum principle only gives necessary conditions for the optimality of a controller. Usually, this narrows down the search over the infinite-dimensional space of controls, sometimes to a finite-dimensional search space (Boscain and Piccoli, 2005). In very simple cases like the example below, it gives enough information to fully determine the optimal controller.

Example 1.2: Application of the maximum principle

We can apply the maximum principle to the double integrator, as defined in Example 1.1. Assuming that the problem is non-degenerate, the Hamiltonian writes:

$$H(x, u, p) = p_1 x_2 + p_2 u - 1. \tag{1.15}$$

Part (a) of the PMP leads to a system of four scalar differential equations:

$$\begin{cases} \dot{x}_1^*(t) = x_2^*(t) \\ \dot{x}_2^*(t) = u^*(t) \\ \dot{p}_1^*(t) = 0 \\ \dot{p}_2^*(t) = -p_1^*(t), \end{cases} \tag{1.16}$$

along with four boundary conditions fixing $x_1^*(0)$, $x_2^*(0)$, $x_1^*(t_f)$, and $x_2^*(t_f)$.
Part (b) of the PMP defines $u^*(t)$ as follows:

$$u^*(t) = \underset{u \in [-1,1]}{\operatorname{argmax}}\, H(x^*(t), u, p^*(t)) = \operatorname{sign}(p_2^*(t)). \tag{1.17}$$

Because t_f is a free variable, we need another equation. It is brought by the condition fixing the value of the Hamiltonian:

$$p_1^*(t) x_2^*(t) + p_2^*(t) u^*(t) - 1 = 0. \tag{1.18}$$

Combining (1.16) and (1.17), we obtain the fact that the trajectories have at most two pieces, depending on the sign of $p_2^*(t)$. On each piece, the position of the car $x_1^*(t)$ evolves as a polynomial of degree 2 of t, and the acceleration $u^*(t)$ is constant equal to -1 or 1. The fact that the control is piecewise constant, and hitting the boundaries of $\mathcal{U}$ is called the *bang-bang* phenomenon. For this control problem, this is quite intuitive: the optimal controller consists in going full throttle, and then full brakes (see Figure 1.1).

The *bang-bang* principle described above is a generic phenomenon for time-optimal control problems with

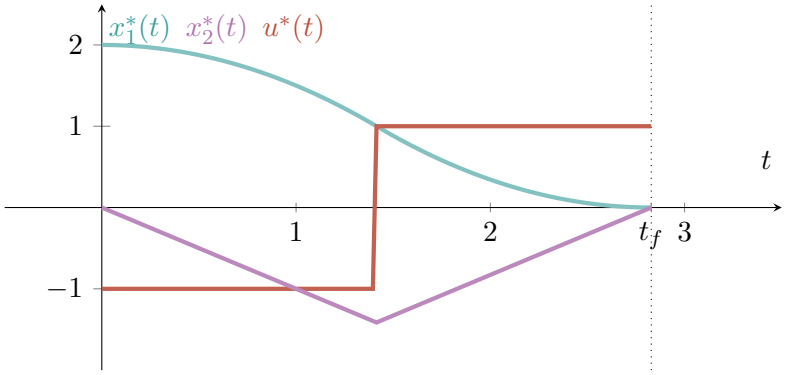


Figure 1.1: Optimal state and controller trajectories of the double integrator, for the initial condition $x(0) = (2, 0)^\top$. The controller is *bang-bang*, and the optimal value of the OCP is $t_f = 2\sqrt{2}$.

bounded control sets. As a consequence, one cannot expect much regularity for the optimal controller $t \mapsto u^*(t)$: in general this mapping is not even continuous. Furthermore, it can even switch very fast between different input values, and in the limit infinitely often (Fuller, 1963). This is called *chattering* and makes the theoretically optimal controller impossible to implement on a real system. Some form of regularization is then required (Caponigro et al., 2018).

1.1.3 The Hamilton-Jacobi-Bellman Approach

We now look at a different approach to optimality conditions for optimal control. It was developed in the United States, almost concurrently to the maximum principle which originated in the Soviet Union. It strongly relies on Bellman's principle of optimality (Bellman, 1954), at the basis of dynamic programming:

> *" An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decisions. "*

To exploit this principle, let us introduce the value function $V^* : [0, T] \times \mathcal{X}$ of the optimal control problem (1.2). It is defined as the optimal value of the OCP, starting from time and initial position, as follows, for $x_0 \in \mathcal{X}$ and $t \in [0, T]$:

$$\begin{aligned} V^*(t_0, x_0) = \inf_{u(\cdot)} & \int_{t_0}^{T} L(x(t), u(t))\mathrm{d}t + M(x(T)) \\ \text{s.t. } & \forall t \in [t_0, T], \ \ \dot{x}(t) = f(x(t), u(t)) \\ & x(t_0) = x_0. \end{aligned} \tag{1.19}$$

The optimal value function is also called the optimal cost-to-go, that is, the remaining cost starting from (t_0, x_0). Note that, in particular, for all $x \in \mathcal{X}$, $V^*(T, x) = M(x)$. Bellman's principle of optimality allows to compute V^* backwards in time from $V^*(T, \cdot)$, which is at the core of dynamic programming in discrete-time settings (Bertsekas, 2011). Such simple ideas can be extended to the continuous-time setting through infinitesimal changes in time, albeit with greater technical subtleties, as we will see later.

For now we assume that the optimal value function V^* of (1.2) is well-defined, finite and continuously differentiable in t and x. Let $(t_0, x_0) \in [0, T) \times \mathcal{X}$ and $\Delta t \in (0, T - t_0)$. Then, by definition, we have:

$$V^*(t_0, x_0) = \inf_{u:[t_0,T]\to\mathcal{U}} \int_{t_0}^{T} L(x(t), u(t))\mathrm{d}t + M(x(T)) \tag{1.20}$$

$$= \inf_{u:[t_0,T]\to\mathcal{U}} \int_{t_0}^{t_0+\Delta t} L(x(t), u(t))\mathrm{d}t + \int_{t_0+\Delta t}^{T} L(x(t), u(t))\mathrm{d}t + M(x(T)). \tag{1.21}$$

We can separate u into two variables $u|_{[t_0,t_0+\Delta t]}$ and $u|_{[t_0+\Delta T]}$. Using the principle of optimality, this is equivalent to:

$$V^*(t_0, x_0) = \inf_{u:[t_0,t_0+\Delta t]\to\mathcal{U}} \left\{ \int_{t_0}^{t_0+\Delta t} L(x(t), u(t))\mathrm{d}t + V^*(t_0 + \Delta t, x(t_0 + \Delta t)) \right\}. \tag{1.22}$$

In the limit $\Delta t \to 0$, we obtain (Lions, 2015)

$$V^*(t_0, x_0) = \inf_{u\in\mathcal{U}} (\Delta t) L(x_0, u) + V^*(t_0, x_0) + \Delta t \frac{\partial V^*}{\partial t}(t_0, x_0) + \Delta t \nabla V^*(t_0, x_0)^\top f(x_0, u) + o(\Delta t), \tag{1.23}$$

where ∇V^* denotes the gradient of V^* with respect to the variable x only. Dividing both sides by Δt, and taking the limit as $\Delta t \to 0$, this means that the optimal value function V^* is a solution of the following partial differential equation (PDE), called the Hamilton-Jacobi-Bellman (HJB) equation (Evans, 2010):

$$\begin{aligned} \forall (t, x) \in (0, T) \times \mathcal{X}, \quad & \frac{\partial V}{\partial t}(t, x) + \inf_{u\in\mathcal{U}} \left\{ L(x, u) + \nabla V(t, x)^\top f(x, u) \right\} = 0 \\ \forall x \in \mathcal{X}, \quad & V(T, x) = M(x). \end{aligned} \tag{1.24}$$

The fact that V^* is a solution of (1.24) is a necessary and sufficient (although we have not proved it here) optimality condition, contrary to the maximum principle, which is only a necessary condition. The main technical difficulty occurring with this approach is that in general V^* is not differentiable. Hence we need to consider weaker solution of (1.24). The right form of weak solutions to consider is *viscosity solutions*, which have been defined by Crandall et al. (1984) as follows. Let $\mathcal{X} = \mathbb{R}^d$, for some integer $d \geq 1$. A bounded, uniformly continuous function v is a viscosity solution of the HJB equation (1.24) if $v(T, \cdot) = M$ and for any $\phi \in \mathcal{C}^1((0, T) \times \mathbb{R}^d)$,

- if $v - \phi$ attains a local maximum at $(t_0, x_0) \in (0, T) \times \mathbb{R}^d$, then:

$$\frac{\partial \phi}{\partial t}(t_0, x_0) + \inf_{u\in\mathcal{U}} \left\{ L(x_0, u) + \nabla \phi(t_0, x_0)^\top f(x_0, u) \right\} \geq 0, \tag{1.25}$$

- and if $v - \phi$ attains a local minimum at $(t_0, x_0) \in (0, T) \times \mathbb{R}^d$, then:

$$\frac{\partial \phi}{\partial t}(t_0, x_0) + \inf_{u\in\mathcal{U}} \left\{ L(x_0, u) + \nabla \phi(t_0, x_0)^\top f(x_0, u) \right\} \leq 0. \tag{1.26}$$

The notion of viscosity solution captures the right notion of regularity for the value function of a control problem, in the sense that the following theorem holds.

Theorem 2 (Optimality condition for V^* (Evans, 2010))**.** *The value function V^* is the unique viscosity solution of the HJB equation* (1.24)*.*

The lack of differentiability of the value function is not anecdotal. It can occur for optimal control problems as simple as the following one-dimensional example.

Example 1.3: A non-differentiable value function

This example is taken from Liberzon (2011). Consider the following fixed-horizon OCP, for $\mathcal{X} = \mathbb{R}$ and $t_0 \leq t_1$:

$$V(t_0, x_0) = \inf_{u:[t_0,t_1]\to[-1,1]} x(t_1) \quad \text{s.t. } \forall t \in [t_0, t_1],\ \dot{x}(t) = x(t)u(t) \text{ and } x(t_0) = x_0. \tag{1.27}$$

If $x_0 > 0$, the best we can do is to bring x to 0 as fast as possible, by setting $u(t) = -1$, and then $x(t) = e^{-(t-t_0)}x_0$. If $x_0 < 0$, on the contrary the terminal cost is maximized by sending x to $-\infty$, that is, setting $u(t) = 1$ and $x(t) = e^{t-t_0}x_0$. The value function is then:

$$V(t_0, x_0) = \begin{cases} e^{t_1-t_0}x_0 & \text{if } x_0 < 0 \\ 0 & \text{if } x_0 = 0 \\ e^{-(t_1-t_0)}x_0 & \text{if } x_0 > 0. \end{cases} \tag{1.28}$$

This function, which is piecewise affine in the variable x_0, is obviously not differentiable at any point $(t_0, 0)$, for $t_0 < t_1$.

If we are given the optimal value function of an OCP, then we can recover an optimal controller at each time as a minimizer in the HJB equation:

$$u^*(t, x) \in \operatorname*{argmin}_{u\in\mathcal{U}} \left\{ L(x, u) + \nabla V^*(t, x)^\top f(x, u) \right\}. \tag{1.29}$$

Note that u^* is expressed as a function of t and x: it is called a closed-loop controller, as opposed to open-loop controllers obtained by the maximum principle. Having a closed-loop controller is a desirable property for real-life systems. Indeed, it allows to correct the estimation of the current state before computing the controller, hence reducing the effect of a possibly imperfect modeling of the dynamics. Conversely, the maximum principle generates a unique input sequence $u^*(t)$ which must be applied along the whole trajectory, without an opportunity for correction.

Comparing equations (1.9) and (1.29), the maximum principle and the HJB approach are related by the following equality:

$$p^*(t) = -\nabla V^*(t, x^*(t)). \tag{1.30}$$

Importantly, solving the HJB equation is equivalent to solving the optimal control problem for all initial conditions (t_0, x_0) at once. In comparison with the maximum principle, which concerns only one initial condition, and only provides necessary conditions of optimality, this is a much more powerful tool. However, globally solving a PDE is a daunting task: almost no PDE can be solved in closed form, and numerical methods do not scale beyond small dimensions (LeVeque, 1992). In comparison, the optimality conditions (1.8) brought by the PMP are ODEs, which are easier to handle numerically.

1.1.4 The Linear Quadratic Regulator

The linear quadratic regulator (LQR) is an extensively studied control problem at the basis of the theory of linear control (Kwakernaak and Sivan, 1969). Its solution can be computed explicitly, and hence it is of particular interest in practical applications. As we will see, it has many desirable properties, in terms of robustness, stability, ease of use and scalability. One must keep in mind that, even though the LQR only applies to linear systems, this technique is also ubiquitous in non-linear system theory through local linearizations of the system, and gave birth to algorithms such as differential dynamic programming (Mayne, 1966) or iterative LQR (Li and Todorov, 2007).

The time-varying, finite-horizon LQR problem is defined as follows. Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{U} = \mathbb{R}^p$, $T > 0$, and we define the following matrices, for $t \in [0, T]$: $A(t) \in \mathbb{R}^{d\times d}$, $B(t) \in \mathbb{R}^{d\times p}$, $Q(t) \in \mathbb{R}^{d\times d}$ such that $Q(t)^\top = Q(t) \succeq 0$, $R(t) \in \mathbb{R}^{p\times p}$ such that $R(t)^\top = R(t) \succ 0$, and $M \in \mathbb{R}^{d\times d}$ such that $M^\top = M \succeq 0$. We assume that A, B, Q and R are locally Lipschitz functions. We consider the problem:

$$\begin{aligned} &\inf_{u(\cdot)} \int_0^T \Big(x(t)^\top Q(t)x(t) + u(t)^\top R(t)u(t)\Big)\,\mathrm{d}t + x(T)^\top Mx(T) \\ &\text{s.t. } \forall t \in [0,T],\ \ \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ &\quad x(0) = x_0. \end{aligned} \tag{1.31}$$

Although they were presented above in the time-invariant case, both the maximum principle and the HJB optimality conditions can be readily extended to dynamics and costs depending on t. The Hamiltonian writes:

$$H(t, x, u, p) = p^\top(A(t)x + B(t)u) - x^\top Q(t)x - u^\top R(t)u. \tag{1.32}$$

Applying condition (b) of the maximum principle, we have:

$$u^*(t) \in \operatorname*{argmax}_u \left\{p^*(t)^\top B(t)u - u^\top R(t)u\right\}, \tag{1.33}$$

hence $u^*(t) = \frac{1}{2}R^{-1}(t)B(t)^\top p^*(t)$. Condition (a) gives the coupled ODEs:

$$\begin{cases} \dot{x}^*(t) &= A(t)x^*(t) + B(t)u^*(t) \\ \dot{p}^*(t) &= 2Q(t)x^*(t) - A(t)^\top p^*(t). \end{cases} \tag{1.34}$$

Using (1.33), this is equivalent to:

$$\begin{cases} \dot{x}^*(t) &= A(t)x^*(t) + \frac{1}{2}B(t)R^{-1}(t)B(t)^\top p^*(t) \\ \dot{p}^*(t) &= 2Q(t)x^*(t) - A(t)^\top p^*(t), \end{cases} \tag{1.35}$$

which can equivalently be written as:

$$\begin{pmatrix} \dot{x}^*(t) \\ \dot{p}^*(t) \end{pmatrix} = \mathcal{A}(t) \begin{pmatrix} x^*(t) \\ p^*(t) \end{pmatrix}, \tag{1.36}$$

for a suitable matrix $\mathcal{A}(t) \in \mathbb{R}^{2d\times 2d}$, sometimes called the Hamiltonian matrix (Liberzon, 2011), and (1.35) is the *extremal system*. Finally, the maximum principle contains an additional transversality condition (not detailed in Section 1.1.2) corresponding to the terminal cost: $p^*(T) = -2Mx^*(T)$. This suggests looking

for a similar linear relation between x^* and p^* for all t: $p^*(t) = -2P(t)x^*(t)$. Assuming that this relation holds, we can inject it back into (1.35), using the fact that $\dot{p}^*(t) = -2\dot{P}(t)x^*(t) - 2P(t)\dot{x}^*(t)$. After basic manipulations, we obtain that $P(t)$ must satisfy an ODE:

$$\dot{P}(t) = -Q(t) - A(t)^\top P(t) - P(t)A(t) + P(t)B(t)R^{-1}(t)B^\top(t)P(t), \tag{1.37}$$

called the Riccati differential equation (RDE), with boundary condition $P(T) = M$. This ODE has a unique solution, which is such that $P(t) \succeq 0$, $\forall t \in [0, T]$. The solution of the RDE is not known in closed form and is typically computed by standard numerical integration methods (Dormand and Prince, 1980), with a complexity that is polynomial in d. This means that the solution of an LQR can be computed efficiently for large dimensional problems, and does not suffer from the *curse of dimensionality*.

One can check that the RDE is also a sufficient optimality condition using the HJB equation. The relation (1.30) between p^* and V^* suggests to look for the following candidate value function:

$$V^*(t, x) = x^\top P(t)x. \tag{1.38}$$

The HJB equation is then verified if and only if: $\forall(t, x)$,

$$x^\top \dot{P}(t)x + x^\top Q(t)x + 2x^\top P(t)A(t)x + \underbrace{\min_u \left(u^\top R(t)u + 2x^\top P(t)B(t)u\right)}_{-x^\top P(t)B(t)R^{-1}(t)B(t)^\top P(t)x} = 0, \tag{1.39}$$

which is exactly equivalent to the RDE (1.37). The optimal controller computed by the HJB approach has the advantage of being expressed in closed-loop as:

$$u^*(t, x) = -R(t)^{-1}(t)B(t)^\top P(t)x. \tag{1.40}$$

A variant of the finite-horizon LQR is the time-invariant, infinite-horizon LQR:

$$\begin{aligned} &\inf_{u(\cdot)} \int_0^{+\infty} \Big(x(t)^\top Q x(t) + u(t)^\top R u(t)\Big)\, \mathrm{d}t \\ &\text{s.t. } \forall t \geq 0, \;\; \dot{x}(t) = Ax(t) + Bu(t) \\ &\quad x(0) = x_0. \end{aligned} \tag{1.41}$$

We assume that the pair (A, B) is controllable, *i.e.*, that the linear dynamical system can be brought to any x in finite time. It is well-known that the controllability of a linear system is equivalent to the Kalman rank condition (Kwakernaak and Sivan, 1969):

$$\text{rank}(B, AB, \ldots, A^{d-1}B) = d. \tag{1.42}$$

One can see (1.41) as the limit as $T \to +\infty$ of the finite-horizon problem (1.31). Although we state it informally, this argument can be used to prove that the optimal value function and controller are time invariant and have the closed form expressions:

$$V^*(x) = x^\top P x, \quad u^*(x) = -R^{-1}B^\top P x, \tag{1.43}$$

where P is the unique positive semi-definite solution (whose existence is ensured by the Kalman condition) of the algebraic Riccati equation (ARE), which is the limit of the solutions of the RDE when $T \to +\infty$:

$$0 = -Q - A^\top P - PA + PBR^{-1}B^\top P. \tag{1.44}$$

Therefore the optimal trajectories are of the form:

$$\dot{x}(t) = (A - BR^{-1}B^\top P)x(t), \tag{1.45}$$

and they all asymptotically converge to 0 regardless of $x(0)$, as soon as all the eigenvalues of $A - BR^{-1}B^\top P$ have negative real parts. This is verified if $Q \succ 0$ and the pair $(A, Q^{1/2})$ is observable (see the precise statement in (Liberzon, 2011, Theorem 6.1)). The fact that all controlled trajectories converge to 0 is called global asymptotic stability of the closed-loop system. This property can be asserted by exhibiting a Lyapunov function, *i.e.*, a non-negative function which decreases to 0 along the trajectories. The role of such functions will be discussed in more details in Chapter 4. Finally, let us mention that the behavior of the infinite-horizon LQR being recovered by studying the limit of the finite-horizon LQR when $T \to +\infty$ is a form of turnpike property (Samuelson, 1972). In particular, this effect has been studied by Trélat and Zuazua (2015) in a more general context, by looking at the limit behavior of the extremal system (1.35).

Beyond its use for control problems with linear dynamics and quadratic cost, the LQR can be used to model the local behavior of a non-linear system around an equilibrium point, *i.e.*, a point (x_e, u_e) such that $f(x_e, u_e) = 0$. Indeed, around this point, the dynamics is approximately linear:

$$f(x, u) \simeq \frac{\partial f}{\partial x}(x_e, u_e)(x - x_e) + \frac{\partial f}{\partial u}(x_e, u_e)(u - u_e). \tag{1.46}$$

It is also possible to linearize a non-linear system around a given trajectory. We will come back to this linearization technique in Chapter 4 and estimate the size of the region where the approximation is valid. Finally, let us mention that there exists a stochastic extension of the LQR called the linear quadratic Gaussian (LQG), which involves a Kalman estimator (Chui and Chen, 1987).

1.1.5 Numerical Methods

Beyond basic examples and the LQR, there is no hope to solve optimal control problems exactly. Many numerical methods have been developed to find approximate solutions. They mainly fall into three categories (Diehl and Gros, 2011; Rao, 2009; Trélat, 2005):

- direct methods, based on a discretization of the OCP (1.2) followed by solving a non-linear programming problem (NLP);
- indirect methods, solving a boundary value problem obtained by the maximum principle;
- methods solving the HJB partial differential equation.

Direct methods. Such methods (Betts, 2010) directly consider the original control problem (1.2). The simplest of them is called direct shooting (Diehl et al., 2006). The control is searched for in a finite-dimension space: it is typically piecewise constant (or parameterized as an affine function, or on a basis of splines...) on time intervals $0 = t_0 < t_1 < \ldots < t_N = T$, with steps $\delta_i = t_{i+1} - t_i$. The dynamics is discretized with some numerical scheme (Rao, 2009), *e.g.*, explicit Euler or Runge-Kutta. The problem is a nonlinear program, to which additional constraints can be added, of the form:

$$\min_{u_i, x_i} \sum_i \delta_i L(t_i, x_i, u_i) + M(x_N) \tag{1.47}$$

$$\text{s.t. } \forall i, u_i \in \mathcal{U} \text{ and } x_{i+1} = x_i + \delta_i f(t_i, x_i, u_i).$$

Problem (1.47) is typically solved with sequential quadratic programming (SQP) (Bonnans et al., 2006). The co-state p is usually obtained as a side product of solving problem (1.47), from the Lagrange multipliers. Overall, this approach is simple and does not require any prior knowledge on the problem. Note that only the time and control are explicitly discretized, not the state. Furthermore, the direct shooting method can be extended to the direct multiple shooting method which uses a subdivision of the time intervals along with continuity conditions on x (Bock and Plitt, 1984). An efficient implementation called `acados`, supporting algorithmic differentiation (Andersson et al., 2019), has been developed by Verschueren et al. (2022).

Another direct method is direct collocation (Von Stryk, 1993). The control is approximated by a piecewise linear function:

$$\forall t_i \leq t < t_{i+1}, \quad u_{coll}(t) = u(t_i) + \frac{t - t_i}{t_{i+1} - t_j} \left(u(t_{i+1}) - u(t_i)\right). \tag{1.48}$$

The state is approximated by a cubic spline:

$$\forall t_i \leq t < t_{i+1}, \quad x_{coll}(t) = \sum_{j=0}^{3} c_{i,j} \left(\frac{t - t_i}{h_i}\right)^j. \tag{1.49}$$

Collocation constraints ensure that the differential equation holds at the t_i and at the centers of the intervals. This choice of approximation functions and collocation points is called Lobatto cubic collocation. Other combinations are possible: pseudospectral methods (Ross and Karpenko, 2012) use orthogonal collocation (Rao, 2009), *i.e.*, roots of Chebyshev, Jacobi, or other orthogonal polynomials as collocation points, and the state is approximated by a global Legendre polynomial. Importantly, with collocation methods, both the state and control are discretized, contrary to direct shooting. The consistency of such numerical schemes is not straightforward and is assessed on a case by case basis. An implementation of direct collocation in `Julia` has been developed by Febbo et al. (2020).

All direct methods are based on a *discretize, then optimize* principle: they first discretize the control and/or the state, and then solve a non-linear program. Their main advantage is that they are very generic and do not require prior knowledge on the problem. However, they usually do not produce high-precision approximations, they can be memory-intensive and they only produce locally optimal solutions, the NLP being *a priori* non-convex (Trélat, 2005).

Indirect methods. Indirect methods are based on an *optimize, then discretize* principle: they derive optimality conditions, namely the maximum principle, which are then discretized. The indirect shooting method (Bonnans, 2019; Trélat, 2005) is defined as follows. Suppose that the time horizon T is fixed. The second condition of the maximum principle (1.9) usually allows to find the optimal control $u^*(t)$ as a function of $z(t) = (x^*(t), p^*(t))$. Injecting this expression into the ODE given by the first condition (1.8), we obtain an *extremal* system, of the form $\dot{z}(t) = F(t, z(t))$. Lastly, the boundary conditions given by the transversality condition on $p^*(t)$ (the additional condition in the maximum principle), and constraints on $x^*(t)$ can be represented by $R(z(0), z(T)) = 0$. The optimal state and co-state couple z is a solution of the boundary value problem:

$$\begin{cases} \forall t \in [0, T], \quad \dot{z}(t) = F(t, z(t)) \\ R(z(0), z(T)) = 0. \end{cases} \tag{1.50}$$

Assuming that the solution of the Cauchy problem $\dot{z}(t) = F(t, z(t)),\ z(0) = z_0$ is provided by an oracle $\tilde{z}(t, z_0)$, then the problem is to find a zero of $G(z_0) = R(z_0, \tilde{z}(T, z_0))$. This can be achieved by Newton or quasi Newton methods (Bonnans et al., 2006; Boyd and Vandenberghe, 2004). Roughly, one "shoots" an initial condition z_0, integrates it to obtain $\tilde{z}(T, z_0)$, checks whether the boundary conditions are verified, and finally corrects z_0 if it is not the case. A classical analogy is an archer shooting an arrow, and successively correcting its angle by looking at the final position of the arrow around the target. Overall, indirect shooting requires to integrate numerically (LeVeque, 1992) the ODE $\dot{z}(t) = F(t, z(t))$, and to compute the Jacobian of G. This can be achieved by using a differentiable numerical integrator, *i.e.*, exploiting automatic differentiation to compute $\frac{\partial \tilde{z}}{\partial z_0}(T, z_0)$, as allowed by libraries such as the `SciML` software ecosystem by Rackauckas et al. (2020).

The simple indirect shooting method can be refined to the indirect multiple shooting method. The time interval $[0, T]$ is divided into N sub-intervals at commutation or junction times (free or fixed), with continuity conditions on z at these times. This subdivision is known to stabilize the method. We now have to find a zero of a function G on a space of dimension proportional to the number of time intervals N, but the integration of the first-order ODE can be parallelized. A variant is indirect collocation (Diehl and Gros, 2011), a shooting method where the state and control are parameterized with piecewise polynomials.

Indirect shooting provides precise approximations, but usually Newton's method has a small convergence domain and is very sensitive to boundary conditions misspecification. In practice, indirect shooting can be used to refine a nearly optimal trajectory, previously computed by a direct method. Newton's method is also often associated to a continuation method (Allgower and Georg, 2003), which solves a sequence of problems of increasing difficulty by varying a parameter, hence providing better initializations if the solution varies smoothly with the parameter. A drawback of indirect shooting is that the maximization condition defining u^* (1.9) must be derived for each particular class of problems. Finally, indirect methods only compute controls in open-loop (u is expressed as a function of t only). The generated trajectory, sometimes called nominal, can then be stabilized by computing a local linear-quadratic control feedback around it, as discussed in Section 1.1.4.

Solving the HJB PDE. A different approach, sometimes classified as a direct method (Trélat, 2005), is to find approximate solutions to the HJB equation (1.24). Solving PDEs is known to be a hard problem, especially in large dimensions, and is a field on its own. Methods that compute wavefronts, *i.e.*, level sets of the value function, are usable in low-dimensional problems (Sethian, 1999).

Other methods are based on various discretization schemes, such as finite differences (Fleming and Soner, 2006), which require a time discretization and space meshing. The space discretization is subject to the curse of dimensionality, in the sense that the size of the discretization grows exponentially with the dimension. Another approach is to discretize the state variable with finite elements (Munos, 2000; Munos and Moore, 2002). This constructs a deterministic Markov decision process (MDP), which can be solved with dynamic programming (see Section 1.2.2). The max-plus finite element method (Akian et al., 2008) is also an alternative approach to approximate a solution of HJB (see Chapter 3).

An advantage of the HJB approach is that the optimal control is computed in closed-loop (u is expressed as a function of t and x, unlike *e.g.*, indirect shooting). On the other hand, the obtained precision is generally not sufficient for critical applications, since any space discretization is subject to the curse of dimensionality.

1.2 Reinforcement Learning

1.2.1 Problem Statement

The field of reinforcement learning (RL) is dedicated to solving a problem which can be defined informally as follows. An agent must learn how to behave in an unknown environment, in order to maximize a long-term reward. The agent navigates between different states of the environment (think of different positions on a map), and the environment sends a reward that depends on the current state and the action taken. The way that the agent moves between states is also determined by the current state and the action taken, along with some randomness (see Figure 1.2).

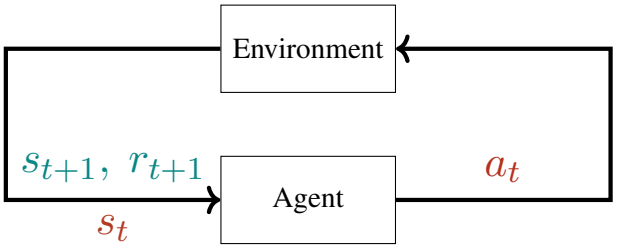


Figure 1.2: Description of the interaction with the environment: at time t, the agent is in state s_t and chooses the action a_t. The environment sends back a reward r_{t+1} and a new state s_{t+1}, which will be used by the agent at time $t+1$.

More formally, we can define the reinforcement learning problem within the framework of discounted Markov decision processes (MDP) (Sutton and Barto, 2018). An MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, in which:

- $\mathcal{S}$ is a finite set of states;
- $\mathcal{A}$ is a finite set of actions;
- $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ represents the state-transition probabilities, so that for $a \in \mathcal{A}$, $(s, s') \in \mathcal{S}^2$:

$$p(s'|s, a) = \mathbb{P}\left[s_{t+1} = s'|s_t = s, a_t = a\right]. \tag{1.51}$$

  Since $\mathcal{S}$ and $\mathcal{A}$ are finite, p can be represented by a tensor P of dimension $|\mathcal{A}| \times |\mathcal{S}| \times |\mathcal{S}|$, such that for all a, P^a is a transition matrix of size $|\mathcal{S}| \times |\mathcal{S}|$, and $P^a_{i,j}$ represents the probability of going to state j, when the agent is in state i and has done action a. All the entries of P^a are non-negative and its rows sum to one.

- $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function. $r(s, a, s')$ is the reward obtained through the transition from state-action (s, a) to state s'. In particular, the expected reward for the state-action (s, a) is (with a slight abuse of notations):

$$r(s, a) = \mathbb{E}[r_{t+1}|s_t = s, a_t = a] = \sum_{s'} p(s'|s, a) r(s, a, s'). \tag{1.52}$$

  Again, since $\mathcal{S}$ and $\mathcal{A}$ are finite, the values of r can be stored in a tensor of dimension $|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{S}|$. We generally assume the rewards to be bounded, to ensure convergence of discounted cumulative rewards.

- $\gamma \in [0, 1)$ is a discount factor. It describes a preference for immediate rewards compared to delayed rewards. Complete indifference to the future corresponds to $\gamma = 0$.

An MDP is Markovian in the sense that the transition probabilities and rewards do not depend on the past transitions, but only on the current state and action (s_t, a_t). Hence the environment is Markovian. Of course, the agent is not Markovian, in the sense that it has a memory and that it can use all of the past information it has received to choose its next action. This process can be called "policy synthesis", that is, building a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maps the current state to an action. In full generality, a policy can be stochastic, but, when looking for optimal policies (see below), we can restrict ourselves to deterministic ones (Bellman, 1957a). Importantly, in the reinforcement learning paradigm, apart from the discount factor, the parameters of the MPD are unknown to the agent.

Assume that the initial state is fixed to $s \in \mathcal{S}$. The aim of the agent is to find a policy which maximizes its expected discounted cumulative reward:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \middle| s_0 = s \right]. \tag{1.53}$$

In the finite case with bounded rewards, an optimal memoryless deterministic policy π^* exists (Bertsekas, 2011), and it is such that:

$$\pi^*(s) \in \operatorname*{argmax}_{\pi:\mathcal{S}\to\mathcal{A}} V^\pi(s), \tag{1.54}$$

and we define the optimal value function of the MDP as:

$$V^*(s) = \max_{\pi:\mathcal{S}\to\mathcal{A}} V^\pi(s) = V^{\pi^*}(s).$$

The optimal policy and the optimal value function are the direct counterparts of the optimal controller and value function (1.19) in optimal control (see Section 1.1). For any policy π, the function V^π is simply called the value function of π. Similarly, we define the action-value function, or Q-function as follows:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \middle| s_0 = s, a_0 = a \right], \tag{1.55}$$

and the optimal Q-function as $Q^*(s, a) = \max_\pi Q^\pi(s, a)$. The Q-function computes the expected reward of choosing a specific action a starting from a state s, and then following the policy π.

A useful property, resulting from the Markov property, is that such functions are invariant to the starting time from which the cumulative rewards are considered. Indeed, for any $t_0 \geq 0$ and $s \in \mathcal{S}$:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \middle| s_0 = s \right] = \mathbb{E}_\pi \left[\sum_{t=0}^{+\infty} \gamma^t r_{t_0+t+1} \middle| s_{t_0} = s \right]. \tag{1.56}$$

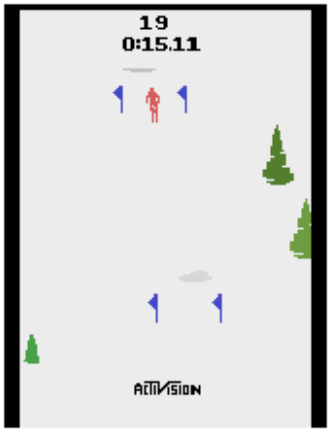


Figure 1.3: A sample state from "Skiing-V4".

Example 1.4: An MDP modeling a video game

"Skiing-v4" is an environment defined in the `gym` library (Brockman et al., 2016). It describes a 1980 Atari game with a finite-state and action MDP. The state is the image displayed by the game emulator (see Figure 1.3). It is a 210×160 picture with 3 color channels, taking integer values in $[0, 255]$. There are 3 actions: left, right and neutral, which shift the skier's lateral direction accordingly. The aim is to ski downhill, past 20 gates, as fast as possible. Hitting an obstacle or missing a gate gives a time penalty. The reward is minus the number of seconds spent from the last state (including penalties). It should be noted that the dynamics of the skier is computed internally by the game emulator, in a *black-box* manner. In particular, contrary to many optimal control problems, one cannot access the dynamics in closed form nor compute its gradients.

Note that we have formally presented discounted, infinite-horizon, discrete MDPs. The MDP formalism is flexible and several variations can be considered as well. This includes undiscounted average-cost MDPs (Puterman, 2014), episodic MDPs (Sutton and Barto, 2018) (which is the case of Example 1.4), continuous sets of states or actions (Powell and Ma, 2011), and even continuous-time MDPs (Guo and Hernández-Lerma, 2009). Therefore, the discrete formulation is not a fundamental difference between optimal control and reinforcement learning. Nor is the stochastic aspect of RL, as one can also consider stochastic control problems (Fleming and Rishel, 2012). Our claim here is rather that reinforcement learning is a more general problem than optimal control, precisely because the environment is unknown to the agent.

Link with optimal control. Consider a deterministic MDP, and assume that this MDP is known to the agent, *i.e.*, all the transitions and rewards are known. Then the problem reduces to an optimal control problem, in discrete-time, and with discrete state and control. The cost corresponds to the opposite of the reward, the states coincide, and the controller corresponds to the action. Of course, maximizing the reward is replaced by minimizing the cost. The dynamic programming approach presented in Section 1.1.3 can be readily applied to such RL problems. However, the maximum principle approach is usually not applied to RL problems, because of stochasticity, and the search for closed-loop policies.

Link with online learning. Consider an MDP, unknown to the agent, and reduced to only one state. This describes a well-known online learning problem: the multi-armed bandit (Berry and Fristedt, 1985), that can be briefly described as follows. At each time step, the agent must choose between k possible actions (k different arms or slot machines). The chosen arm i sends a random reward r, drawn from an unknown probability distribution p_i associated to arm i. The actor must discover which of the k arms has the probability distribution with the highest expectation $\mathbb{E}_{p_i}[r]$, only by successively actioning different arms. This can be modeled by a slight extension (with non-deterministic rewards) of the above MDP framework, with a degenerate probability transition which loops on only one state, k actions representing the k arms, and random reward functions $r(s,a)$. Note that the problem can also be represented within the exact presented MDP framework, by introducing k virtual states and supporting the randomness on the state-transitions. An important feature of the multi-armed bandit problem is that the agent must, at the same time, maximize its reward and discover the best arm. In other words, it must *exploit*, *i.e.*, repeatedly action the arm which it believes to be the best, and *explore*, *i.e.*, action the different arms to discover potentially better arms. The exploration / exploitation trade-off is at the core of the online learning paradigm and, consequently, of reinforcement learning. To tackle this issue, online learning offers a set of tools backed by strong theoretical guarantees, such as ε-greedy strategies, or optimism in the face of uncertainty (see Lattimore and Szepesvári (2020) for a complete exposition). Yet, one must admit that many RL algorithms move away from this truly online paradigm. Indeed, it is often assumed that finding the optimal policy is done offline, that is, without worrying about collecting bad rewards during the training phase.

1.2.2 Dynamic Programming

In this section, we first briefly describe the dynamic programming approach to RL, in the case where the model is known to the agent. It is similar to the HJB approach presented in Section 1.1.3, except that we write the same principles in discrete time. As before, using Bellman's optimality principle, we can split the search of an optimal policy between the choice of the first action a_0, and an optimal policy afterwards:

$$\begin{aligned} V^*(s) &= \max_{\pi} \ \mathbb{E}_{\pi}\left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \Big| s_0 = s\right] \\ &= \max_{a\in\mathcal{A}} \left\{ \mathbb{E}\left[r_1 \Big| s_0 = s, a_0 = a\right] + \mathbb{E}_{\pi^*}\left[\sum_{t=1}^{+\infty} \gamma^t r_{t+1} \Big| s_0 = s, a_0 = a\right] \right\} \\ &= \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s'\sim p(s'|s,a)} V^*(s') \right\}. \end{aligned} \tag{1.57}$$

The last equation (1.57) is called the Bellman equation. It is the counterpart of the HJB equation in discrete-time. Many RL algorithms exploit dynamic programming to find exact or approximate solutions of this equation, or variants thereof. The simplest one is the value iteration algorithm. Let us consider the Bellman operator $T : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$, defined by:

$$\text{for } V \in \mathbb{R}^{\mathcal{S}}, s \in \mathcal{S}, \quad (TV)(s) = \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \mathbb{E}_{s'\sim p(s'|s,a)} V(s') \right\}. \tag{1.58}$$

One can prove that this affine operator is a γ-contraction mapping for the infinity norm, hence it has a unique fixed point V^*. The value iteration algorithm computes a sequence with an arbitrary V_0 and, for $k \geq 1$:

$$V_{k+1} = TV_k. \tag{1.59}$$

One can prove that this sequence converges linearly to V^* as follows. Exploiting successively the fact that $TV^* = V^*$ and the contraction property, then for $k \geq 1$:

$$\begin{aligned} \|V_k - V^*\|_\infty &= \|T(V_{k-1} - V^*)\|_\infty \\ &\leq \gamma \|V_{k-1} - V^*\|_\infty \\ &\leq \gamma^k \|V_0 - V^*\|_\infty. \end{aligned} \tag{1.60}$$

In practice, this algorithm is simple to implement and exhibits fast convergence. The optimal policy can be recovered from V^*, by computing the argmax in (1.57). Note that a variant of value iteration, policy iteration, exploits the Bellman equation to progressively refine a policy (Sutton and Barto, 2018). Yet, effectively running the value iteration algorithm requires:

1. knowing the rewards and transition probabilities of the MDP, *i.e.*, being able to express the Bellman operator (1.58);
2. computing the Bellman operator, *i.e.*, being able to compute efficiently each iteration (1.59).

In many RL problems, both issues should not be taken lightly. First, as discussed above, in general the rewards and transition probabilities are unknown to the agent. Hence the reward and expectation in (1.57) must be somehow estimated. Second, each iteration requires to store $|\mathcal{S}|$ values (one for each state), and to make $|\mathcal{S}| \times |\mathcal{A}|$ computations. This can be overwhelming in large MDPs, and in particular in MDPs issued from a discretization of a continuous control problem. Yet, this cost is unavoidable, as one cannot expect to solve *exactly* the MDP in less than $|\mathcal{S}| \times |\mathcal{A}|$ operations, *i.e.*, its reading size. Moreover, if the MDP has a continuous state space $\mathcal{S}$, such iterations cannot be represented in finite dimension, because they involve functions with infinite support. Therefore, like in supervised learning (Shalev-Shwartz and Ben-David, 2014), one must resort to *estimation* and *approximation*. We detail both notions in the following two sections.

1.2.3 Dynamic Programming with Estimation

The dynamics and rewards being unknown, the Bellman operator cannot be computed explicitly and must be estimated. There are two alternative paradigms to deal with this issue: *model-based* reinforcement learning, and *model-free* reinforcement learning.

Model-based RL. The basic idea is to first learn a model of the state-transitions and rewards, and then solve an optimal control problem with this model (*e.g.*, with value iteration). Learning the model from observations of sample transitions and rewards of the MDP is a regression problem. For instance, in Example 1.4, given many observations of the game, one could learn to predict the next frame given the current frame and the action. In particular, this would mean uncovering the hidden dynamics of the skier (pressing left makes it go to the left and so on), and predicting the positions of the next gates. The reward function must be modeled as well, *i.e.*, learning that hitting the trees or missing gates is harmful. This learning process is feasible up to a certain precision, and generally only locally, because of the complexity and the randomness of the environment. This surrogate model of the MDP is then used to solve an optimal control problem. The main challenge with this approach is to understand the interplay between the surrogate model's uncertainty and the performance of the policy planned from this model. This challenging issue is tightly related to the notions of

robust (or H_∞) control (see Safonov (2012) for an historical review), and certainty equivalence (Bar-Shalom and Tse, 1974) for the stochastic aspect. Model-based RL is a rich subfield of RL that goes way beyond the simplistic method described above. We refer the reader to the recent survey by Moerland et al. (2020).

Model-free RL. In this paradigm, one tries to solve the RL problem without creating an explicit model of the MDP. This allows abstracting from the above difficulties, and hence constitutes the most prominent approach to RL. The main tool of model-free RL is temporal-differences (TD). For simplicity and because we will study this algorithm in Chapter 6, we present temporal-differences with the TD(0) algorithm (also called TD-learning). It is an algorithm for policy evaluation, a subproblem of RL which computes the value function V^π of a given (possibly suboptimal) policy π. To compute an optimal policy, it must be combined with a policy improvement process, such as policy gradient in actor-critic methods (Sutton et al., 1999). Temporal-differences are ubiquitous in model-free RL: they are also at the basis of other algorithms, among which Q-learning and SARSA (Sutton and Barto, 2018).

Let $\pi : \mathcal{S} \to \mathcal{A}$ a given policy. The aim of policy evaluation is to compute V^π, the unique solution of the fixed-point equation:

$$\forall s \in \mathcal{S}, \quad V^\pi(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim p(s'|s,\pi(s))} V^\pi(s'), \tag{1.61}$$

i.e., $V^\pi = T^\pi V^\pi$, where T^π is a modified Bellman operator defined by:

$$\text{for } V \in \mathbb{R}^{\mathcal{S}}, s \in \mathcal{S}, \quad (T^\pi V)(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim p(s'|s,\pi(s))} V(s'). \tag{1.62}$$

An analog of value iteration (1.59) for T^π would be:

$$\forall s \in \mathcal{S}, \quad V_{k+1}(s) = r(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim p(s'|s,\pi(s))} V_k(s'). \tag{1.63}$$

However, since r and p are unknown, the update must be estimated from observations. In TD(0), the observations come from a trajectory generated according to the policy π, thus providing a sequence $(s_k, r_k)_{k \geq 1}$, where $s_{k+1} \sim p(s'|s_k, \pi(s_k))$ and $r_k = r(s_k, \pi(s_k))$. At each iteration, instead of computing (1.63), TD(0) makes an incremental step as follows (Sutton, 1988):

$$V_{k+1}(s_k) = V_k(s_k) + \rho_{k+1} \Big[r_k + \gamma V_k(s_{k+1}) - V_k(s_k) \Big], \tag{1.64}$$

and the other entries of V_{k+1} are copied from V_k. In this update, contrary to (1.63), only one entry of V_k is updated at each iteration. The update uses a step $\rho_k > 0$, and the term inside the brackets in (1.64) is called a *temporal-difference* (TD). In expectation, the temporal-difference is null when $V = V^*$. Alternatively, the same update can be equivalently written as:

$$V_{k+1}(s_k) = (1 - \rho_{k+1}) V_k(s_k) + \rho_{k+1} \left(r_k + \gamma V_k(s_{k+1}) \right), \tag{1.65}$$

highlighting the proximity with (1.63). In particular, if $\rho_{k+1} = 1$ and the MDP is deterministic, then (1.65) is just one of the updates of (1.63). Otherwise, the temporal-difference provides an estimate of its expectation, called the Bellman error.

The convergence of the tabular (*i.e.*, $\mathcal{S}$ is a finite set) TD(0) algorithm is proved by Sutton (1988) under certain sampling schemes, with three central arguments (see Chapter 6 for a more thorough exposition). First, the step size sequence ρ_k must satisfy classical conditions for stochastic approximation (Borkar and

Meyn, 2000). Second, all states s_k must be sampled sufficiently often. Third, one must prove stability of the scheme, given the fact that the update in (1.65) (the rightmost term in parentheses) contains the previous approximation V_k. Relying on such previous iterates is usually called *bootstrapping*. It can be shown that as soon as $\gamma < 1$, this effect is benign and does not damage convergence. Although this method relies on estimation, asymptotically this is an exact method which converges to the true value function V^π.

1.2.4 Dynamic Programming with Function Approximation

When the number of states $|\mathcal{S}|$ is large, or when it is infinite (think of $\mathcal{S} = [0, 1]$ for example), the tabular methods presented above are not tractable. In particular, if $|\mathcal{S}|$ is too large to fit in memory, it is hopeless to look for an exact solution. Considering Example 1.4, the size of the discrete state-space is $|\mathcal{S}| = 256^{210\times160\times3} = 2^{806400}$. Of course, there are much better modeling options. For instance, one could model the state space by a vector of real values with $d = 210 \times 160 \times 3 = 100800$ entries in $[0, 1]$, *i.e.*, $\mathcal{S} = [0, 1]^d$, where the $[0, 1]$ interval continuously maps the 256 possible discrete values. The state space can be even more compressed, *e.g.* by using a gray-scale image, or by down-sampling the resolution. However, this is already an approximation, more or less likely to harm the performance of the policy. Similarly, one could be tempted to transform a control problem of the form (1.2) into an MDP by a naive discretization. This would also result in a very large $|\mathcal{S}|$, exponential in the state dimension. This phenomenon is known as the curse of dimensionality, and we will illustrate this issue in Chapter 3.

Overall, when the problem is too hard to solve exactly, one must resort to approximations. Such approximations implicitly assume that the problem has more structure than a purely combinatorial one. In our example, this could mean *e.g.*, that states where the gates are at similar positions play similar roles. Some model-free algorithms, originally designed for tabular settings, have been adapted to handle approximations, such as TD-learning and Q-learning. The value function V_θ is selected in a parametric space, with parameter $\theta \in \Theta \subset \mathbb{R}^{\tilde{d}}$, with typically $\tilde{d} \ll \dim \mathcal{S}$. One can use linear function approximation, *i.e.*, $V_\theta = \theta^\top \psi$, for some function $\psi : \mathcal{S} \to \mathbb{R}^{\tilde{d}}$, or more complicated parameterizations with neural networks (leading to the deep Q-learning algorithm). However, convergence results are scarce with non-linear parameterizations. The TD(0) iterations with function approximation only update the parameter θ, and write:

$$\theta_{k+1} = \theta_k + \rho_{k+1}\Big[r_k + \gamma V_{\theta_k}(s_{k+1}) - V_{\theta_k}(s_k)\Big]\nabla_\theta V_{\theta_k}(s_k). \tag{1.66}$$

Such iterations are tractable even with large dimensional parameters θ, including neural networks thanks to automatic differentiation (Paszke et al., 2019). In the case of linear function approximation, the convergence of TD(0) is well-understood (Bhandari et al., 2018; Tsitsiklis and Van Roy, 1997). The algorithm converges to the unique fixed-point of the projected Bellman operator $\Pi_\Theta \circ T^\pi$, where Π_Θ is the L^2 orthogonal projection onto $\mathcal{F}_\Theta = \{\theta^\top \psi \mid \theta \in \Theta\}$. This cannot be equal to V^π as soon as $V^\pi \notin \mathcal{F}_\Theta$, but the quality of the approximation should improve as the size of $\mathcal{F}_\Theta$ increases. We revisit this question in Chapter 6, where we study a non-parametric variant of TD(0) with linear function approximation. In particular, we prove that under generic conditions, this version converges to V^π without approximation error.

Our analysis is inspired by the link between TD-learning and stochastic gradient descent (SGD) (Shalev-Shwartz and Ben-David, 2014). Indeed, when $\gamma = 0$ (indifference to the future), the update (1.66) is the same as a step of SGD on the optimization problem:

$$\min_\theta \frac{1}{2n}\sum_{k=1}^{n}\Big(V_\theta(s_k) - r(s_k, \pi(s_k))\Big)^2. \tag{1.67}$$

Minimizing this mean squared error with SGD is a well-studied problem. In particular with linear function approximation, it reduces to a simple least-squares problem, and one can benefit from extensive analyses (Pillaud-Vivien, 2020). By this digression, we have incidentally shown that policy evaluation, a subproblem of RL, contains the problem of regression, at the core of supervised learning. Again, this confirms that RL is a very general paradigm.

1.2.5 The Linear Programming Formulation

The linear programming (LP) formulation of RL (Bertsekas, 2011; Puterman, 2014) was historically one of the first approaches to be considered. Like direct methods in optimal control (see Section 1.1.5), it directly converts the original problem into a solvable optimization problem, here, a linear program. Let e the vector of dimension $|\mathcal{S}|$ with all entries equal to one. It can be shown that the RL problem is equivalent to the following LP formulation:

$$\begin{aligned} &\min_{v \in \mathbb{R}^{|\mathcal{S}|}} \; e^\top v \\ &\text{s.t. } \forall (s,a) \in \mathcal{S} \times \mathcal{A},\, v(s) \geq r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} v(s'). \end{aligned} \tag{1.68}$$

Note that for simplicity, since $\mathcal{S}$ is a finite set, we have used the same notation v for the function and for the vector representing its $|\mathcal{S}|$ values. Hence v is sought as the smallest supersolution of the Bellman equation, and at optimality it corresponds to V^*. Of course, this formulation cannot fit in memory when $|\mathcal{S}|$ gets very large, but for reasonable values of $|\mathcal{S}|$, the corresponding LP can be solved efficiently with powerful and scalable solvers.

There is a similar formulation of optimal control problems (1.2), called the *weak formulation* of optimal control. It requires extra convexity assumptions on the dynamics to be equivalent to the original formulation. We use this weak formulation with the notations of optimal control in Chapter 5, but note that the method presented in this chapter could indifferently be applied to RL problems.

1.3 Comparison

In the summarizing Table 1.1, we list the main notations and results of optimal control and their counterpart in reinforcement learning. Note that since Pontryagin's maximum principle has no direct counterpart in reinforcement learning, we only list the results related to dynamic programming. For simplicity, we consider the infinite-horizon, discounted, time-invariant setting for both problems.

	Optimal Control	Reinforcement Learning
Time variable	$t \in \mathbb{R}_+$	$t \in \mathbb{N}$
State variable	$x(t)$	s_t
Control / action	$u(t)$	a_t
Closed-loop policy	$u(t) = \bar{u}(x(t))$	$a_t \sim \pi(a\|s = s_t)$
Dynamics	$\dot{x}(t) = f(x(t), u(t))$	$s_{t+1} \sim p(\cdot \mid s = s_t, a = a_t)$
Cost / negative reward	$L(x(t), u(t))$	$-r(s_t, a_t, s_{t+1})$
Discount factor	$\eta > 0$	$\gamma \in [0, 1)$
Cumulative cost	$\int_0^\infty e^{-\eta t} L(x(t), u(t)) \mathrm{d}t$	$-\sum_{t=0}^\infty \gamma^t r(s_t, a_t, s_{t+1})$
Optimization problem	$\min_{u(\cdot)} \int_0^\infty e^{-\eta t} L(x(t), u(t)) \mathrm{d}t$ s.t. $\forall t,\ \dot{x}(t) = f(x(t), u(t))$	$\max_\pi \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t r(s_t, \pi(s_t), s_{t+1})\right]$ s.t. $\forall t,\ s_{t+1} \sim p(s'\|s_t, \pi(s_t))$
Value function	$V^u(x) = \int_0^\infty e^{-\eta t} L(x(t), u(t)) \mathrm{d}t$ s.t. $x(0) = x$	$V^\pi(s) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t, s_{t+1}) \| s_0 = s]$
Optimal value function	$V^*(x) = \min_u V^u(x)$	$V^*(s) = \max_\pi V^\pi(s)$
Bellman's optimality principle	$\rho V^*(x) = \min_u \{L(x, u) + \nabla V(x)^\top f(x, u)\}$	$V^*(s) = \max_a r(s, a) + \gamma \mathbb{E}[V^*(s')]$
Hamiltonian / Q-function	$H^*(x, u) = L(x, u) + \nabla V^*(x)^\top f(x, u)$	$Q^*(s, a) = r(s, a) + \gamma \mathbb{E}[V^*(s')]$
Optimal policy	$\bar{u}^*(x) \in \mathrm{argmin}_u H^*(x, u)$	$\pi^*(s) \in \mathrm{argmax}_a Q^*(s, a)$
Value iteration	$V_{k+1} = \frac{1}{\rho} \min_u \{L(\cdot, u) + \nabla V_k^\top f(\cdot, u)\}$	$V_{k+1} = \max_a r(\cdot, a) + \gamma \mathbb{E}_{s' \sim p(\cdot, a)}[V_k(s')]$
Linear programming formulation	$\max_V \rho \int V(x) \mathrm{d}\mu_0(x)$ s.t. $\forall (x, u),\ -\rho V(x) + L(x, u)$ $+\nabla V(x)^\top f(x, u) \geq 0$	$\min_V (1 - \gamma) \int V(s) \mathrm{d}\mu_0(s)$ s.t. $\forall (s, a),\ V(s) \geq r(s, a) + \gamma \mathbb{E}[V(s')]$

Table 1.1: Correspondence between the main concepts of optimal control and reinforcement learning.

Chapter 2

Conceptual & Numerical Tools

Abstract. *In this chapter, we present different tools that will be used in subsequent chapters. These tools have not been developed specifically for optimal control or reinforcement learning applications, but we try to highlight the most obvious connections that will be further developed in the rest of this thesis. Although we have chosen to present them in the most convenient order, the four parts of this chapter can be read mostly independently. In the table below, we detail the tools required in each of the next chapters of the thesis.*

		Chapter 3	Chapter 4	Chapter 5	Chapter 6
Section 2.1:	Rigid-body Dynamics		✓		
Section 2.2:	Polynomial Optimization		✓	✓	
Section 2.3:	Kernel Methods			✓	✓
Section 2.4:	Max-Plus Algebra	✓			

Contents

2.1 Rigid-Body Dynamics

This section is based on the reference book by Siciliano et al. (2008), and in particular on the chapter dedicated to dynamics by Featherstone and Orin (2008). We use the presented notions in the experiments of Chapter 4, to apply our stability assessment method to a robotic arm. The visualizations of the robotic systems are produced using the graphical interface from Lamiraux and Mirabel (2014).

2.1.1 The Configuration Space

In order to model a robotic task within the optimal control or reinforcement learning frameworks, we must be able to define a state variable representing the current state of the robot, and a dynamics describing its evolution in time. Describing the position (kinematics) and motion (dynamics) of a robot is often much more complicated than for basic systems like pendulums. A robot is composed of a collection of rigid bodies connected by joints, some of the joints being actuated (controlled). The joints between the links form a graph describing the robot, and most of the time we can assume that it is a tree. There are many ways to represent the relative positions of each rigid body: they typically involve 6-dimensional vectors (3 dimensions for translations and 3 dimensions for rotations) when the Plücker coordinate system is used, but other representations involve 4×4 matrices, or elements of a Lie group (Arnold, 1966; Murray et al., 2017).

The most convenient coordinate system for control tasks is called the *configuration space*. An element $q \in \mathbb{R}^d$ of the configuration space describes every joint variable in the mechanism, *i.e.*, the angle of a revolute joint (like a door handle) or the displacement of a prismatic joint (like a slider). For instance, the double pendulum in Figure 2.1 is composed of two rigid bodies and a fixed base (in yellow at the bottom). There are two revolute joints between the base and the first arm of the pendulum, and between the first and second arm, each one orthogonal to the same plane. The configuration space is the torus $(-\pi, \pi]^2$, and the displayed position is represented by the configuration $q = (\pi/6, -\pi/3)^\top$.

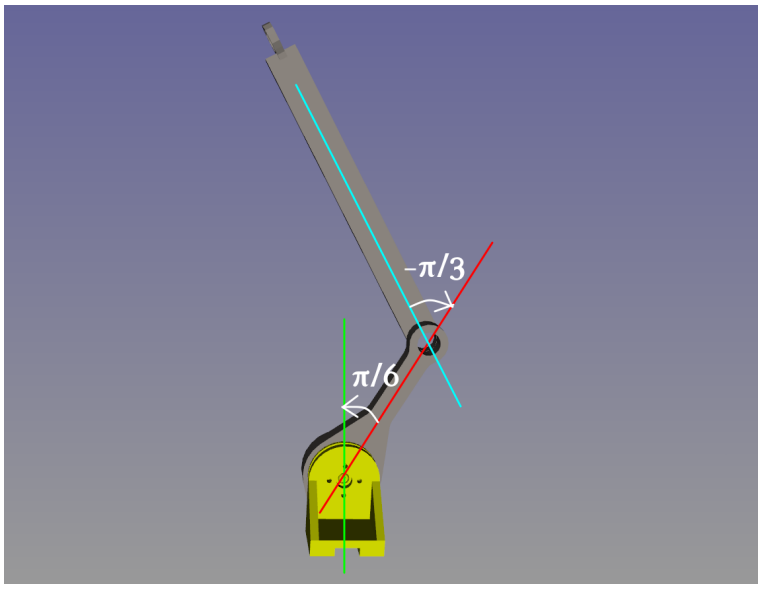


Figure 2.1: The double pendulum is a robot with two rigid bodies and two revolute joints.

2.1.2 Inverse and Forward Dynamics

The Newton-Euler equation of motion, *i.e.*, Newton's law for a robot, can be expressed in the configuration space under the form:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = \tau, \tag{2.1}$$

where q, $\dot{q}$ and $\ddot{q}$ are the joint position, velocity and acceleration, and τ represents the force (or the torque) exerted on the joints. For a fully actuated robot, τ can be the controller u. H is the inertia matrix, C is a matrix containing the Coriolis and centrifugal terms, and τ_g is the vector of gravity forces. The former three terms depend on the design of the robot, and on the current position and velocity. Because the robot is composed of several rigid bodies, closed-form expressions are complicated to obtain.

Let $u = \tau$ and $x = (q, \dot{q})$. The Newton-Euler equation (2.1) is an implicit expression defining the dynamics of the robot:

$$\dot{x} = f(x, u). \tag{2.2}$$

Going from (2.1) to an analytic expression of (2.2) is not usually feasible, but there are algorithms that perform such computations numerically. From now on, we assume that the structure of the robot, called its kinematic graph, is a kinematic tree. This greatly simplifies the implementation of the algorithms presented below.

Inverse Dynamics. Inverse dynamics is the task of, given a configuration q and velocity $\dot{q}$, computing the torque τ required to obtain a given acceleration $\ddot{q}$. This amount to computing the right-hand side of the Newton-Euler equation (2.1). This is called *inverse* dynamics because the computation is from the acceleration to the force, contrary to classical dynamics equation which compute the acceleration from the forces. An efficient implementation of inverse dynamics in robots is the recursive Newton-Euler algorithm (RNEA) (Luh et al., 1980). This algorithm is recursive in the sense that it makes several passes of computations on the kinematic tree of the robot.

Forward Dynamics. Forward dynamics is the task of, given a configuration q and velocity $\dot{q}$, computing the acceleration $\ddot{q}$ generated by a certain torque τ. This task is achieved by the articulated body algorithm (ABA) (Featherstone, 1983). Obtaining the acceleration $\ddot{q}$ allows simulating the system with numerical integration, or computing the dynamics $\dot{x} = f(x, u)$ at one given $x = (q, \dot{q})$ and $u = q$. Accessing the forward dynamics is at the basis of all control problems.

Example 2.1: Dynamics of a Humanoid Robot

We consider the TALOS robot (Stasse et al., 2017), an articulated humanoid robot with the corpulence of an adult (1.75m, 95kg). The configuration space is a subset of $\mathbb{R}^{51}$, representing the articulated joints. A representation of a random configuration of the robot is shown in Figure 2.2. It corresponds to the following configuration vector q_0:

```
q0 = [  0.05414991, -0.64573404,  0.85573156, -0.71946408, -0.18806232,
       -0.2723181 ,  0.61061187,  0.87659227,  0.20979564, -0.5873395 ,
        1.06451821,  0.33373042,  0.08134467,  0.1984546 , -0.29140879,
        0.1109362 ,  0.97965779, -0.52531908, -0.42150787,  0.77398604,
        0.14491015,  0.18993392,  2.68267067,  1.6946712 , -0.37361567,
        1.07785106,  0.37048131,  0.02197035, -0.39305706,  0.52611338,
        0.60613105,  0.70355   ,  0.03086743, -0.2342755 ,  0.62801123,
       -0.45788523, -2.45905674, -1.47550817, -0.49455662,  1.73054626,
        0.64479222, -0.43015432, -0.34902351,  0.32776189,  0.11055924,
        0.93036471,  0.10705843, -0.4993781 ,  0.28307897, -0.01074671,
       -0.55570481 ]
```

The dimension $d = 51$ of the configuration space can be considered large from the viewpoint of optimal control. In particular, polynomial optimization methods (see Section 2.2) usually cannot handle such dimensions, nor can direct shooting methods based on discretizations. However, in reinforcement learning (and in machine learning in general), this is not considered a particularly high-dimensional problem.

Because of gravity, in general a robot cannot sustain a particular position without injecting torque in its joints. This is the case for any humanoid robot or human which would collapse even when standing on the ground, if no torque or muscle control is used. Given the position q_0, a velocity $v_0 = 0$ and acceleration $a_0 = 0$, we can compute the required torque to stay in position q_0 using inverse dynamics. The RNEA (as well as the ABA used below) is implemented in the `pinocchio` library (Carpentier et al., 2019), which provides the following simple command:

```
u0 = pinocchio.rnea(robot.model, robot.data, q0, v0, a0)
```

This provides an equilibrium point such that $f(x_0, u_0) = 0$, where $x_0 = (q_0, v_0)$. Conversely, one can check that the acceleration generated by the torque (or control) u_0 is indeed equal to zero, using the ABA algorithm:

```
a = pinocchio.aba(robot.model, robot.data, q0, v0, u0)
print(max(abs(a)))
>>> 7.815970093361102e-14
```

In Chapter 4, we will use the RNEA and ABA algorithms to identify equilibrium points. We will also use the derivatives of the dynamics, *i.e.*, of the ABA algorithm, to compute a local linearization of the dynamics. This is allowed by the implementation in `pinocchio` which supports analytical derivatives and automatic differentiation.

Figure 2.2: The randomly sampled configuration q_0 of the TALOS robot.

2.2 Polynomial Optimization

Polynomial or semi-algebraic optimization is a relatively recent field with deep theoretical foundations in algebraic geometry, offering practical numerical optimization methods. We first present the main ideas of polynomial optimization, applied to constrained optimization in Section 2.2.1, and then present an application to stability assessment problems in Section 2.2.2 and optimal control problems in Section 2.2.3. Stability assessment with polynomial optimization will be used as a baseline method in Chapter 4. The polynomial optimization approach to solving optimal control problems will be extended to non-polynomial smooth problems in Chapter 5. The current section is based on the reference books by Lasserre (2010, 2015), and on the lecture notes by Henrion (2014).

2.2.1 Optimization of Polynomials on Semi-algebraic Sets

Let us first consider the following optimization problem:

$$f^* = \min_{x \in \mathbb{R}^n} f(x) \;\text{ s.t. }\; \forall j \in \{1, ..., n_X\},\; g_j(x) \geq 0, \tag{2.3}$$

where f and the g_j, for $j \in \{1, ..., n_X\}$ are polynomials. Let X be a basic semialgebraic set, *i.e.*, a set defined by polynomial inequalities: $X = \{x \in \mathbb{R}^n \mid \forall j \in \{1, ..., n_X\},\; g_j(x) \geq 0\}$. We assume that X is compact, and look for the global minimum of (2.3). The compactness of X can be enforced by adding an extra constraint of the form $g(x) = R^2 - \|x\|^2 \geq 0$. We denote by $\mathbb{R}[x]$ the set of polynomials in the variable x.

2.2.1.1 Representation of Non-negative Functions as Sums-of-Squares

An equivalent characterization of the global minimum of f on X is:

$$f^* = \sup\{\lambda \mid f(x) - \lambda \geq 0, \forall x \in X\}. \tag{2.4}$$

This formulation is a linear program in the scalar variable λ. However, it requires to handle a dense set of inequality constraints, or equivalently, to represent the non-negative polynomial function $x \mapsto f(x) - \lambda$. Several certificates of positivity are available for polynomials, among which Krivine's and Putinar's Positivstellensätze. The first one involves the resolution of a linear program, however it has some drawbacks due to ill-conditioning and non-exactness at finite degree. We only present Putinar's Positivstellensatz, which leads to a semi-definite program, as follows.

One can construct a non-negative polynomial by taking a sum-of-squares of other polynomials:

$$\forall x \in \mathbb{R}^n, \quad p(x) = \sum_{i=1}^{k} p_i(x)^2 \geq 0. \tag{2.5}$$

If p verifies (2.5) where the p_i are polynomials, we say that p is a sum-of-squares (SoS) of polynomials, or simply a SoS, and p is of course non-negative. Putinar's Positivstellensatz gives a sufficient condition under which, conversely, a non-negative polynomial can be represented as a SoS of polynomials.

Theorem 3 (Putinar's Positivstellensatz (Putinar, 1993))**.** *Assume that for some $j \in \{1, \ldots, n_X\}$, the set $\{x \in \mathbb{R}^n \mid g_j(x) \geq 0\}$ is compact. Let p be a polynomial that is strictly positive on X. Then there exist $(\sigma_j)_{0 \leq j \leq n_X}$, each of which being a SoS of polynomials, such that:*

$$\forall x \in \mathbb{R}^n, \quad p(x) = \sigma_0(x) + \sum_{j=1}^{n_X} \sigma_j(x) g_j(x). \tag{2.6}$$

One can easily check in this expression that for $x \in X$, $p(x) \geq 0$. Note that this theorem does not provide the degree of the SoS polynomials σ_j, which can be larger than the degree of p, due to possible cancellations.

Let us now apply this representation to the polynomial $x \mapsto f(x) - \lambda$ in our optimization problem (2.4). For the sake of explanation and insightfulness, we use Putinar's Positivstellensatz as a justification for the numerical method presented below. Note that it not a rigorous argument (namely because $f - f^*$ is not a strictly positive polynomial), but we refer to Lasserre (2010) for a formal construction with moments. Assume that one of the sets $\{x \in \mathbb{R}^n \mid g_j(x) \geq 0\}$, for $j \in \{1, \ldots, n\}$ is compact. If $f - \lambda > 0$, then, by Putinar's Positivstellensatz, there exists SoS polynomials $\sigma_0, ..., \sigma_{n_X}$ such that:

$$f(x) - \lambda = \sigma_0(x) + \sum_{j=1}^{n_X} \sigma_j(x) g_j(x). \tag{2.7}$$

An important property of SoS polynomials is that they can be represented in a convenient way with a linear parameterization, using a positive semi-definite matrix. Indeed, suppose that σ is a SoS polynomial of degree $2d$ (SoS polynomials necessarily have an even degree). Let $v_d(x) = (1, x_1, ..., x_n, x_1^2, ..., x_n^d)$ the vector of monomials of degree less than d, with an n dimensional variable x. To each polynomial $h(x)$ of degree less than d, we can associate its vector of coefficients $\mathbf{h}$ in the basis $v_d(x)$. The vector has size $s_n(d) := \mathrm{C}_{n+d}^n$ and is indexed by $\mathbb{N}_d^n := \{\alpha \in \mathbb{N}^n : |\alpha| \leq d\}$. Then:

$$\sigma(x) = \sum_{i=1}^{k} h_i(x)^2 = \sum_{i=1}^{k} (\mathbf{h}_i^\top v_d(x))^2$$

$$= \sum_{i=1}^{k} v_d(x)^\top \mathbf{h}_i \mathbf{h}_i^\top v_d(x) = v_d(x)^\top \underbrace{\sum_{i=1}^{k} \mathbf{h}_i \mathbf{h}_i^\top}_{Q \succeq 0} v_d(x) = v_d(x)^\top Q v_d(x). \tag{2.8}$$

From now on, when there is no ambiguity, we will use the same notation for a polynomial and its coefficients in a canonical basis adapted to its degree. The coefficients $(\sigma_\alpha)_{\alpha \in \mathbb{N}^n_{2d}}$ of the polynomial $\sigma(x)$ are such that: $\sigma(x) = \sum_{\alpha \in \mathbb{N}^n_{2d}} \sigma_\alpha x^\alpha$. We can compute the real symmetric $s_n(d) \times s_n(d)$ matrices $(B_\alpha)_{\alpha \in \mathbb{N}^n_{2d}}$ such that

$$v_d(x) v_d(x)^\top = \sum_{\alpha \in \mathbb{N}^n_{2d}} B_\alpha x^\alpha.$$

Then for all x, $\sigma(x) = v_d(x)^\top Q v_d(x)$ if and only if:

$$\forall x,\ \sigma(x) = \mathrm{Tr}(v_d(x)^\top Q v_d(x)) = \mathrm{Tr}(Q v_d(x) v_d(x)^\top) = \mathrm{Tr}(Q \sum_{\alpha \in \mathbb{N}^n_{2d}} B_\alpha x^\alpha) \tag{2.9}$$

$$\iff \forall x,\ \sum_{\alpha \in \mathbb{N}^n_{2d}} \sigma_\alpha x^\alpha = \sum_{\alpha \in \mathbb{N}^n_{2d}} \mathrm{Tr}(Q B_\alpha) x^\alpha \tag{2.10}$$

$$\iff \forall \alpha \in \mathbb{N}^n_{2d},\ \sigma_\alpha = \langle B_\alpha, Q \rangle. \tag{2.11}$$

Consequently, checking whether the polynomial $\sigma(x)$ is SoS is equivalent to the feasibility of the following SDP:

$$\text{Find } Q \succeq 0 \text{ such that } \forall \alpha \in \mathbb{N}^n_{2d},\ \sigma_\alpha = \langle B_\alpha, Q \rangle. \tag{2.12}$$

The matrix Q in the SDP has size $s_n(d) \times s_n(d)$ and there are $s_n(2d)$ constraints.

Going back to the Positivstellensatz, the problem is also an SDP, now involving several matrices corresponding to the SoS polynomials in the decomposition (2.6). For $j \in \{1, ..., n_X\}$, let $d_j = \lceil \deg(g_j)/2 \rceil$. We want to check that $f(x) - \lambda$ (of degree d_0) has Putinar's representation with an *a priori* upper-bound $2(r - d_j)$ on the degree of σ_j, for $r \geq d_0/2$. This bound ensures that all the polynomials involved in (2.6) have a degree less than $2r$. Let $g_0(x) = 1$, and $f(x) = \sum_\alpha f_\alpha x^\alpha = f^\top v_{d_0}$. For each $j \in \{0, ..., n_X\}$, we compute the real symmetric matrices $(C^j_\alpha)_{\alpha \in \mathbb{N}^n_{2r}}$, of size $s_n(r - d_j) \times s_n(r - d_j)$, such that:

$$g_j(x) v_{r-d_j}(x) v_{r-d_j}(x)^\top = \sum_{\alpha \in \mathbb{N}^n_{2r}} C^j_\alpha x^\alpha. \tag{2.13}$$

With the same computations as above, we obtain the following SDP:

$$\begin{aligned} \sup_{\lambda,\ X_j \succeq 0,\ j=0,...,n_X} \quad & \lambda \\ \text{s.t. } \forall \alpha \in \mathbb{N}^n_{d_0}, \quad & f_\alpha - \lambda \mathbf{1}_{\alpha=0} = \sum_{j=0}^{n_X} \langle C^j_\alpha, X_j \rangle \\ \forall \alpha \in \mathbb{N}^n_{2r} \setminus \mathbb{N}^n_{d_0}, \quad & 0 = \sum_{j=0}^{n_X} \langle C^j_\alpha, X_j \rangle. \end{aligned} \tag{2.14}$$

This SDP has $n_X + 1$ matrix variables of sizes $s_n(r - d_j) \times s_n(r - d_j)$, and there are $s_n(2r)$ constraints. The integer r defines an upper-bound on the degree of the terms in Putinar's representation. We remind

that Theorem 3 gives no information on this degree. Let us define as $\mathfrak{S}_r$ the class of polynomials with the decomposition described above. The SDP (2.14) can be summarized by:

$$f_r^{\text{SOS}} = \sup\{\lambda \mid f - \lambda \in \mathfrak{S}_r\}. \tag{2.15}$$

It is a tightening of the original problem (2.4):

$$f^* = \sup\{\lambda \mid f - \lambda \geq 0\}. \tag{2.16}$$

Hence f_r is a lower-bound on f^*. The principle of Lasserre's hierarchy is to solve (2.15) for increasing values of r. This hierarchy converges to f^*, and for most polynomials the tightening is exact after a finite rank r. This approach also provides optimality certificates as stopping criteria, based on testing the rank of matrices (we refer to Lasserre (2015) for a complete exposition). SoS verification tools such as `SOSTOOLS` (Papachristodoulou et al., 2021) allow to solve problems of the form (2.15), without having to model the SDP (2.12) explicitly. Lasserre's hierarchy is also called the *Moment-SoS* hierarchy, referring to both this SoS formulation, and its dual formulation with moments that we present below.

2.2.1.2 Lasserre's Hierarchy on Moments

Problem (2.3) can be re-written as a linear program in terms of measures (Henrion, 2014):

$$\begin{aligned} f^* = \inf_{\mu} & \int_X f(x)\mu(\mathrm{d}x) \\ \text{s.t.} & \int_X \mu(\mathrm{d}x) = 1,\ \mu \in \mathcal{M}_+(X), \end{aligned} \tag{2.17}$$

where $\mathcal{M}_+(X)$ denotes the set of measures on the set X. This problem has the same value as (2.3): it is attained for any μ^* putting all its mass on minimizers of f on X. For a measure $\mu \in \mathcal{M}_+(X)$, its moment of order $\alpha \in \mathbb{N}^n$ is defined by:

$$y_\alpha = \int_X x^\alpha \mu(\mathrm{d}x). \tag{2.18}$$

Consider a sequence of moments $(y_\alpha)_{\alpha \in \mathbb{N}^n}$. The Riesz functional associated to y is the linear functional:

$$\ell_y : \sum_\alpha p_\alpha x^\alpha \in \mathbb{R}[x] \mapsto \sum_\alpha p_\alpha y_\alpha. \tag{2.19}$$

If μ has moments $(y_\alpha)_{\alpha \in \mathbb{N}^n}$, *i.e.*, if y represents μ, we have that:

$$\ell_y(p(x)) = \sum_\alpha p_\alpha y_\alpha = \sum_\alpha p_\alpha \int_X x^\alpha \mu(\mathrm{d}x) = \int_X p(x)\mu(\mathrm{d}x). \tag{2.20}$$

The moment matrix of order d associated to y is the symmetric matrix $M_d(y)$ such that, if $p(x)$ has degree d,

$$\ell_y(p(x)^2) = \mathbf{p}^\top M_d(y)\mathbf{p}, \tag{2.21}$$

where $\mathbf{p}$ is the vector of monomials of $p(x)$. The matrix $M_d(y)$ has size $\mathrm{C}_{n+d}^n \times \mathrm{C}_{n+d}^n$ and is a linear function of y. In particular, $M_d(y)_{\alpha,\beta} = \ell_y(x^\alpha x^\beta) = y_{\alpha+\beta}$. The localizing matrix $M_d(q,y)$ is, given a polynomial $q(x)$, the symmetric matrix such that for $p(x)$ of degree d, $\ell_y(q(x)p(x)^2) = \mathbf{p}^\top M_d(q,y)\mathbf{p}$. $M(q,y)$ is a

bilinear function of (q, y). Finally, $M(y)$ and $M(q, y)$ are the generalized versions of the previous matrices for polynomials $p(x)$ of infinite degree ($d \to \infty$).

Since f is a polynomial, problem (2.17) can be re-written with moments:

$$\begin{aligned} f^* = \inf_{\mu, y} \sum_{\alpha} f_\alpha y_a \\ \text{s.t. } y_0 = 1,\ y \text{ has representing measure } \mu \in \mathcal{M}_+(X). \end{aligned} \tag{2.22}$$

Under the same conditions as Theorem 3, the dual formulation of Putinar's theorem (Henrion, 2014) states that y has a representing measure in $\mathcal{M}_+(X)$ if and only if

$$M(y) \succeq 0, \text{ and } M(g_j, y) \succeq 0,\ \forall j \in \{1, ..., n_X\}. \tag{2.23}$$

Note that the positive semi-definite operators are countably infinite dimensional operators, and must be interpreted as limits of finite-dimensional matrices. Then, using this result, problem (2.22) is equivalent to:

$$\begin{aligned} f^* = \inf_{y} \sum_{\alpha} f_\alpha y_a \\ \text{s.t. } y_0 = 1, M(y) \succeq 0, \text{ and } M(g_j, y) \succeq 0,\ \forall j \in \{1, ..., n_X\}. \end{aligned} \tag{2.24}$$

The decision variable y has infinite dimension, and all the constraints are linear matrix inequalities (LMI) in y. Lasserre's LMI hierarchy, or the moment hierarchy, is a series of relaxations of the former LMIs, that gives increasing lower-bounds $f_r^{\text{mom}} \leq f_{r+1}^{\text{mom}}$ on f^*. Indeed, when r grows, the problem has more constraints and hence its optimal value increases. Let $d_j = \lceil \deg(g_j)/2 \rceil$, for $j \in \{0, ..., r_X\}$. For $r \geq \max\{\max_j d_j, 1\}$, a relaxation of order r is:

$$\begin{aligned} f_r^{\text{mom}} = \inf_{y} \sum_{\alpha} f_\alpha y_a \\ \text{s.t. } y_0 = 1, M_r(y) \succeq 0, \text{ and} \\ M_{r-d_j}(g_j, y) \succeq 0,\ \forall j \in \{1, ..., n_X\}. \end{aligned} \tag{2.25}$$

Like (2.14), this is solved with an SDP solver. The size of the SDP is the number of monomials of degree $2r$, *i.e.*, C_{n+2r}^n. The sequence of $(f_r^{\text{mom}})_r$ converges to f^*, and there is finite convergence for generic data choices (the set where it does not hold has measure zero), that is, for some r the relaxation is exact and there is a computable certificate on the rank of the PSD matrices that certifies it. There is also an algorithm using Cholesky factorization to extract minimizers of f from y^*: the optimal measure is a sum of atoms which are all global minimizers. All of this is included in the dedicated software `GloptiPoly` (Henrion et al., 2009).

2.2.1.3 Duality Between the Moment and SoS Formulations

The non-negative formulation (2.4) is the Lagrange dual (Boyd and Vandenberghe, 2004) of the measure formulation (2.17). This can be shown easily as follows. The primal is an LP on measures:

$$\begin{aligned} p^* = \inf_{\mu} \langle f, \mu \rangle \\ \text{s.t. } \langle 1, \mu \rangle = 1,\ \mu \geq 0. \end{aligned} \tag{2.26}$$

The dual is also an LP (with an infinite number of constraints):

$$\begin{aligned} d^* &= \sup_{g\geq 0,\,\lambda} \inf_{\mu} \langle f,\mu\rangle + \lambda(1-\langle 1,\mu\rangle) - \langle g,\mu\rangle \\ &= \sup_{g\geq 0,\,\lambda} \inf_{\mu} \langle f-\lambda-g,\mu\rangle + \lambda \\ &= \sup_{g\geq 0,\,\lambda} \lambda \quad \text{s.t. } f-\lambda = g \\ &= \sup_{f-\lambda\geq 0} \lambda \\ &= \sup\{\lambda \mid \forall x, f(x)-\lambda \geq 0\}. \end{aligned} \tag{2.27}$$

For this problem, strong duality holds and $p^* = d^*$ as soon as one of the problems is feasible. Both problems are relaxed with a hierarchy of SDPs that use the basis of monomials of size $\mathrm{C}^n_{n+r} \leq \min(r^n, n^r)$, which is polynomial in n or r. However, practical computations can be quickly limited in large dimensions (n large). The finite-dimensional relaxations are also dual to each other: the dual of (2.25) is (2.14). Indeed, let $g_0(x) = 1$, the relaxed primal at rank r is:

$$\begin{aligned} p_r^* &= \inf_y \sum_\alpha f_\alpha y_a \\ \text{s.t. } y_0 &= 1, M_{r-d_j}(g_j, y) \succeq 0,\ \forall j \in \{0,...,n_X\}. \end{aligned} \tag{2.28}$$

Its Lagrange dual writes:

$$\begin{aligned} d_r^* &= \sup_{X_j\succeq 0,\,\lambda} \inf_y \sum_\alpha f_\alpha y_a + \lambda(1-y_0) - \sum_j \langle M_{r-d_j}(g_j,y), X_j\rangle \\ &= \sup_{X_j\succeq 0,\,\lambda} \lambda + \inf_y \mathcal{L}_{f_\alpha, X_j, \lambda}(y), \end{aligned} \tag{2.29}$$

where $\mathcal{L}_{f_\alpha,X_j,\lambda}$ is a linear form in y that must necessarily be equal to zero for all y. Let us express the right constraint for this. By definition of the localizing matrix:

$$\forall \beta,\gamma \in \mathbb{N}^n_{2r}, \quad \Big(M_{r-d_j}(q_j,y)\Big)_{\beta,\gamma} = \ell_y(g_j(x)x^\beta x^\gamma). \tag{2.30}$$

Remembering that we have already defined matrices C^j_α such that:

$$g_j(x) v_{r-d_j}(x) v_{r-d_j}(x)^\top = \sum_{\alpha\in\mathbb{N}^n_{2r}} C^j_\alpha x^\alpha, \tag{2.31}$$

Then, in particular:

$$\forall \beta,\gamma \in \mathbb{N}^n_{2r}, \quad g_j(x)x^\beta x^\gamma = \sum_{\alpha\in\mathbb{N}^n_{2r}} (C^j_\alpha)_{\beta,\gamma} x^\alpha. \tag{2.32}$$

Applying ℓ_y on both sides of (2.32), we get:

$$\forall \beta,\gamma \in \mathbb{N}^n_{2r}, \quad \Big(M_{r-d_j}(q_j,y)\Big)_{\beta,\gamma} = \ell_y(g_j(x)x^\beta x^\gamma) = \sum_{\alpha\in\mathbb{N}^n_{2r}} (C^j_\alpha)_{\beta,\gamma} y_\alpha, \tag{2.33}$$

which, in terms of matrices, is equivalent to:

$$M_{r-d_j}(q_j, y) = \sum_{\alpha \in \mathbb{N}^n_{2r}} C^j_\alpha y_\alpha. \tag{2.34}$$

The linear form appearing in the dual is then:

$$\mathcal{L}_{f_\alpha, X_j, \lambda}(y) = \sum_{\alpha \in \mathbb{N}^n_{2r}} f_\alpha y_\alpha - \lambda y_0 - \sum_{j=1}^{n_X} \sum_{\alpha \in \mathbb{N}^n_{2r}} \langle C^j_\alpha, X_j \rangle y_\alpha, \tag{2.35}$$

which is null everywhere if and only if:

$$\forall \alpha \in \mathbb{N}^n_{2r}, \quad f_\alpha - \lambda \mathbf{1}_{\alpha=0} = \sum_j \langle C^j_\alpha, X_j \rangle, \tag{2.36}$$

with the convention that $f_\alpha = 0$ for $\alpha \in \mathbb{N}^n_{2r} \backslash \mathbb{N}^n_{d_0}$. The Lagrange dual (2.29) of the moment relaxation (2.25) is then exactly the SoS tightening (2.14):

$$\begin{aligned} d^*_r = & \sup_{X_j \succeq 0,\ \lambda} \lambda \\ \text{s.t. } & \forall \alpha \in \mathbb{N}^n_{2r}, f_\alpha - \lambda \mathbf{1}_{\alpha=0} = \sum_j \langle C^j_\alpha, X_j \rangle. \end{aligned} \tag{2.37}$$

2.2.2 Application to Lyapunov Stability Assessment

Consider an ordinary differential equation:

$$\dot{x}(t) = f(x(t)), \tag{2.38}$$

with $f(0) = 0$. The origin is an equilibrium point of the ODE: as soon as x reaches it, it stays there. One can study the stability of the ODE, *i.e.*, find a region around the origin where the ODE stays close or asymptotically converges to 0. For example, a simple pendulum has two equilibrium points: the top and the bottom positions. The bottom position is stable whereas the top one is unstable. A convenient way to assess the stability of an ODE is to exhibit a Lyapunov function (Slotine and Li, 1991). J is a Lyapunov function for the ODE (2.38) over a ball $\mathcal{R}$ centered around 0, if it is a scalar function such that $J(0) = 0$, $J(x) > 0$ on $\mathcal{R} \backslash \{0\}$, and $\nabla J(x)^\top f(x) < 0$ on $\mathcal{R}$. Since $\nabla J(x)^\top f(x) = \dot{J}(x)$, this condition says that the Lyapunov function is strictly decreasing along the trajectories of the ODE. Lyapunov's second method (Lyapunov, 1992) states that if one can exhibit a Lyapunov function as above, then the system is asymptotically stable, *i.e.*, if $x(0) \in \mathcal{R}$, then $\lim_{t \to \infty} \|x(t)\| = 0$.

Example 2.2: Global asymptotic stability

Consider, with $x \in \mathbb{R}^d$, the linear ODE:

$$\dot{x}(t) = -x(t). \tag{2.39}$$

The origin is the unique equilibrium point of the ODE. Let us use $J(x) = \frac{1}{2}\|x\|^2$ as a candidate Lyapunov function. $J(0) = 0$ and J is strictly positive on $\mathbb{R}^d \backslash \{0\}$. Furthermore:

$$\dot{J}(x) = \nabla J(x)^\top f(x) = -x^\top x = -\|x\|^2 < 0, \tag{2.40}$$

for $x \neq 0$. Using Lyapunov's theorem, we have proved that the ODE is globally asymptotically stable at the origin (the stability region $\mathcal{R}$ is the whole domain).

For non-linear systems, finding a suitable Lyapunov function can be much more difficult. Even checking whether a given function is a Lyapunov function can also be hard. Suppose that f is polynomial and that we have a candidate polynomial Lyapunov function $J(x)$. We want to prove that for any x in a given sublevel set of J, we have $\dot{J}(x) < 0$. In practice $\dot{J}$ is replaced by $\dot{J}(x) + \varepsilon \leq 0$, for a small parameter ε. A sufficient condition inspired by Putinar's theorem is that:

$$-\dot{J}(x) - \varepsilon = \sigma_0(x) + \sigma_1(x)(\rho - J(x)), \tag{2.41}$$

where σ_0 and σ_1 are SoS polynomial. Indeed, if (2.41) holds, then on the sublevel set $S_\rho = \{x \mid J(x) \leq \rho\}$, we have $\dot{J}(x) + \varepsilon \leq 0$. Using the representation of SoS polynomials presented above, a finite-dimensional relaxation is:

$$-\dot{J}(x) - \varepsilon = v_d(x)^\top H_0 v_d(x) + v_d(x)^\top H_1 v_d(x)(\rho - J(x)), \quad H_0, H_1 \succeq 0. \tag{2.42}$$

The feasibility of the SDP (2.42) certifies that the ODE is asymptotically stable on S_ρ. We may as well optimize ρ to find the largest sublevel set that is a stability region:

$$\begin{aligned} &\sup\ \rho \\ \text{s.t.}\ \ &-\dot{J}(x) - \varepsilon = v_d(x)^\top H_0 v_d(x) + v_d(x)^\top H_1 v_d(x)(\rho - J(x)), \\ &H_0 \succeq 0,\ H_1 \succeq 0,\ \rho \geq 0, \end{aligned} \tag{2.43}$$

which can be solved numerically with an SDP solver. There exist several variations of this SDP which can be more computationally efficient. This approach also generalizes to trajectory tracking, with polynomials now depending on (x, t), and involving a series of SoS feasibility programs, one for each time step (Tedrake et al., 2010).

2.2.3 Polynomial Optimization for Optimal Control

2.2.3.1 Formulation with Occupation Measures

We consider the following fixed-horizon control problem:

$$\begin{aligned} \inf_u J(u) &= \int_0^T L(t,x(t),u(t))\mathrm{d}t + K(x(T)) \quad \text{s.t.} \\ \forall t \in [0,T], \quad \dot{x}(t) &= f(t,x(t),u(t)), \quad x(0) = x_0 \in \mathcal{X} \\ \forall t \in [0,T], \quad (x(t),u(t)) &\in \mathcal{X} \times \mathcal{U}, \quad x(T) \in \mathcal{X}_T. \end{aligned} \tag{2.44}$$

We suppose that $\mathcal{X}$, $\mathcal{X}_T \subset \mathbb{R}^n$ and $\mathcal{U} \subset \mathbb{R}^m$ are compact. Note that the Mayer cost is denoted in this section by K instead of M to avoid any confusion with moment matrices. Let us define the following two linear applications, $\mathcal{D} : C^1([0,T] \times \mathcal{X}) \to C([0,T] \times \mathcal{X} \times \mathcal{U})$, defined by :

$$w \mapsto \mathcal{D}(w)(t,x,u) = \frac{\partial w}{\partial t} + \nabla_x w(t,x)^\top f(t,x,u), \tag{2.45}$$

and $\mathcal{L} : w \mapsto (-\mathcal{D}w, w_T)$, where $w_T(x) = w(T,x)$.

We define the occupation measure $\mu \in \mathcal{M}_+([0,T] \times \mathcal{X} \times \mathcal{U})$, and the terminal occupation measure $\nu \in \mathcal{M}_+(\mathcal{X}_T)$ by:

$$\nu(D) = \mathbf{1}_D(x(T)), \quad \mu(A \times B \times C) = \int_A \mathbf{1}_{B \times C}(x(t),u(t))\mathrm{d}t, \tag{2.46}$$

for Borel sets $A \subset [0,T], B \subset \mathcal{X}, C \subset \mathcal{U}, D \subset \mathcal{X}_T$. $\mathcal{D}$ is a time-differentiation operator: it is such that, for any continuously differentiable test function w,

$$w_T(x(T)) - w(0,x_0) = \int_{[0,T]} \text{“ } \frac{\mathrm{d}w_t(x(t))}{\mathrm{d}t} \text{ ” } \mathrm{d}t = \int_{[0,T]} \mathcal{D}w(t,x(t),u(t))\mathrm{d}t, \tag{2.47}$$

and we have a similar relation with occupation measures:

$$\int_{\mathcal{X}_T} w_T \mathrm{d}\nu = w(0,x_0) + \int_{[0,T]\times\mathcal{X}\times\mathcal{U}} \mathcal{D}w \mathrm{d}\mu. \tag{2.48}$$

This can be written more compactly as:

$$\langle (\mu,\nu), \mathcal{L}w \rangle = \langle (\mu,\nu), (-\mathcal{D}w, w_T) \rangle = w(0,x_0) = \langle \delta_{(0,x_0)}, w \rangle, \tag{2.49}$$

or equivalently, $\mathcal{L}^*(\mu,\nu) = \delta_{(0,x_0)}$. This last equation is a reformulation of Liouville's equation, a PDE which governs the evolution of the occupation measure:

$$\frac{\partial \mu}{\partial t} + \text{div}(f\mu) = \delta_{x_0} - \nu. \tag{2.50}$$

The cost can be written with ν and μ as well:

$$J(u) = \int_{[0,T]\times\mathcal{X}\times\mathcal{U}} L \mathrm{d}\mu + \int_{\mathcal{X}_T} K \mathrm{d}\nu = \langle (\mu,\nu), (L,K) \rangle. \tag{2.51}$$

2.2.3.2 Primal and Dual Weak Formulations of Optimal Control

We can then define the primal infinite-dimensional LP verified by the occupation measures:

$$\begin{aligned} p^* = \inf_{\mu,\nu} \; & \langle (\mu,\nu), (L,K) \rangle \\ \text{s.t. } & \mathcal{L}^*(\mu,\nu) = \delta_{(0,x_0)} \\ & \mu \in \mathcal{M}_+([0,T]\times\mathcal{X}\times\mathcal{U}),\ \nu \in \mathcal{M}_+(\mathcal{X}_T). \end{aligned} \tag{2.52}$$

The Lagrange dual can be derived easily:

$$\sup_{w} \inf_{\mu\geq 0,\ \nu\geq 0} \langle (\mu,\nu),(L,K)\rangle + \langle (\delta_{(0,x_0)} - \mathcal{L}^*(\mu,\nu)), w\rangle, \tag{2.53}$$

equivalently written as:

$$\begin{aligned} d^* = \sup_{w} \; & \langle \delta_{(0,x_0)}, w\rangle \\ \text{s.t. } & \mathcal{L}w \leq (L,K) \\ & w \in C^1([0,T]\times X). \end{aligned} \tag{2.54}$$

Because $\mathcal{L}w \leq (L,K) \iff (\mathcal{D}w + L \geq 0 \text{ and } w_T \leq K)$, the constraint in (2.54) is exactly equivalent to saying that w is a smooth subsolution of the HJB equation. In the primal (2.52), we allow μ,ν to be measures instead of deterministic functions. In both cases, this is not exactly equivalent to the original control problem (2.44), but to a different control problem. This is called the *weak formulation* of optimal control, which is known to solve the relaxed control problem where the dynamics is convexified as follows:

$$\dot{x}(t) \in \text{conv}\left(\{f(t,x(t),u), u\in\mathcal{U}\}\right). \tag{2.55}$$

If for example, $f(t,x,u) = u$ and $\mathcal{U} = \{-1,1\}$, in this relaxed problem one can choose the control in the set $[-1,1]$ instead of $\{-1,1\}$. The primal, the dual and the original control problem coincide under certain convexity and compactness conditions (see Lasserre et al. (2008) for sufficient conditions). Note that the dual weak-formulation (2.54) is the continuous counterpart of the LP formulation of reinforcement learning problems presented in Section 1.2.5.

2.2.3.3 Relaxation of the Primal

We now assume that f, L and K are polynomials, and that $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{X}_T$ are compact basic semi-algebraic sets. We can then adapt the approach presented above for constrained optimization, to the weak-formulation of optimal control. First, let us write (2.52) as a moment problem. By density of the polynomials in continuous functions, we have that:

$$\begin{aligned} \mathcal{L}^*(\mu,\nu) = \delta_{(0,x_0)} &\iff \forall w, \langle (\mu,\nu), \mathcal{L}w\rangle = \langle \delta_{(0,x_0)}, w\rangle \\ &\iff \forall (\alpha,\beta)\in\mathbb{N}^n\times\mathbb{N}, \langle (\mu,\nu), \mathcal{L}(x^\alpha t^\beta)\rangle = \langle \delta_{(0,x_0)}, x^\alpha t^\beta\rangle \\ &\iff \forall(\alpha,\beta), -\int \left(\frac{\partial (x^\alpha t^\beta)}{\partial t} + \nabla_x (x^\alpha t^\beta)^\top f\right) \mathrm{d}\mu + \int x^\alpha T^\beta \mathrm{d}\nu = x_0^\alpha \mathbf{1}_{\beta=0} \\ &\iff \forall(\alpha,\beta), -\ell_z(\beta x^\alpha t^{\beta-1} + t^\beta (\nabla x^\alpha)^\top f(t,x,u)) + T^\beta y_\alpha = x_0^\alpha \mathbf{1}_{\beta=0}, \end{aligned} \tag{2.56}$$

where the (y_α) and $(z_{\alpha,\beta})$ are the respective moments of ν and μ.

Then (2.52) is equivalent to the following LP with countably many linear constraints (Lasserre, 2015):

$$\begin{aligned} p^* = \inf_{\mu,\nu} \ & \langle (\mu,\nu), (L,K) \rangle \\ \text{s.t. } & \forall (\alpha,\beta) \in \mathbb{N}^n \times \mathbb{N}, \langle (\mu,\nu), \mathcal{L}(x^\alpha t^\beta) \rangle = x_0^\alpha \mathbf{1}_{\beta=0} \\ & \mu \in \mathcal{M}_+([0,T] \times \mathcal{X} \times \mathcal{U}), \ \nu \in \mathcal{M}_+(\mathcal{X}_T). \end{aligned} \tag{2.57}$$

Assume that $\mathcal{X}_T$ and $[0,T] \times \mathcal{X} \times \mathcal{U}$ are respectively described by polynomial inequalities $g_j(x) \geq 0$ and $h_k(t,x,u) \geq 0$, with $d_j = \lceil \deg(g_j)/2 \rceil$, $r_k = \lceil \deg(h_k)/2 \rceil$ ($j \in \{1,...,n_f\}$, $k \in \{1,...,n_t\}$). For $2d \geq \max\{\deg(L), \deg(K), \max(d_j), \max(r_k)\}$, the rank d moment relaxation of (2.57) is:

$$\begin{aligned} p_d^* = \inf_{y,z} \ & \ell_z(L) + \ell_y(K) \\ \text{s.t. } & M_d(y) \succeq 0, \quad M_d(z) \succeq 0, \quad M_{d-d_j}(y, g_j) \succeq 0, \forall j, \quad M_{d-r_k}(z, h_k) \succeq 0, \forall k, \\ & \forall (\alpha,\beta) \in \mathbb{N}^n \times \mathbb{N}, |\alpha|, |\beta| \leq 2d, -\ell_z(\beta x^\alpha t^{\beta-1} + t^\beta (\nabla x^\alpha)^\top f) + T^\beta y_\alpha = x_0^\alpha \mathbf{1}_{\beta=0}. \end{aligned} \tag{2.58}$$

2.2.3.4 Dual of the Relaxation

In this final section, we prove that the Lagrange dual of the relaxation (2.58) of the primal problem is also a natural relaxation of the dual problem (2.54). The computations are a bit tedious and can be skipped without harm, the main take-away message being summarized in Figure 2.3.

Let us derive the Lagrange dual of (2.58). We assume that $L(t,x,u) = \sum_{\alpha,\beta,\gamma} L_{\alpha,\beta,\gamma} x^\alpha t^\beta u^\gamma$, and $K(x) = \sum_\alpha K_\alpha x^\alpha$. Let $g_0 = h_0 = 1$. The dual is:

$$\begin{aligned} \sup_{\substack{\{Y_j\},\{Z_k\} \succeq 0, \\ \{\lambda_{\alpha,\beta}\}}} \inf_{y,z} \ & \ell_z(L) + \ell_y(K) - \sum_{j=0}^{n_f} \langle M_{d-d_j}(y, g_j), Y_j \rangle - \sum_{k=0}^{n_t} \langle M_{d-r_r}(z, h_k), Z_k \rangle \\ & + \sum_{\alpha,\beta} \lambda_{\alpha,\beta} (x_0^\alpha \mathbf{1}_{\beta=0} + \ell_z(\beta x^\alpha t^{\beta-1} + t^\beta (\nabla x^\alpha)^\top f) - T^\beta y_\alpha), \end{aligned} \tag{2.59}$$

or, equivalently:

$$\begin{aligned} \sup_{\substack{\{Y_j\},\{Z_k\} \succeq 0, \\ \{\lambda_{\alpha,\beta}\}}} & \sum_{\alpha,\beta} \lambda_{\alpha,\beta} x_0^\alpha \mathbf{1}_{\beta=0} \ \text{ s.t. } \ \forall y, \ell_y(K) - \sum_j \langle M_{d-d_j}(y, g_j), Y_j \rangle - \sum_{\alpha,\beta} \lambda_{\alpha,\beta} T^\beta y_\alpha = 0 \\ & \forall z, \ell_z(L) - \sum_k \langle M_{d-r_k}(z, h_k), Z_k \rangle + \sum_{\alpha,\beta} \lambda_{\alpha,\beta} \ell_z(\beta x^\alpha t^{\beta-1} + t^\beta (\nabla x^\alpha)^\top f) = 0. \end{aligned} \tag{2.60}$$

Following Henrion (2014), we define matrices B_α^j, with the following relation with $M_{d-d_j}(g_j, y)$:

$$\begin{cases} g_j(x) v_{d-d_j}(x) v_{d-d_j}(x)^\top = \sum_\alpha B_\alpha^j x^\alpha \\ M_{d-d_j}(g_j, y) = \sum_\alpha B_\alpha^j y_\alpha, \end{cases} \tag{2.61}$$

and similarly:

$$\begin{cases} h_k(t,x,u) v_{d-r_k}(t,x,u) v_{d-r_k}(t,x,u)^\top = \sum_{\alpha,\beta,\gamma} C_{\alpha,\beta,\gamma}^k x^\alpha t^\beta u^\gamma \\ M_{d-r_k}(h_k, z) = \sum_{\alpha,\beta,\gamma} C_{\alpha,\beta,\gamma}^k z_{\alpha,\beta,\gamma}. \end{cases} \tag{2.62}$$

Expressing the constraints in (2.60) with the matrices B_α^j and $C_{\alpha,\beta,\gamma}^k$, we obtain:

$$\sup_{\substack{\{Y_j\},\{Z_k\}\succeq 0,\\ \{\lambda_{\alpha,\beta}\}}} \sum_\alpha \lambda_{\alpha,0} x_0^\alpha \quad \text{s.t.} \quad \forall \alpha, K_\alpha - \sum_\beta \lambda_{\alpha,\beta} T^\beta = \sum_j \langle B_\alpha^j, Y_j\rangle \tag{2.63}$$
$$\forall \alpha,\beta,\gamma, L_{\alpha,\beta,\gamma} + \sum_{a,b} \lambda_{a,b} (\mathcal{D}(x^a t^b))_{\alpha,\beta,\gamma} = \sum_k \langle C_{\alpha,\beta,\gamma}^k, Z_k\rangle.$$

Let $w(t,x) = \sum_{\alpha,\beta} \lambda_{\alpha,\beta} t^\beta x^\alpha$. Multiplying the constraints by x^α (resp. by $x^\alpha t^\beta u^\gamma$) and summing on α (resp. on α,β,γ), we obtain:

$$\sup_{\substack{\{Y_j\},\{Z_k\}\succeq 0,\\ w(t,x)}} w(0,x_0) \quad \text{s.t.} \quad \forall x, K(x) - w(T,x) = \sum_j \langle \sum_\alpha B_\alpha^j x^\alpha, Y_j\rangle \tag{2.64}$$
$$\forall (t,x,u), L(t,x,u) + \mathcal{D}(w)(t,x,u) = \sum_k \langle \sum_{\alpha,\beta,\gamma} C_{\alpha,\beta,\gamma}^k t^\beta u^\gamma x^\alpha, Z_k\rangle.$$

Finally, using the definition of B_α^j and $C_{\alpha,\beta,\gamma}^k$, we get:

$$\sup_{\substack{\{Y_j\},\{Z_k\}\succeq 0,\\ w(t,x)}} w(0,x_0) \quad \text{s.t.} \quad \forall x, K(x) - w(T,x) = \sum_j g_j(x) \mathrm{Tr}(v_{d-d_j}(x) v_{d-d_j}(x)^\top Y_j) \tag{2.65}$$
$$\forall (t,x,u), L(t,x,u) + \mathcal{D}(w)(t,x,u) = \sum_k h_k(t,x,u) \mathrm{Tr}(v_{d-r_k}(t,x,u) v_{d-r_k}(t,x,u)^\top Z_k).$$

We recognize that the terms inside the traces are SoS polynomials $\sigma_j(x)$ and $\psi_k(t,x,u)$, hence the dual of the relaxation (2.58) is:

$$\begin{aligned} d_d^* = \sup_{\substack{\{\sigma_j\},\{\psi_k\} \text{ SoS},\\ w(t,x)}} & w(0,x_0) \\ \text{s.t. } K - w_T &= \sigma_0 + \sum_{j=1}^{n_f} \sigma_j g_j \\ L + \mathcal{D}w &= \psi_0 + \sum_{k=1}^{n_t} \psi_k h_k. \end{aligned} \tag{2.66}$$

This problem (2.66) is a SoS tightening of the dual (2.54), whose constraints are:

$$\begin{cases} -\mathcal{D}w \leq L \\ w_T \leq K \end{cases} \iff \begin{cases} L + \mathcal{D}w \geq 0 \\ K - w_T \geq 0 \end{cases} \tag{2.67}$$

More precisely, (2.66) tightens (2.54), by replacing non-negative polynomials by SoS polynomials of degree less than r. To summarize, the diagram of Figure 2.3 commutes.

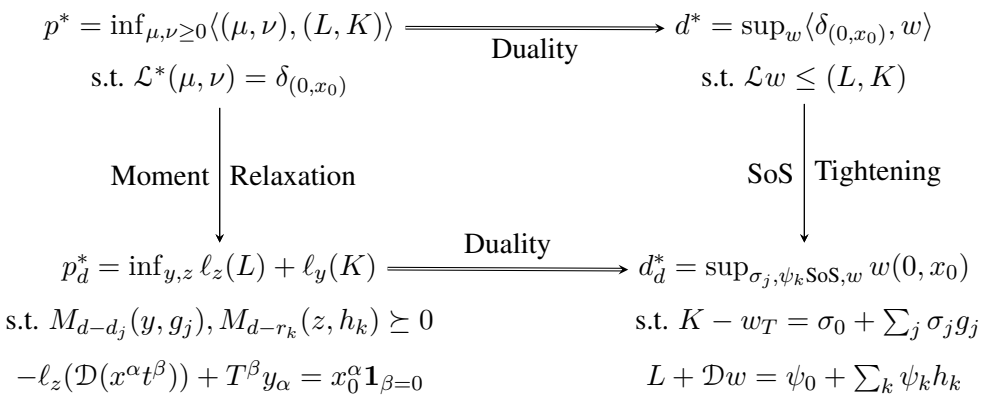


Figure 2.3: Relations between the primal and dual weak formulations of control, and the Moment-SoS hierarchy.

2.3 Kernel Methods

Kernel methods are a class of algorithms for machine learning (Mairal and Vert, 2018; Schölkopf and Smola, 2001; Shawe-Taylor and Cristianini, 2004). They exploit similarities between data points to perform a wide range of tasks, such as classification, clustering or principal component analysis. Kernel methods are inherently *sample-based*, or *data-driven*, which makes them particularly suitable for reinforcement learning or model-free control applications. We will use kernel methods to design a new sample-based algorithm for optimal control in Chapter 5, and to analyze an existing model-free RL algorithm in Chapter 6.

2.3.1 Representing Functions

Let X an arbitrary set. We consider the problem of representing a function $f : X \to \mathbb{R}$ so that it can be stored in memory and used in any computational method. One can think of representing the value function of a control problem, but the question is much more general. If X is a finite set, we can simply store all of its values. But as we have seen in Section 1.2, $|X|$ can be so large that any exhaustive storage is intractable. Furthermore, X can be a dense set like $[0, 1]$, also preventing storage. A simple and fairly generic solution is to represent f as a linear combination in a basis of functions $(\varphi_1, \dots \varphi_k)$:

$$f \simeq f_\theta = \sum_{j=1}^{k} \theta_j \varphi_j, \tag{2.68}$$

where $\theta \in \mathbb{R}^k$ is a vector *representing* the function f in the basis. Of course, this representation is not exact in general. If $X = [0, 1]$ and the φ_j are a finite basis of monomials and f is a continuous function (*e.g.*, the sine function), then f does not belong to the span of the φ_j, but the Weierstrass approximation theorem ensures that f can be uniformly approximated by polynomials. Many other bases can be considered, like sine and cosine functions forming the Fourier basis, or Chebyshev polynomials (Cheney and Light, 2009). Note that in Section 2.4, we present a *max-plus* variant of such linear combinations, that we will use later to approximate a value function in Chapter 3.

Let $\varphi = (\varphi_1, \dots \varphi_k)^\top$, then an equivalent expression of f_θ in (2.68) is:

$$\forall x \in X, \quad f_\theta(x) = \theta^\top \varphi(x). \tag{2.69}$$

Kernel methods extend such representations to any Hilbert space (a complete vector space of functions endowed with an inner product $\langle \cdot, \cdot \rangle$):

$$\forall x \in X, \quad f_\theta(x) = \langle \theta, \varphi(x) \rangle, \tag{2.70}$$

where θ and $\varphi(x)$ are elements of the Hilbert space.

2.3.2 Reproducing Kernel Hilbert Spaces

A positive-definite kernel is a symmetric function $K : X \times X \to \mathbb{R}$ such that for any integer $n \geq 1$, and for any $x_1, \dots, x_n \in X$, the kernel (or Gram) matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ with entries $\mathbf{K}_{i,j} = K(x_i, x_j)$ is positive semi-definite (PSD). The kernel function K must be thought of as a way to measure dissimilarity between two elements of X. We will give concrete examples below.

Consider, for $x \in X$, the function:

$$\begin{aligned} K_x : X &\to \mathbb{R} \\ y &\mapsto K(x, y). \end{aligned} \tag{2.71}$$

We are going to use such functions as building blocks of a Hilbert space of functions. Let:

$$\mathcal{H}_0 = \text{span}\{K_x, \text{for } x \in X\}. \tag{2.72}$$

We can define an inner product on $\mathcal{H}_0$ by the bilinear form $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that:

$$\langle K_x, K_y \rangle_{\mathcal{H}} = K(x, y). \tag{2.73}$$

The fact that K is a positive-definite kernel ensures that this has all the properties of an inner product. Indeed, for any $a \in \mathbb{R}^n$ and $x_1, \dots, x_n \in X$, let $\mathbf{K}$ the $n \times n$ PSD kernel matrix, then:

$$\begin{aligned} \left\langle \sum_{j=1}^n a_j K_{x_j}, \sum_{k=1}^n a_k K_{x_k} \right\rangle_{\mathcal{H}} &= \sum_{j=1}^n \sum_{k=1}^n a_j a_k \langle K_{x_j}, K_{x_k} \rangle_{\mathcal{H}} \\ &= a^\top \mathbf{K} a \geq 0. \end{aligned} \tag{2.74}$$

Yet, the space of functions $\mathcal{H}_0$ is not a Hilbert space because it is not complete with respect to the metric induced by the inner product. Instead, we consider its closure, *i.e.*, the Hilbert space:

$$\mathcal{H} = \overline{\text{span}\{K_x, \text{for } x \in X\}}. \tag{2.75}$$

The property (2.73) extends to $\mathcal{H}$ by density. Informally, let $f \in \mathcal{H}$, it is the limit of some sequence $(f_n)_{n \geq 0}$ of elements of $\mathcal{H}_0$, which have the reproducing property (by linearity). Then for any $x \in X$:

$$\begin{aligned} \langle K_x, f_n \rangle_{\mathcal{H}} &= f_n(x) \\ \downarrow \quad & n \to \infty \downarrow \\ \langle K_x, f \rangle_{\mathcal{H}} &= f(x). \end{aligned} \tag{2.76}$$

Note that we have not formally justified the validity of the limits, which would involve using Cauchy sequences (see a complete proof in Mairal and Vert (2018)). The latter property (2.76) is called the *reproducing property* and is central to kernel methods. A formal justification is provided by the following theorem by Aronszajn (1950).

Theorem 4 (Moore-Aronszajn)**.** *For any positive-definite kernel K, there exists a unique reproducing kernel Hilbert space (RKHS) $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ with reproducing kernel K. It is a Hilbert space of real-valued functions on X endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, such that:*

- *for any $x \in X$, the function K_x belongs to $\mathcal{H}$;*
- *for any $x \in X$ and $f \in \mathcal{H}$, the reproducing property holds:*

$$\langle K_x, f \rangle_{\mathcal{H}} = f(x). \tag{2.77}$$

The reproducing property can be seen as an extension of the linear parameterization of a function (2.69). The expression $f_\theta(x) = \theta^\top \varphi(x)$ is replaced by:

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}}, \tag{2.78}$$

for $f \in \mathcal{H}$, with $\phi(x) = K_x \in \mathcal{H}$, called the embedding or feature map of x in $\mathcal{H}$. In this expression, the representation of f is *non-parametric*, in the sense that it is not represented by a vector $\theta \in \mathbb{R}^d$, but directly by being an element of $\mathcal{H}$, a possibly infinite-dimensional Hilbert space. This could apparently look like a major drawback to perform practical computations, but as we will see later (see *e.g.*, Sections 2.3.3, 2.3.4 and 2.3.5, and also Chapters 5 and 6), this representation in $\mathcal{H}$ is never considered explicitly, and many computations only involve evaluations of the kernel function. This phenomenon is generically called a "kernel trick".

Parametric representations of functions are a particular case of RKHS. Consider the basis of functions $(\varphi_1, \ldots, \varphi_k)$ in (2.68). Let $K(x, y) = \varphi(x)^\top \varphi(y)$. It is called the linear kernel. Identifying the vectors of $\mathbb{R}^k$ and the linear forms on $\mathbb{R}^k$, the RKHS $\mathcal{H} = \text{span}(\varphi_1, \ldots, \varphi_k)$ is a subset of $\mathbb{R}^k$, associated with the canonical dot product on $\mathbb{R}^k$. A function $f \in \mathcal{H}$ is associated to a unique vector $\theta \in \mathbb{R}^k$, and the reproducing property writes:

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}} = \theta^\top \varphi(x). \tag{2.79}$$

Example of kernels. The RKHS framework is much wider than finite-dimensional, parametric spaces of functions. There are many examples of kernels which lead to different Hilbert spaces. We list just a few here, and refer to Fasshauer (2011) for other examples. On the underlying space $X = \mathbb{R}^d$, we can consider:

- The polynomial kernel $K(x, y) = (1 + x^\top y)^p$, for $p \geq 1$. The associated RKHS is a finite-dimensional Hilbert space, whose embedding $\phi(x)$ corresponds to the monomials of degree less than p (up to some scaling factors due to binomials coefficients), and has dimension C_{d+p}^{d} (see Section 2.2). $p = 2$ is commonly used, higher degree polynomial kernels being impractical because of numerical instabilities.
- The Gaussian kernel $K(x, y) = \exp(-\|x - y\|^2 / \sigma^2)$, which leads to an infinite-dimensional Hilbert space of very smooth C^∞ functions. The parameter σ is the bandwidth of the Gaussian filter, also called the radial basis function.

- The Laplace or exponential kernel $K(x,y) = \exp(-\|x-y\|/\sigma)$, for $\sigma > 0$, is associated to a Hilbert space of less smooth functions, *i.e.*, the Sobolev space of functions with d derivatives.

The latter two kernels are translation-invariant kernels of the form $K(x,y) = \tilde{K}(x-y)$. The regularity of the functions in the associated RKHS can be related to the Fourier coefficients of $\tilde{K}$. In particular, other Sobolev spaces can be obtained using different translation-invariant kernels (see Wahba (1990) and the numerical examples in Chapter 6).

Finally, let us mention that, apart from the positive-definiteness of the kernel, nothing is assumed about the underlying space X. It must not necessarily be a Euclidean space, as one can compute positive-definite kernels on many objects, such as graphs (Borgwardt and Kriegel, 2005; Ralaivola et al., 2005), text sequences (Lodhi et al., 2002), biological sequences (Schölkopf et al., 2004), probabilistic models (Jaakkola and Haussler, 1998), any many others. The RKHS framework can be extended to handle vector-valued or operator-valued kernels, with interesting connections to dynamical systems (Aubin-Frankowski, 2021b; Heinonen and d'Alché Buc, 2014).

2.3.3 Kernel Methods for Supervised Learning

We consider the non-parametric least-squares regression problem. Let $(x_i, y_i)_{1\leq i\leq n} \in (X \times \mathbb{R})^n$ a collection of data points. We want to find a regression function f in a given RKHS, such that for all i, $f(x_i) \simeq y_i$. More formally, we look for a function which minimizes the following regularized mean squared error, for some $\lambda > 0$:

$$\min_{f\in\mathcal{H}} \frac{1}{2n}\sum_{i=1}^{n} (f(x_i) - y_i)^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \,. \tag{2.80}$$

This problem is usually called *kernel ridge regression*. The objective is composed of a data fitting term, and a ridge (also called Tikhonov) regularization term which controls the regularity of f. Indeed, for $x, x' \in X$, we have:

$$\begin{aligned} |f(x) - f(x')| &= |\langle f, K_x - K_{x'}\rangle_{\mathcal{H}}| \\ &\leq \|f\|_{\mathcal{H}}\|\phi(x) - \phi(x')\|_{\mathcal{H}} \,. \end{aligned} \tag{2.81}$$

In other words, the $\mathcal{H}$-norm of f controls the Lipschitz constant of f with respect to the pseudometric defined on X by $d_K(x,x') = \|\phi(x) - \phi(x')\|_{\mathcal{H}}$. Apart from this concrete effect, the quadratic regularization term is crucial from a theoretical viewpoint: it allows to apply the following representer theorem by Schölkopf et al. (2001). Note that the theorem is slightly more general but we adapt it here to problem (2.80) and succinctly remind its proof.

Theorem 5 (Representer theorem)**.** *Any minimizer f^* of the optimization problem (2.80) admits a representation of the form $f^*(\cdot) = \sum_{i=1}^n \alpha_i K(x_i, \cdot)$, for some $\alpha \in \mathbb{R}^n$.*

Proof. Using the orthogonal projection on $\mathrm{span}(\phi(x_1), \dots, \phi(x_n))$, any $f \in \mathcal{H}$ can be decomposed as:

$$f = \sum_{i=1}^{n} \alpha_i \phi(x_i) + f_\perp, \tag{2.82}$$

with, for any i, $\langle f_\perp, \phi(x_i)\rangle_{\mathcal{H}} = 0$. Then, for any $i \in \{1, \dots, n\}$:

$$\begin{aligned} f(x_i) &= \left\langle \sum_{j=1}^{n} \alpha_j \phi(x_j) + f_\perp, \phi(x_i) \right\rangle_{\mathcal{H}} \\ &= \sum_{j=1}^{n} \alpha_j \langle \phi(x_j), \phi(x_i)\rangle_{\mathcal{H}}, \end{aligned} \tag{2.83}$$

which is independent of $f_\perp$, hence the data-fitting term does not depend on $f_\perp$ either. Then $f_\perp$ only affects the regularization term. Assume that $f_\perp \neq 0$. By definition of the orthogonal projection, and using that $\lambda > 0$:

$$\begin{aligned} \frac{\lambda}{2} \| \sum_{i=1}^{n} \alpha_i \phi(x_i) + f_\perp \|_{\mathcal{H}}^2 &= \frac{\lambda}{2} \| \sum_{i=1}^{n} \alpha_i \phi(x_i) \|_{\mathcal{H}}^2 + \frac{\lambda}{2} \| f_\perp \|_{\mathcal{H}}^2 \\ &> \frac{\lambda}{2} \| \sum_{i=1}^{n} \alpha_i \phi(x_i) \|_{\mathcal{H}}^2 . \end{aligned} \tag{2.84}$$

Overall, choosing $f_\perp \neq 0$ strictly increases the objective of (2.80), theorefore any minimizer of (2.80) must lie on $\text{span}(\phi(x_1), \dots, \phi(x_n))$. □

The representer theorem ensures that the kernel ridge regression problem (2.80) can be equivalently formulated as the following finite-dimensional optimization problem:

$$\inf_{\alpha \in \mathbb{R}^n} \frac{1}{2n} \| \mathbf{K}\alpha - y \|_{\mathcal{H}}^2 + \frac{\lambda}{2} \alpha^\top \mathbf{K} \alpha \, . \tag{2.85}$$

Using the first-order optimality condition, it follows that the unique solution is $f_\lambda = \sum_{i=1}^{n} \alpha_i^\lambda \phi(x_i)$, with $\alpha^\lambda = (\mathbf{K} + n\lambda I_n)^{-1} y$. The solution is computed from the kernel matrix only: this is a kernel trick.

Example 2.3: Solving non-parametric least-squares in closed-form

We perform kernel ridge regression on a simple one-dimensional example. The data is generated as follows. Let $n = 10$, and for $i \in \{1, \dots, n\}$, $x_i = i/n$ and $y_i = 3x_i^2 - 2x_i + 0.2 z_i$, where the z_i are independent identically distributed (*i.i.d.*) random variables such that $z_i \sim \mathcal{N}(0, 1)$. This can be seen as a noisy interpolation of the function $f^*(x) = 3x^2 - 2x$. We use the Gaussian kernel with $\sigma = 0.5$, and study the effect of the regularization parameter λ.

The obtained regression functions f_λ are shown in Figure 2.4 for different values of λ. If it is chosen too small, f_λ is not regular enough: it fits the noise in the data, *i.e.*, it *overfits*. If λ is too small, the variations of f_λ are almost zero: it *underfits* the data. Both are likely to perform poorly on unseen data points. Choosing the right λ can be addressed from a practical (using cross-validation) or theoretical perspective. Theoretical analyses of convergence rates often prescribe choices of λ to balance bias and variance terms, as we will also do in Chapter 6.

The complexity of kernel ridge regression boils down to the computation of the kernel matrix $\mathbf{K}$, *i.e.*, a time and space complexity of $O(n^2)$, and then a system inversion (with complexity $O(n^3)$). Low-rank matrix approximation methods, such as Nyström approximation (Williams and Seeger, 2000) or random features (Yang et al., 2012) can be applied to reduce the computational and memory loads. They typically require to compute and store a matrix of size $n \times q$, where q is the rank of the approximate kernel matrix, without having to first compute the whole $n \times n$ kernel matrix.

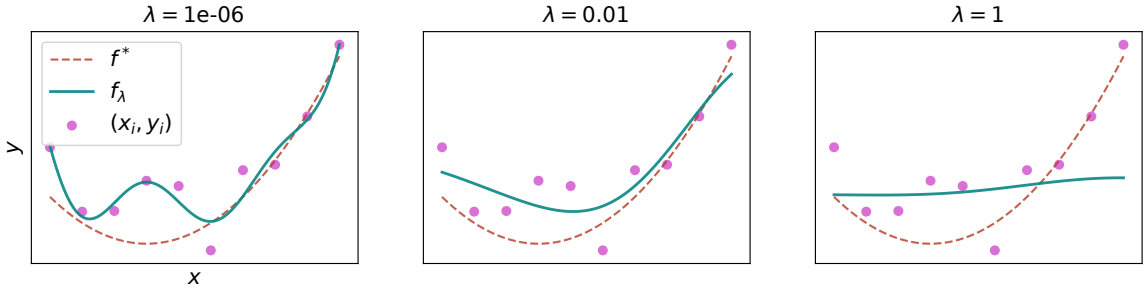


Figure 2.4: Solutions of the kernel ridge regression problem (2.80) for different values of λ.

2.3.4 Non-parametric Stochastic Gradient Descent

If the number of sample points n is too large, the computation of the closed form solution α^λ is intractable. In this case, as is common in machine learning, we can opt for incremental methods which handle one data point at a time (Bottou and Bousquet, 2007). Let us write again problem (2.80):

$$\min_{f\in\mathcal{H}} \frac{1}{2n}\sum_{i=1}^{n}\Big(\langle f,\phi(x_i)\rangle_{\mathcal{H}} - y_i\Big)^2 + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2 \,. \tag{2.86}$$

Since the objective function of (2.86) appears as a finite sum over the samples, we can apply stochastic gradient descent (SGD). The subtlety here is that since the variable is $f \in \mathcal{H}$, SGD must be carried in the – possibly infinite-dimensional – RKHS. At step k, we randomly sample a data point $(x_{i(k)}, y_{i(k)})$. Let $f_0 = 0$, having computed the stochastic gradients, the iterates write, for $k \geq 1$ and some step size $\rho_k > 0$:

$$f_k = (1-\lambda\rho_k)f_{k-1} + \rho_k\Big(y_{i(k)} - f(x_{i(k)})\Big)\phi(x_{i(k)}) \,. \tag{2.87}$$

Looking at the iterates (2.87), we can see that $f_k \in \operatorname{span}(\phi(x_{i(1)}),\dots\phi(x_{i(k)}))$. This is a very simple form of representer theorem. From that, we can obtain a closed-form, finite-dimensional expression of the iterations, using an n-dimensional vector α_n for the n-th iterate $f_n = \sum_{k=1}^n \alpha_{n,k}\phi(x_{i(k)})$. The expression of the recursion building the triangular matrix α will be detailed in Chapter 6, for the TD-learning algorithm which is a generalization of SGD (simply set $\gamma = 0$ to recover SGD).

Coming back to Example 2.3, we can apply SGD to this least-squares regression problem. In this experiment, a fresh random sample (x_k, y_k) is drawn at each iteration (single-pass SGD). Different iterates of the recursion f_n are displayed on Figure 2.5, using $\lambda = 10^{-2}$ and a sequence of decreasing step sizes $\rho_k = 1/k$. Note that this choice of step size fulfills the Robbins-Monro conditions which are classical in stochastic approximation theory (Borkar, 2009). One could also use Polyak-Ruppert averaging and output $\bar{f}_n = \sum_{k=0}^{n-1} f_k/n$ instead of f_n, which theoretically improves the convergence of SGD,

Random design and covariance operators. We have described above the fixed design setting, where the x_i are deterministic. In Chapter 6, we will rather carry out a random design analysis, *i.e.*, we suppose that the successive samples x_i are drawn according to some probability distribution p, and that $y_i = f^*(x_i) + \varepsilon_i$ (where the ε_i are *i.i.d.* scalar white noise random variables, independent of the x_i). Following Borkar and Meyn (2000), we can use the ODE method to study non-parametric SGD (and non-parametric temporal-

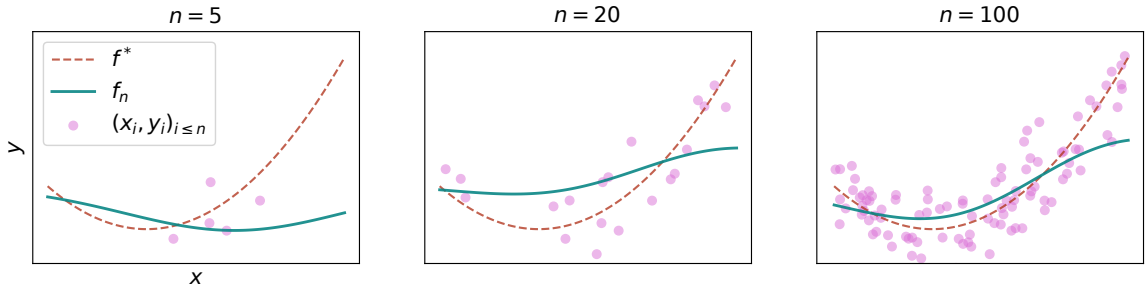


Figure 2.5: Iterates f_n of SGD on problem (2.86), using the Gaussian kernel.

difference in Chapter 6). Equation (2.87) is an instance of stochastic approximation (SA), yet an infinite-dimensional one. Borkar and Meyn (2000) suggest to study an SA iteration:

$$Z_n = Z_{n-1} + a_n[h(Z_{n-1}) + w_n], \tag{2.88}$$

where $a_n > 0$ and the w_n are uncorrelated with noise terms, by asserting the stability of an averaged continuous-time version of (2.88), *i.e.*, of the ODE:

$$\dot{z}(t) = h(z(t)). \tag{2.89}$$

For non-parametric SGD (2.87), this corresponds to the following ODE in $\mathcal{H}$:

$$\frac{\mathrm{d}f_t}{\mathrm{d}t} = -(\Sigma + \lambda I)f_t + \Sigma f^*, \tag{2.90}$$

where Σ is the covariance operator:

$$\Sigma = \int_X \phi(x) \otimes \phi(x) \mathrm{d}p(x), \tag{2.91}$$

where $\otimes$ denotes the outer product in $\mathcal{H}$ defined by $g \otimes h : f \mapsto \langle f, h \rangle_{\mathcal{H}} g$. The covariance operator Σ is an operator from $\mathcal{H}$ to $\mathcal{H}$, which plays a central role at the interface between kernel methods and probabilities (see in particular Bach (2022)).

The main properties of the covariance operator have been studied by Dieuleveut and Bach (2016), highlighting the link between the spectrum of Σ and the convergence rate of SGD (see also the related analyses by Berthier et al. (2020); Pillaud-Vivien et al. (2018b)). Importantly, contrary to the original ODE method which only proves the asymptotic convergence of SA, such analyses are non-asymptotic and provide finite-sample convergence rates. Since they deal with non-parametric SGD, the rates are *dimensionless* and provide insights on high-dimensional phenomena, *e.g.* linear regression for $d \gg n$. More generally, kernel methods replace the intrinsic dimension of the data d by the number of data points n, not only in terms of rates but also in terms of computational complexity. In Chapter 6, we will extend such non-asymptotic analyses of SGD to non-parametric TD-learning, relying on a second covariance operator.

2.3.5 Representing Non-negative Functions

As we have seen in Section 2.2, it can be useful to represent non-negative functions, be it for global optimization (see Section 2.2.1) or for optimal control (our main focus in this thesis, see Section 2.2.3). We have

introduced the sum-of-squares (SoS) representation of non-negative polynomials. In particular, assume that $X = \mathbb{R}^d$, and let $\phi_p(x)$ the vector of monomials of degree less than p. Then, a SoS polynomial of degree $2p$ writes:

$$\sigma(x) = \phi_p(x)^\top A \phi_p(x), \tag{2.92}$$

where A is a PSD matrix of appropriate size. This is a parametric representation of the function σ, the parameter being a matrix in the cone of PSD matrices, instead of a vector θ as we have seen in Section 2.3.1 to model a function without the non-negativity constraint. It turns out that this parametric PSD representation can be extended to a non-parametric representation of non-negative functions in RKHS. It has been recently proposed by Marteau-Ferey et al. (2020) to model a non-negative function as an infinite-dimensional SoS:

$$f_{\mathcal{A}}(x) = \langle \phi(x), \mathcal{A}\phi(x) \rangle_{\mathcal{H}}, \tag{2.93}$$

where $\mathcal{A} \in S_+(\mathcal{H})$ is a PSD operator, and $\phi(x) \in \mathcal{H}$ is the feature map of x in the RKHS. Clearly, f is a non-negative function and it is a sum-of-squares of functions on $\mathcal{H}$. This can be seen from an eigenvalue decomposition of $\mathcal{A}$: assume that it can be decomposed as $\mathcal{A} = \sum_{i \geq 1} \lambda_i u_i \otimes u_i$, with $\lambda_i \geq 0$, then:

$$f_{\mathcal{A}}(x) = \sum_{i \geq 1} \lambda_i \langle \phi(x), u_i \otimes u_i \phi(x) \rangle_{\mathcal{H}} = \sum_{i \geq 1} \left(\sqrt{\lambda_i} u_i(x) \right)^2. \tag{2.94}$$

This non-parametric representation contains in particular the parametric representation of SoS polynomials (2.92), which can be recovered (up to multiplicative constants) by choosing K as the polynomial kernel of degree p, *i.e.*, $K(x, y) = (1 + x^\top y)^p$. The representation (2.93) is not the only possibility to model a non-negative function, but it has several desirable properties for machine learning applications: linearity with respect to the parameter, universal approximation, finite-dimensional representation (with a representer theorem) and differentiability (Marteau-Ferey et al., 2020).

Similarly to polynomial SoS (see Section 2.2), the non-parametric SoS models (or PSD models, or "kernel SoS") have been applied to global optimization, leading to a practical algorithm with nearly optimal rates for optimizing very smooth functions in Sobolev spaces (Rudi et al., 2020). In Chapter 5, we propose an application to optimal control, hence extending the polynomial method presented in Section 2.2.3. An important result by Rudi et al. (2020) and in Chapter 5 is to derive the equivalent of a Positivstellensatz: when can a non-negative function be represented as a SoS of the form (2.93)? In optimization, this is true for smooth non-negative functions which touch zero at isolated points, in optimal control, we prove a similar result for the Hamiltonian under some assumptions.

Among other applications (to optimal transport (Vacher et al., 2021), probability modeling (Rudi and Ciliberto, 2021),...), we can use a SoS model to do non-parametric regression, where the regression function is constrained to be non-negative. Indeed, we can solve the regression problem:

$$\min_{\mathcal{A} \in \mathcal{S}_+(\mathcal{H})} \frac{1}{2n} \sum_{i=1}^{n} \left(\langle \phi(x_i), \mathcal{A}\phi(x_i) \rangle_{\mathcal{H}} - y_i \right)^2 + \lambda \Omega(\mathcal{A}), \tag{2.95}$$

where Ω is a regularization, *e.g.*, $\Omega(A) = \|\mathcal{A}\|_\star + 0.01\|\mathcal{A}\|_F^2/2$, a combination of the nuclear and Frobenius norms. For such a problem, Marteau-Ferey et al. (2020) provide a representer theorem, which allows to replace the infinite-dimensional decision variable (the operator $\mathcal{A}$) by an $n \times n$ matrix. The problem then becomes an SDP which can be solved with a standard SDP solver. In Figure 2.6, we show the obtained non-negative functions $f_{\mathcal{A}}$ for different levels of regularization, on the data of Example 2.3. A more interesting application of non-negative regression is density estimation.

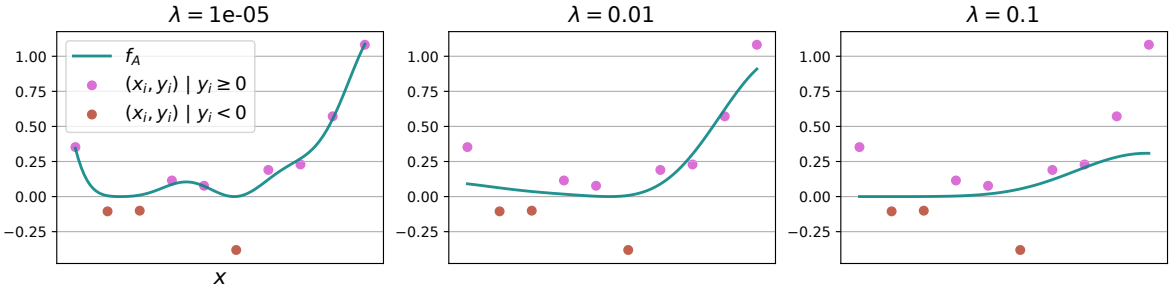


Figure 2.6: Non-negative regression functions obtained from solving problem (2.95) for different λ.

2.4 Max-Plus Algebra

In this section, we present basic notions on max-plus (also called *tropical*) algebra, along with a link with optimal control. This section is largely based on the introduction by Gaubert and Plus (1997). These notions will be used in Chapter 3.

2.4.1 The Max-Plus Semiring

The max-plus semiring $(\mathbb{R}_{\max}, \oplus, \otimes)$ is the set $\mathbb{R} \cup \{-\infty\}$, equipped with the two operations:

$$\begin{cases} x \oplus y = \max\{x, y\} \\ x \otimes y = x + y. \end{cases} \tag{2.96}$$

The relations $\oplus$ and $\otimes$ are associative and commutative. The $\mathbb{0}$ element for $\oplus$ is $-\infty$, which is such that:

$$x \otimes -\infty = \max\{x, -\infty\} = x\ . \tag{2.97}$$

The $\mathbb{1}$ element for $\otimes$ is 0, such that $x \otimes 0 = x + 0 = x$. All non-zero elements (*i.e.*, different from $-\infty$) have an inverse for $\otimes$, equal to $-x$ (hence making the structure a semifield):

$$x \otimes -x = x + (-x) = 0 = \mathbb{1}. \tag{2.98}$$

Moreover, $\otimes$ is distributive over $\oplus$:

$$x \otimes (y \oplus z) = x + \max\{y, z\} = \max\{x + y, x + z\} = (x \otimes y) \oplus (x \otimes z)\ . \tag{2.99}$$

An interesting property is that the semiring is idempotent:

$$x \oplus x = \max\{x, x\} = x. \tag{2.100}$$

Overall, this structure satisfies all the axioms of a ring, except that $\oplus$ is not invertible. This is an intuitive phenomenon: taking the maximum of two real numbers is an operation that cannot be inverted, as all the information contained in the smallest number is lost, except that it is smaller than the first one.

Apart from this phenomenon, all classical algebraic computations can be considered in the max-plus semiring, such as exponentiation: for any positive integer n,

$$x^{\otimes n} = \underbrace{x \otimes \cdots \otimes x}_{n \text{ times}} = nx\ , \tag{2.101}$$

or matrix computations:

$$\begin{pmatrix} \mathbb{1} & 3 \\ 2 & 5 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \mathbb{0} \end{pmatrix} = \begin{pmatrix} \mathbb{1} \otimes 1 \oplus 3 \otimes \mathbb{0} \\ 2 \otimes 1 \oplus 5 \otimes \mathbb{0} \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}. \tag{2.102}$$

Example 2.4: A max-plus linear system

Consider the following linear system, with unknown $z = (x, y)^\top \in \mathbb{R}^2_{\max}$:

$$\begin{pmatrix} \mathbb{1} & 2 \\ -4 & \mathbb{1} \end{pmatrix} \otimes \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}. \tag{2.103}$$

Unrolling the max-plus notations, this is equivalent to the following system of equations:

$$\begin{cases} \max\{x, y+2\} = 1 \\ \max\{x-4, y\} = 2. \end{cases} \tag{2.104}$$

The first line of (2.104) is equivalent to:

$$(x = 1 \text{ and } y + 2 \leq 1) \text{ or } (x \leq 1 \text{ and } y + 2 = 1), \tag{2.105}$$

with a similar condition for the second line:

$$(x - 4 = 2 \text{ and } y \leq 2) \text{ or } (x - 4 \leq 2 \text{ and } y = 2). \tag{2.106}$$

The solutions (x, y) of (2.105) and (2.106) are plotted in Figure 2.7, respectively in blue and red. In this example, the two sets do not intersect, therefore the max-plus linear system (2.103) does not have any solutions in $\mathbb{R}^2_{\max}$.

An unpleasant consequence of the non-invertibility of $\oplus$ is that, in general, like in Example 2.4, a max-plus linear system $Ax = b$ has no solutions. This is not an anecdotal phenomenon: in fact it is highly unlikely for a matrix to be surjective or injective in the max-plus sense. A strategy to cope with this issue is called *residuation*. Let us define a natural order relation on $\mathbb{R}_{\max}$:

$$x \preceq y \iff x \oplus y = y. \tag{2.107}$$

Let us define the *residuation* operation by:

$$\begin{aligned} a \backslash b &= \max\{x \mid a \otimes x \preceq b\} \\ &= \max\{x \mid a + x \preceq b\} \\ &= \begin{cases} b - a & \text{if } a \neq \mathbb{0} \\ +\infty & \text{else.} \end{cases} \end{aligned} \tag{2.108}$$

One can see from (2.108) that the residuation operation plays the role of a pseudo-inverse, on the set $\mathbb{R}_{\max}$ completed by $\{+\infty\}$, denoted by $\bar{\mathbb{R}}_{\max}$.

This construction can be extended to matrices. Let $A \in \bar{\mathbb{R}}^{n \times m}_{\max}$. The map:

$$\lambda_A : \bar{\mathbb{R}}^m_{\max} \to \bar{\mathbb{R}}^n_{\max}, \quad x \mapsto Ax \tag{2.109}$$

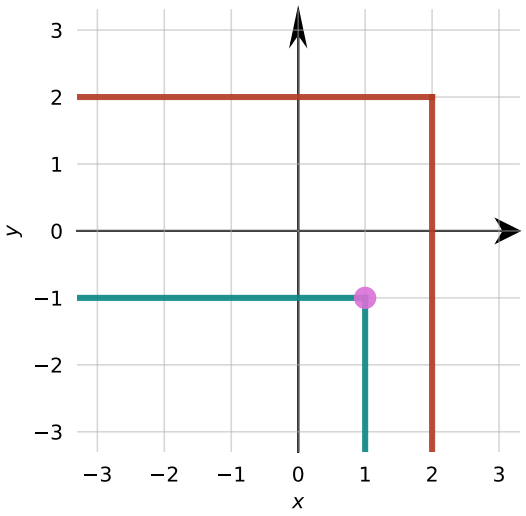


Figure 2.7: Sets of solutions of equations (2.105) and (2.106), respectively in blue and red. The pink dot is the solution of the residuation operation.

is *residuated*, *i.e.*, for any $b \in \bar{\mathbb{R}}_{\max}^n$, the set $\{x \in \bar{\mathbb{R}}_{\max}^m \mid Ax \preceq b\}$ has a maximal element denoted by $A\backslash b$, defined by, for $j \in \{1, \dots m\}$:

$$(A\backslash b)_j = \min_{1 \leq i \leq n} (-A_{ij} + b_i)\,. \tag{2.110}$$

Instead of looking for solutions of $Ax = b$, we can instead look for subsolutions, specifically the largest solution of $Ax \preceq b$, *i.e.*, $A\backslash b$. In Example 2.4, this solution is equal to $(x, y) = (1, -1)$. One can check on Figure 2.7 that it is the largest point in the intersection of the subsolution sets:

$$\begin{cases} x \leq 1 \text{ and } y + 2 \leq 1 \\ x - 4 \leq 2 \text{ and } y \leq 2. \end{cases} \tag{2.111}$$

Furthermore, the equation $Ax = b$ has solutions if and only if $A(A\backslash y) = y$ (Gaubert and Plus, 1997). Note that in Chapter 3, we will use a different notation, A^+b, for the residuation operation. The notation A^+ for the residuation differs from the convention in the max-plus literature (the usual notation being $A^\flat$), and in particular, it does not denote the operator $\oplus_{k=1}^{\infty} A^k$ like in Baccelli et al. (1992).

2.4.2 Max-Plus Linear Parameterizations

In Section 2.3.1, we have presented a parametric representation of a function as a linear combination of basis functions $(\varphi_1, \dots \varphi_k)$:

$$f_\theta(\cdot) = \sum_{j=1}^{k} \theta_j \varphi_j(\cdot). \tag{2.112}$$

One can derive the max-plus counterpart of this linear parameterization as:

$$\begin{aligned} f_\theta(\cdot) &= \bigoplus_{j=1}^{k} \theta_j \otimes \varphi_j(\cdot) \\ &= \max_{1\leq j\leq k} \theta_j + \varphi_j(\cdot). \end{aligned} \tag{2.113}$$

This generates a max-plus linear space of functions which is stable under linear combinations. Indeed, if $f = \bigoplus_{j=1}^{k} \theta_j \otimes \varphi_j$ and $g = \bigoplus_{j=1}^{k} \nu_j \otimes \varphi_j$, then:

$$\begin{aligned} \alpha \otimes f(\cdot) \oplus \beta \otimes g(\cdot) &= \max\{\alpha + f(\cdot), \beta + g(\cdot)\} \\ &= \max\{\max_j(\alpha + \theta_j + \varphi_j(\cdot)), \max_j(\beta + \nu_j + \varphi_j(\cdot))\} \\ &= \max_j \max\{\alpha + \theta_j, \beta + \nu_j\} + \varphi_j(\cdot) \\ &= \bigoplus_j \mu_j + \varphi_j(\cdot), \end{aligned} \tag{2.114}$$

where $\mu = \alpha \otimes \theta \oplus \beta \otimes \nu$. This max-plus linear structure allows defining familiar operations, like projection onto the max-plus span of the φ_j, whose construction will be detailed in Chapter 3. In particular, one can study the approximation properties of specific basis of functions φ, as is done by Akian et al. (2008) for functions of the form $\varphi_j(x) = -a\|x - x_j\|_1$, or $\varphi_j(x) = -c/2\|x - x_j\|_2^2$.

Finally, let us mention that an interesting connection between max-plus function spaces and reproducing kernel Hilbert spaces (which we remind extend linear representations of the form (2.112) in infinite-dimension) has been made recently by Aubin-Frankowski and Gaubert (2022).

2.4.3 Application to Optimal Control

Beyond its theoretical interest, the max-plus algebra is a powerful tool in a large panel of applications, including discrete-event systems and synchronization (Baccelli et al., 1992), systems theory (Cohen et al., 1999), or decision theory (Simon, 1978). Here, we focus on the connection to optimal control, which relies on the intrinsic max-plus structure of optimal control problems (Akian et al., 2008; McEneaney, 2003).

Consider an optimal control problem (with a supremum to fit the max-plus structure):

$$\begin{aligned} V^*(t_0, x_0) = \sup_{u(\cdot)} & \int_{t_0}^{T} L(x(t), u(t))\mathrm{d}t + M(x(T)) \\ \text{s.t. } & \forall t \in [t_0, T], \;\; \dot{x}(t) = f(x(t), u(t)) \\ & x(t_0) = x_0. \end{aligned} \tag{2.115}$$

V^* is the optimal value function, and is such that $V^*(T, \cdot) = M$, and V^* verifies the HJB equation. We define, for $t \in [0, T]$, the evolution semigroup S^t of the control problem (or the Lax-Oleinik semigroup), which, to any terminal cost function M, associates the solution $V^*(T-t, \cdot)$. In particular, $S^0(M) = M$, and hence $S^0 = I$. The collection $\{S^t\}$ has a semigroup structure, directly inherited from Bellman's optimality principle:

$$S^t \circ S^{t'} = S^{t+t'}. \tag{2.116}$$

An interesting property of the semigroup is that it is max-plus linear. This property has been first remarked by Maslov (1973), as a "superposition principle", and refered to by Fleming and McEneaney (2000) as max-plus linearity. Indeed, one can easily check using (2.115) that, for $c \in \mathbb{R}$:

$$S^t(c \otimes M) = S^t(c + M) = c + S^t(M) = c \otimes S^t(M). \tag{2.117}$$

Similarly, it can be proved that, for M, M' functions from $\mathcal{X} \to \mathbb{R}$:

$$S^t(M \oplus M') = S^t(M) \oplus S^t(M'). \tag{2.118}$$

This max-plus linearity explains the particular interest of function representations of the form (2.113) for the optimal value function, as we will see in more details in Chapter 3.

Chapter 3

Max-Plus Discretization of Deterministic Markov Decision Processes

Abstract. *We consider deterministic continuous-state Markov decision processes (MDPs). We apply a max-plus linear method to approximate the value function with a specific dictionary of functions that leads to an adequate state-discretization of the MDP. This is more efficient than a direct discretization of the state space, typically intractable in high dimension. We propose a simple strategy to adapt the discretization to a problem instance, thus mitigating the curse of dimensionality. We provide numerical examples showing that the method works well on simple MDPs.*

This chapter is based on our work *Max-Plus Linear Approximations for Deterministic Continuous-State Markov Decision Processes*, with Francis Bach, published in the IEEE Control Systems Letters, 2020.

Contents

3.1 Introduction

Reinforcement learning problems (Sutton and Barto, 2018) are generally formulated as Markov decision processes (MDPs). Dynamic programming provides simple algorithms, such as value iteration, to compute the optimal value function and an optimal policy for a discrete MDP, when the model is known.

Yet many problems formalized as MDPs are time- and space-discretizations of control problems, with a continuous underlying state space. To faithfully reproduce the dynamics of the control problem, one needs to compute a sharp space-discretization, subject to the curse of dimensionality: for high-dimensional problems, the space-discretized MDP will not even fit in memory.

Following the method of McEneaney (2003) and Akian et al. (2008), we compute approximations of the optimal value function for deterministic MDPs, namely max-plus linear approximations within a dictionary of functions. These methods have been developed for optimal control and deal with continuous state spaces. For certain choices of function dictionaries, they can be viewed as an efficient way to discretize the state-continuous MDP while preserving its dynamics. Adaptively choosing the basis functions used to approximate the value function is a way to circumvent the curse of dimensionality when the true value function has a sparse representation.

Our contributions are the following:

- we propose in Section 3.3.4 a new approximation method to solve subproblems appearing in the max-plus value iteration algorithm, namely to optimize some objectives over the state-space with gradient ascent;
- we present a specific dictionary of functions simplifying the method in Section 3.3.3, and show how it can be used to build an adaptive discretization of the state space in Section 3.6;
- in Section 3.7, we provide numerical simulations on MDPs where this adaptive max-plus approximation method computes nearly optimal policies with significantly less parameters than discretized value iteration.

Setting. We consider (Hernández-Lerma and Lasserre, 2012) a deterministic, time-homogeneous, infinite-horizon, discounted MDP defined by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a bounded reward function $r : \mathcal{S} \times \mathcal{A} \to [-R, R]$ for some $R \geq 0$, a dynamics $\varphi_\cdot(.) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and a discount factor $0 \leq \gamma < 1$, with the following assumptions:

1. the state space $\mathcal{S}$ is a bounded subset of $\mathbb{R}^d$ ($d \geq 1$);
2. the action space $\mathcal{A}$ is finite.

We want to approximate the optimal value function $V^* : \mathcal{S} \to \mathbb{R}$ corresponding to an optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ maximizing the cumulative discounted reward. The greedy policy π corresponding to a value function V

is obtained by:

$$\pi(s) \in \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ r(s,a) + \gamma V(\varphi_a(s)). \tag{3.1}$$

The value iteration algorithm consists in computing V^* as the unique fixed point of the Bellman operator $T : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ (where $\mathbb{R}^{\mathcal{S}}$ denotes the set of functions from $\mathcal{S}$ to $\mathbb{R}$) defined as:

$$TV(s) := \max_{a \in \mathcal{A}}\ r(s,a) + \gamma V(\varphi_a(s)). \tag{3.2}$$

The value iteration algorithm iteratively computes the recursion $V_{k+1} = TV_k$ that converges to V^*, with linear rate since T is strictly contractive with factor $\gamma < 1$. But if $\mathcal{S}$ is a finite set, it requires $O(|\mathcal{A}| \cdot |\mathcal{S}|)$ computations, and the storage of $O(|\mathcal{S}|)$ values of V_k at each step.

From now on, we consider that $\mathcal{S}$ is a compact, but potentially not discrete set. In this case, one can directly look for a discretization of the MDP and perform value iteration, but this will become intractable in high dimension, since the size of the discretized state space grows exponentially with the dimension. Alternatively, one can consider the space-continuous MDP, and compute an approximation of the optimal value function, without having to discretize the MDP.

3.2 Max-Plus Linear Approximations

Let $\mathcal{W}$ be a finite dictionary of functions $w : \mathcal{S} \to \mathbb{R}$. The value function can be approximated by a "linear" combination of functions in $\mathcal{W}$, with an adapted definition of linearity. The max-plus semiring (Gaubert and Plus, 1997) is defined as $(\mathbb{R} \cup \{-\infty\}, \oplus, \otimes)$, where $\oplus$ represents the maximum operator, and $\otimes$ represents the usual sum. Like linear combinations in the usual ring, for $\alpha \in \mathbb{R}^{\mathcal{W}}$, we define the max-plus linear combination:

$$V(s) = \bigoplus_{w \in \mathcal{W}} \alpha(w) \otimes w(s) = \max_{w \in \mathcal{W}} \alpha(w) + w(s). \tag{3.3}$$

The Bellman operator's structure is naturally compatible with max-plus operations, as it is max-plus additive and homogeneous: for $c, V, V' \in \mathbb{R}^{\mathcal{S}}$, we have

$$T(V \oplus V') = T(\max\{V, V'\}) = \max\{TV, TV'\} = TV \oplus TV' \tag{3.4}$$

$$T(c \otimes V) = T(c + V) = \gamma c + TV = c^{\otimes \gamma} TV. \tag{3.5}$$

The basis functions used in Akian et al. (2008) and McEneaney (2003) are smooth ($w_i(s) := -c\|s - s_i\|^2$ for some $s_i \in \mathcal{S}$) or Lipschitz-continuous ($w_i(s) := -c\|s - s_i\|$). However, the scale $c > 0$ of such functions must be chosen according to the regularity of the true value function. Since it is unknown in practice, it needs to be tuned as a hyperparameter. Other somewhat simpler choices of basis functions can be considered as well. Let $(A(w_1), ..., A(w_n))$ be a partition of the state space, where each w_i is defined as the max-plus indicator of a set $A(w_i)$:

$$w_i(s) := \begin{cases} 0 & \text{if } s \in A(w_i) \\ -\infty & \text{otherwise.} \end{cases} \tag{3.6}$$

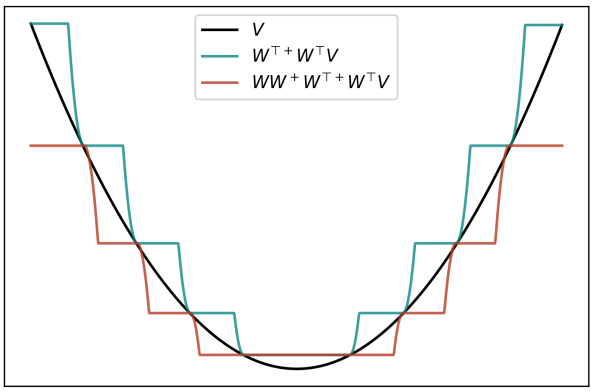


Figure 3.1: A function V, its upper-projection onto the dictionary of soft indicators (see Section 3.3.3) in blue, and the lower-projection of the upper-projection in red.

Then the max-plus linear combinations of $(w_1, ..., w_n)$ span the set of value functions that are piecewise constant with respect to the partition (Bach, 2019). This is thus a way to discretize the value function.

Following the notations of Bach (2019), for a given dictionary of functions $\mathcal{W}$, we define the following four operators:

$$W : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{S}}, \quad W\alpha(s) := \max_{w \in \mathcal{W}} \alpha(w) + w(s) \tag{3.7}$$

$$W^+ : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{W}}, \quad W^+V(w) := \inf_{s \in \mathcal{S}} V(s) - w(s) \tag{3.8}$$

$$W^\top : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{W}}, \quad W^\top V(w) := \sup_{s \in \mathcal{S}} V(s) + w(s) \tag{3.9}$$

$$W^{\top+} : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{S}}, \quad W^{\top+}\alpha(s) := \min_{w \in \mathcal{W}} \alpha(w) - w(s). \tag{3.10}$$

W maps a vector α to a function $W\alpha$ which is the max-plus linear combination of the dictionary $\mathcal{W}$ with coefficient α. W^+ is known as the residuation (Cohen et al., 2004) of W and acts as a pseudo-inverse:

$$W\alpha \leq V \Leftrightarrow \alpha \leq W^+V. \tag{3.11}$$

The transposed notation for $W^\top$ comes from the definition of a max-plus *dot product* (it is only a max-plus bilinear form) between functions on $\mathcal{S}$, which will be used in the rest of the chapter:

$$\forall z, w \in \mathbb{R}^{\mathcal{S}}, \quad \langle z, w \rangle := \sup_{s \in \mathcal{S}} z(s) + w(s). \tag{3.12}$$

The lower-projection of a function $V \in \mathbb{R}^{\mathcal{S}}$ onto the span of the dictionary is computed as WW^+V, and $W^{\top+}W^\top V$ is its upper-projection (see Figure 3.1). Both projection operators WW^+ and $W^{\top+}W^\top$ are idempotent and non-expansive for the ℓ_∞ norm.

3.3 Approximate Value Iteration

These max-plus tools can be used to compute a tractable approximation of the optimal value function of an MDP.

3.3.1 Projection Method

A simple way to approximate the value function has been proposed in Akian et al. (2008), as an extension of the method of McEneaney (2003), both for control problems. Following Chandrashekar and Bhatnagar (2014) and Bach (2019), we apply it to MDPs. The idea is to represent the value function as a max-plus linear combination in a dictionary of functions, and to apply alternatingly the Bellman operator and a projection onto the span of the dictionary: $V_{k+1} = WW^+TV_k$. Hence if V_k is represented as $W\alpha_k$, then α_{k+1} is given by $\alpha_{k+1} = W^+TW\alpha_k$, where the operator $W^+TW : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{W}}$ is computed by:

$$\alpha_{k+1}(w) = \inf_{s\in\mathcal{S}} \max_{w'\in\mathcal{W}} \gamma\alpha_k(w') + Tw'(s) - w(s). \tag{3.13}$$

This computation is a min/max problem, which is not easy to solve in general. If $\mathcal{S}$ is finite, this requires to compute $|\mathcal{S}| \cdot |\mathcal{W}|$ values at each iteration.

3.3.2 Variational Method

A slightly more involved approximation method has been also proposed by Akian et al. (2008). Let us define two dictionaries of functions $\mathcal{W}$ and $\mathcal{Z}$. $\mathcal{W}$ plays the same role as before, while $\mathcal{Z}$ is a set of test functions which can be taken equal to $\mathcal{W}$. The value iteration recursion $V_{k+1} = TV_k$ is replaced by a variational formulation:

$$\langle z, V_{k+1}\rangle = \langle z, TV_k\rangle \quad \forall z \in \mathcal{Z}, \tag{3.14}$$

of which we consider the maximal solution in span(W), given by (Akian et al., 2008, Proposition 4): $V_{k+1} = WW^+Z^{\top+}Z^\top TV_k$. It can be interpreted as a first projection on the min-plus span generated by $\mathcal{Z}$, before a second projection on the max-plus span of $\mathcal{W}$. Again if V_k is represented by $W\alpha_k$, we have the following recursion:

$$\alpha_{k+1} = W^+Z^{\top+}Z^\top TW\alpha_k. \tag{3.15}$$

The operator $W^+Z^{\top+}Z^\top TW : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{W}}$ decomposes as $M \circ K$, with $K = Z^\top TW : \mathbb{R}^{\mathcal{W}} \to \mathbb{R}^{\mathcal{Z}}$ and $M = W^+Z^{\top+} : \mathbb{R}^{\mathcal{Z}} \to \mathbb{R}^{\mathcal{W}}$. The recursion may be recast as:

$$\begin{aligned}
\beta_{k+1}(z) = K\alpha_k(z) &= \sup_{s\in\mathcal{S}} z(s) + \max_{w\in\mathcal{W}} \gamma\alpha_k(w) + Tw(s) \\
&= \max_{w\in\mathcal{W}} \gamma\alpha_k(w) + \langle z, Tw\rangle \qquad (3.16)\\
\alpha_{k+1}(w) = M\beta_{k+1}(w) &= \inf_{s\in\mathcal{S}} -w(s) + \min_{z\in\mathcal{Z}} \beta_{k+1}(z) - z(s) \\
&= \min_{z\in\mathcal{Z}} \beta_{k+1}(z) - \langle z, w\rangle. \qquad (3.17)
\end{aligned}$$

The operator $W^+Z^{\top+}Z^\top TW$ is a γ-contraction, hence the recursion will converge with linear rate to the unique fixed point. An interesting property is that the $|\mathcal{Z}| \cdot |\mathcal{W}|$ values $\langle z, Tw\rangle$ for $(z, w) \in \mathcal{Z} \times \mathcal{W}$ can be precomputed at a cost that is independent of the horizon $1/(1-\gamma)$ of the MDP. The main difficulty here is their prior computation. Unlike in Bach (2019) where $\mathcal{S}$ is finite, for a continuous state space these computations might only be performed approximately.

3.3.3 Basis Functions and Clustered MDP

Discrete versions of the MDP can be built from the preceding approximation methods with $\mathcal{W}$ a set of max-plus indicators corresponding to a partition of the state space, as mentioned earlier. Indeed, when $\mathcal{W} = \mathcal{Z}$ and the $(w_i)_{1\leq i\leq n}$ are max-plus indicators, the above operator M is the identity and $W^+W^{\top+}W^\top TW = W^\top TW$ (a max/max problem), to be compared with W^+TW (a min/max problem) for the projection method. Note that with the approximate indicators introduced below, M will not be equal to the identity, even though $\mathcal{W} = \mathcal{Z}$.

With max-plus indicators and the variational method, the approximate value iteration becomes:

$$\alpha_{k+1}(w) = \max_{w'\in\mathcal{W}} \langle w, Tw'\rangle + \gamma\alpha_k(w'), \tag{3.18}$$

which we interpret as classical value iteration on the MDP formed with the clusters $(A(w))_{w\in\mathcal{W}}$ as states, and as rewards the maximal achievable reward going from one cluster to the other, that is:

$$\begin{aligned} R(w,w') = \langle w, Tw'\rangle &= \sup_{s\in\mathcal{S}} w(s) + Tw'(s) \\ &= \sup_{s\in A(w)} \max_{\substack{a\in\mathcal{A} \text{ s.t.} \\ \varphi_a(s)\in A(w')}} r(s,a), \end{aligned} \tag{3.19}$$

with $R(w,w') = -\infty$ if $\{(s,\varphi_a(s)) \mid s\in\mathcal{S}, a\in\mathcal{A}\}\cap A(w)\times A(w') = \varnothing$. This reduced problem is appealing but hard to solve in a continuous state space. Even finding if $R(w,w')$ is finite is both a controllability and reachability problem (Liberzon, 2011), whose solution is not straightforward. A differentiable version of the max-plus indicators is the following:

$$w(s) = -c\,\mathrm{dist}(s, A(w))^2, \tag{3.20}$$

where $\mathrm{dist}(s, A(w))$ is the euclidean distance between s and the set $A(w)$, and $c > 0$ is a hyperparameter, typically chosen large compared to the scale of the true value function. We refer to such basis functions as soft indicators. When $c\to+\infty$, we recover the preceding clustered MDP and elements in the span of W are *almost* (asymptotically) piecewise constant with respect to the partition (see Figure 3.1).

3.3.4 Oracle Subproblem

We now take a closer look at the subproblems that must be solved before running the approximate value iteration recursion, namely $\langle z, w\rangle$ and $\langle z, Tw\rangle$ for the variational method. First, $\langle z, w\rangle$ is independent of the MDP and can be computed in closed form for general choices of dictionaries, and:

$$\begin{aligned} \langle z, Tw\rangle &= \sup_{s\in\mathcal{S}} z(s) + Tw(s) \\ &= \sup_{s\in\mathcal{S},\ a\in\mathcal{A}} z(s) + r(s,a) + \gamma w(\varphi_a(s)). \end{aligned} \tag{3.21}$$

This is a discrete-time control problem, easier than the original one (finding the optimal value function) since its horizon is one time step. As mentioned by Akian et al. (2008), this is a perturbed version of the computation of $\langle z, w\rangle$ as soon as T is close to the identity, that is, in the context of optimal control when the time-discretization of the MDP is small.

In Akian et al. (2008), $\langle z, Tw\rangle$ is approximated using the Hamiltonian of a control problem. For general MDPs, we may look at this problem from a different perspective. It is an optimization problem, and, as noted by Akian et al. (2008), even though computing $\langle z, Tw\rangle$ is not a concave maximization problem, choosing strongly concave basis functions z and w has a regularizing effect.

Hence an approximation of $\langle z, Tw\rangle$ can be computed by gradient ascent on

$$f_a(s) := z(s) + r(s,a) + \gamma w(\varphi_a(s)), \tag{3.22}$$

for each $a \in \mathcal{A}$, and then taking the maximum on a. For differentiable z, w, φ_a and $r(.,a)$, f_a is differentiable with:

$$\nabla f_a(s) = \nabla z(s) + \nabla r(s,a) + \gamma J_{\varphi_a}(s)^\top \nabla w(\varphi_a(s)), \tag{3.23}$$

where J_{φ_a} denotes the Jacobian of φ_a and ∇r is the gradient of r with respect to s. Seeing this problem like Akian et al. (2008) as a perturbation of $\langle z, w\rangle$, an efficient initialization for gradient ascent on this problem is given by $s_0 \in \operatorname{argmax}_s z(s) + w(s)$. Furthermore, for continuous basis functions, reward function and dynamics, since $\mathcal{S}$ is compact by assumption, the supremum in $\langle z, Tw\rangle$ is a maximum attained at some $(s,a) \in \mathcal{S} \times \mathcal{A}$. The full procedure to obtain the approximate value function is described in Algorithm 1.

As noted in Bach (2019), the Bellman operator T can be replaced by T^ρ for some integer $\rho \geq 1$, replacing accordingly γ by γ^ρ. This makes the computation of $\langle z, T^\rho w\rangle$ more complicated, as it requires to run $|\mathcal{A}|^\rho$ gradient ascents. A simplification is to consider only sequences of constant actions for ρ steps.

Comparison with existing methods. Approximate value iteration is usually performed by fitted value iteration (Sutton and Barto, 2018), with a linear parameterization of the value function. With max-plus parameterizations, the projections are computed efficiently, which spares the repeated use of stochastic optimization, and leads to an explicit error analysis. Nonlinear approximations can be handled with Q-learning (see Mehta and Meyn (2009) for continuous MDPs), with weaker convergence guarantees (Sutton and Barto, 2018).

Projection method. The projection method (3.13) requires to approximate $\langle w, -Tw'\rangle$, a problem which does not benefit from the same regularizing effect of concave basis functions. Indeed, taking for T the identity operator, this is not even a concave problem. Numerically, this problem is more challenging than the previous one. This concern has already been raised by McEneaney (2003) for unbounded $\mathcal{S}$, with the caveat that the basis functions must be chosen such that $\langle w, -Tw'\rangle$ is finite, *i.e.*, such that $w - Tw'$ has a finite maximum. In practice this must be checked for each particular dynamics. This is the reason why we prefer to focus on the variational method, possibly with $\mathcal{W} = \mathcal{Z}$.

3.4 Error Analysis

3.4.1 Error Decomposition

The operator $\bar{T} := WW^+Z^{\top+}Z^\top T$ is γ-contractive, since T is γ-contractive and both WW^+ and $Z^{\top+}Z^\top$ are non-expansive. If the $\langle z, w\rangle$ and $\langle z, Tw\rangle$ are computed exactly, the error of the *exact* max-plus approximation is controlled only by projection errors. In practice, the values $K_{z,w} := \langle z, Tw\rangle$ are approximated

Algorithm 1 Max-plus Approximate Value Iteration

Input: MDP, $\mathcal{W}$ and $\mathcal{Z}$, gradient steps k, step size ξ
Output: approximate value function V

Precomputations:
for $z \in \mathcal{Z}, w \in \mathcal{W}$ **do**
  $s, \langle z, w\rangle \leftarrow \operatorname{argmax}, \max_{s\in\mathcal{S}}\ z(s) + w(s)$
  for $a \in \mathcal{A}$ **do**
    $\langle z, Tw\rangle \leftarrow z(s) + r(s,a) + w(\varphi_a(s))$
    for $i = 1$ **to** k **do**
      $g \leftarrow \nabla z(s) + \nabla r(s,a) + J_{\varphi_a}(s)^\top \nabla w(\varphi_a(s))$
      $s \leftarrow s + \xi g$
      $f \leftarrow z(s) + r(s,a) + w(\varphi_a(s))$
      $\langle z, Tw\rangle \leftarrow \max\{f, \langle z, Tw\rangle\}$
Reduced value iteration:
$\alpha \leftarrow 0$
repeat
  for $z \in \mathcal{Z}$ **do**
    $\beta(z) \leftarrow \max_{w\in\mathcal{W}}\ \gamma\alpha(w) + \langle z, Tw\rangle$
  for $w \in \mathcal{W}$ **do**
    $\alpha(w) \leftarrow \min_{z\in\mathcal{Z}}\ \beta(z) - \langle z, w\rangle$
until convergence
return $V = W\alpha$

by some $\hat{K}_{z,w}$ obtained by gradient ascent with some error due to the finite number of iterations and to the non-concavity of the objective function.

Proposition 1 (Decomposition of errors)**.** *Let V^* be the optimal value function of the MDP, $\hat{V} = W\hat{\alpha}$, where $\hat{\alpha}$ is the fixed point of $M \circ \hat{K}$, and*

$$\|\hat{K} - K\|_\infty := \sup_{z\in\mathcal{Z}, w\in\mathcal{W}} |\hat{K}_{z,w} - K_{z,w}|. \tag{3.24}$$

Then:

$$\begin{aligned}\|\hat{V} - V^*\|_\infty \leq \frac{1}{1-\gamma}\Big(&\|WW^+V^* - V^*\|_\infty \\ &+\|Z^{\top+}Z^\top V^* - V^*\|_\infty + \|\hat{K} - K\|_\infty\Big).\end{aligned} \tag{3.25}$$

Proof. Let V_∞ be the unique fixed point of $\bar{T}$. We have $V_\infty = \bar{T}V_\infty$, $V^* = TV^*$ and $\bar{T}$ is γ-contractive. Then, adding and subtracting $\bar{T}V^*$, we get:

$$\begin{aligned}\|V_\infty - V^*\| &\leq \|\bar{T}V_\infty - \bar{T}V^*\| + \|\bar{T}V^* - V^*\| \\ &\leq \gamma\|V_\infty - V^*\| + \|WW^+Z^{\top+}Z^\top V^* - V^*\|.\end{aligned} \tag{3.26}$$

Grouping the instances of $\|V_\infty - V^*\|$ in (3.26), and adding and subtracting WW^+V^*, we obtain:

$$(1-\gamma)\|V_\infty - V^*\| \leq \|WW^+Z^{\top+}Z^\top V^* - WW^+V^*\|$$

$$
\begin{aligned}
&+ \|WW^+V^* - V^*\| \\
&\leq \|Z^{\top+}Z^\top V^* - V^*\| + \|WW^+V^* - V^*\|,
\end{aligned} \tag{3.27}
$$

because the projection WW^+ is non-expansive.

On the other hand, inserting $\bar{T}\hat{V}$, and using the γ-contractivity of $\bar{T}$, we get:

$$
\|\hat{V} - V_\infty\| \leq \|\hat{V} - \bar{T}\hat{V}\| + \gamma\|\hat{V} - V_\infty\|. \tag{3.28}
$$

Since $\hat{V} = W\hat{\alpha}$ and $\hat{\alpha} = M \circ \hat{K}\hat{\alpha}$, this writes:

$$
\begin{aligned}
(1-\gamma)\|\hat{V} - V_\infty\| &\leq \|WW^+Z^{\top+}\hat{K}\hat{\alpha} - WW^+Z^{\top+}K\hat{\alpha}\| \\
&\leq \|Z^{\top+}\hat{K}\hat{\alpha} - Z^{\top+}K\hat{\alpha}\| \\
&\leq \|\hat{K}\hat{\alpha} - K\hat{\alpha}\| \\
&\leq \|\hat{K} - K\|_\infty.
\end{aligned} \tag{3.29}
$$

The last two inequalities result from the reverse triangle inequality for the infinity norm, which writes:

$$
|\max_w \phi(w) - \max_w \psi(w)| \leq |\|\phi\|_\infty - \|\psi\|_\infty| \leq \|\phi - \psi\|_\infty. \tag{3.30}
$$

Indeed, for any $s \in \mathcal{S}$, and any functions u and v :

$$
\begin{aligned}
|Z^{\top+}u(s) - Z^{\top+}v(s)| &= |\min_z\{u(z) - z(s)\} - \min_z\{v(z) - z(s)\}| \\
&= |\max_z\{z(s) - u(z)\} - \max_z\{z(s) - v(z)\}| \\
&\leq \|u - v\|_\infty.
\end{aligned} \tag{3.31}
$$

Similarly, from the definitions of $\hat{K}$ and K, we get:

$$
\begin{aligned}
|\hat{K}\hat{\alpha}(z) - K\hat{\alpha}(z)| &= |\max_w\{\gamma\hat{\alpha}(w) + \hat{K}_{z,w}\} - \max_w\{\gamma\hat{\alpha}(w) + K_{z,w}\}| \\
&\leq \|\hat{K} - K\|_\infty.
\end{aligned}
$$

The result follows from combining the upper-bound (3.29) on $\|\hat{V} - V_\infty\|$ with the upper-bound (3.27) on $\|V_\infty - V^*\|$. □

In numerical implementations, the fact that the reduced value iteration algorithm is stopped after a finite number of iterations causes a last source of error. Since the convergence is fast (linear), it will often be negligible compared to the other approximations.

Proposition 2 (Convergence and optimality criterion for reduced value iteration)**.** *For $\alpha_0 \in \mathbb{R}^{\mathcal{W}}$, let $\alpha_{k+1} = M\hat{K}\alpha_k$ for $k \geq 1$. Then denoting as $\hat{\alpha}$ the unique fixed point of $M \circ \hat{K}$:*

$$
\|W\alpha_k - W\hat{\alpha}\|_\infty \leq \|\alpha_k - \hat{\alpha}\|_\infty \leq \gamma^k\|\alpha_0 - \hat{\alpha}\|_\infty. \tag{3.32}
$$

3.4.2 Projection Error

For any $V \in \mathbb{R}^{\mathcal{S}}$, $W^{\top +}W^{\top}V = -WW^{+}(-V)$, so we can only consider the projection error for WW^{+}, *i.e.*, the lower projection.

Proposition 3 (Approximation properties of soft-indicators)**.** *Let $c_1, c_2 > 0$ and $(A_1, ..., A_n)$ a partition of $\mathcal{S}$ where each A_i is convex, compact and non-empty, and let $D = \max_{1 \leq i \leq n} \operatorname{diam}(A_i)$, where* diam *denotes the diameter* $\operatorname{diam}(A_i) = \max_{s,s' \in A_i} \|s - s'\|$*. Let $\mathcal{W}_1 = \{w_1^1, ..., w_n^1\}$ and $\mathcal{W}_2 = \{w_1^2, ..., w_n^2\}$ defined by:*

$$\forall i \in \{1, ..., n\},\ \forall s \in \mathcal{S}, \quad \begin{cases} w_i^1(s) = -c_1 \operatorname{dist}(s, A_i) \\ w_i^2(s) = -c_2 \operatorname{dist}(s, A_i)^2. \end{cases} \tag{3.33}$$

If V is Lispchitz continuous with Lipschitz constant L, and $c_1 \geq L$, $c_2 \geq \frac{L}{4D}$, then

$$\|V - W_1 W_1^{+} V\|_{\infty} \leq LD \tag{3.34}$$

$$\|V - W_2 W_2^{+} V\|_{\infty} \leq LD + \frac{L^2}{4c_2} \leq 2LD. \tag{3.35}$$

Proof. For $s \in \mathcal{S}$, we have:

$$WW^{+}V(s) \leq \max_{w} w(s) + V(s) - w(s) \leq V(s). \tag{3.36}$$

On the other hand, there exists $i \in \{1, ..., n\}$ such that $s \in A_i$. Then:

$$\begin{aligned} WW^{+}V(s) &= \max_{w} w(s) + \inf_{s'} V(s') - w(s') \\ &\geq \underbrace{w_i(s)}_{=0} + \inf_{s'} V(s') - w_i(s'). \end{aligned} \tag{3.37}$$

The Lipschitz continuity of V implies that for $p \in \{1, 2\}$:

$$WW^{+}V(s) \geq V(s) + \inf_{s' \in \mathcal{S}} \left\{ c_p \operatorname{dist}(s', A_i)^p - L\|s - s'\| \right\}, \tag{3.38}$$

and the results follow from using that $\|s - s'\| \leq \operatorname{diam}(A_i) + \operatorname{dist}(s', A_i)$.

□

Unlike for smooth or Lipschitz-continuous basis functions (Akian et al., 2008; Bach, 2019), there is no dependency in c in the bound, for c large enough. This avoids oscillations of the approximation when c is chosen too large and simplifies parameter selection.

3.5 Comparison with the Method of Akian, Gaubert & Lakhoua for Control Problems

Deterministic MDPs and optimal control problems are closely related. Applying our method to an MDP that is a time-discretization of a control problem is similar to directly applying the original method by Akian et al.

(2008) to the control problem. Let $0 \leq \eta < 1$ be a discount factor, $\bar{r} : \mathcal{S} \times \mathcal{A} \to [-R, R]$, $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and define the optimal control problem (Fleming and Soner, 2006):

$$\sup_{a(.)} \int_0^{+\infty} \eta^t \bar{r}(s(t), a(t)) \mathrm{d}t, \tag{3.39}$$

with $s(0) = s_0$, and for all $t \geq 0$, $\dot{s}(t) = f(s(t), a(t))$, $(s(t), a(t)) \in \mathcal{S} \times \mathcal{A}$.

3.5.1 Time-Discretization of a Control Problem

A control problem can be approximated by a state-continuous MDP by time-discretization, using a semi-Lagragian scheme (Capuzzo Dolcetta, 1983; Falcone, 1987). The corresponding time-discretized MDP with step $\tau > 0$ and Euler scheme is:

$$r(s, a) = \tau \bar{r}(s, a), \quad \varphi_a(s) = s + \tau f(s, a), \quad \gamma = \eta^\tau. \tag{3.40}$$

For $\tau > 0$, the continuous- and discrete-time Bellman operators $S_\tau, T_\tau : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$ are defined by, for each $u \in \mathbb{R}^{\mathcal{S}}$:

$$(S_\tau u)(s) := \sup_{a(.)} \int_0^\tau \eta^t \bar{r}(s(t), a(t)) \mathrm{d}t + \eta^\tau u(s(\tau)) \tag{3.41}$$

$$(T_\tau u)(s) := \max_a \tau \bar{r}(s, a) + \eta^\tau u(s + \tau f(s, a)). \tag{3.42}$$

Under regularity assumptions, the value function of the MDP converges to the value function of the control problem (Falcone and Ferretti, 2013). This is obtained in a similar way as the Hamilton-Jacobi-Bellman (HJB) equation (Fleming and Soner, 2006). Let us also mention a recent analysis of time-discretization in reinforcement learning by Tallec et al. (2019).

3.5.2 Hamiltonian Approximation for the Oracle Subproblem

For a continuous-time control problem, in the max-plus approximation method, the role of T is played by the continuous Bellman operator S_τ. The HJB equation provides a first order approximation (Akian et al., 2008) of $S_\tau w$ with respect to the horizon τ:

$$\begin{aligned} S_\tau w(s) &= \sup_{a(.)} \int_0^\tau \eta^t \bar{r}(s(t), a(t)) \mathrm{d}t + \eta^\tau w(s(t + \tau)) \\ &= \sup_{a \in \mathcal{A}} \tau \bar{r}(s, a) + o(\tau) + (1 + \tau \log \eta + o(\tau)) \cdot (w(s) + \tau \nabla w(s) \cdot f(s, a) + o(\tau)) \\ &= (1 + \tau \log \eta) w(s) + \tau H(s, \nabla w) + o(\tau), \end{aligned} \tag{3.43}$$

where $H(s, p) := \sup_{a \in \mathcal{A}} \bar{r}(s, a) + p \cdot f(s, a)$ is the Hamiltonian of the control problem. Instead of optimizing over $\mathcal{S}$, a second approximation made by Akian et al. (2008) is to consider only

$$s_0 \in S(z, w) := \operatorname*{argmax}_s z(s) + w(s), \tag{3.44}$$

since $\langle z, Tw \rangle$ is a perturbation of $\langle z, w \rangle$ for τ small. The final approximation is:

$$\langle z, Tw \rangle \simeq \sup_{s \in S(z,w)} z(s) + (1 + \tau \log \eta) w(s) + \tau H(s, \nabla w). \tag{3.45}$$

This is a valid approximation up to $O(\tau\sqrt{\tau})$ terms, if the Hamiltonian is Lipschitz-continuous and $z + w$ is strongly concave. This prevents the use of Lipschitz bases for $\mathcal{Z}$ and $\mathcal{W}$ at the same time in Akian et al. (2008). Without these assumptions, the approximation is weaker, in $O(\tau)$, breaking the convergence of the method. In this case, one cannot avoid optimizing on s. McEneaney (2003) and Akian et al. (2008) use the first order approximation, but τ is a parameter of their method that can be made arbitrarily small. In the context of MDPs, τ is fixed and in principle it cannot be modified while solving the MDP. Besides, some MDPs are not natural time-discretizations of control problems.

For control problems, time-discretization and Hamiltonian approximation result in the same approximation of $\langle z, Tw\rangle$, up to $o(\tau)$ terms (or $O(\tau^2)$ assuming more regularity on w):

$$\begin{aligned}\hat{K}_{z,w} &= \sup_{s,a}\ z(s) + \tau\bar{r}(s,a) + \eta^{\tau} w(s + \tau f(s,a)) \\ &= \sup_{s}\ z(s) + (1 + \tau\log\eta)w(s) + \tau H(s, \nabla w) + o(\tau). \end{aligned} \tag{3.46}$$

If τ is not negligible, the Hamiltonian approximation is no longer valid, nor is the approximate computation of $\langle z, S_\tau w\rangle$. On the other hand, the computation of $\langle z, T_\tau w\rangle$ is still valid, but the MDP no longer approximates the control problem.

After convergence of the reduced value iteration algorithm, our method provides an approximation of the value function of the control problem with an error of order

$$O\left(D/\tau + \tau + \|\hat{K} - K\|_\infty/\tau\right), \tag{3.47}$$

where D is the maximal diameter of the partition, which is similar to Akian et al. (2008). Reaching a fixed precision requires a number of basis functions exponential in the dimension d. Exploiting the structure of the problem like Gaubert et al. (2011) may reduce this effect.

Remark on the use of T^ρ. As previously mentioned, T_τ^ρ can be used for $\rho \geq 1$, instead of T_τ. In the error bounds, τ is replaced by $\tau\rho$, which can be advantageous for D fixed. Considering only constant actions during ρ steps, T_τ^ρ is very close to $T_{\tau\rho}$, up to the Euler scheme used to compute the dynamics (ρ steps of size τ *vs*. one step of size $\tau\rho$).

3.6 Adaptive Selection of Basis Functions

From a partition $(A_1, ..., A_n)$ of the state space, we define a dictionary $\mathcal{W} = \mathcal{Z}$ of soft-indicators $w_i(.) = -c\ \mathrm{dist}(., A_i)^2$. Running Algorithm 1 with this dictionary returns a value function that is *almost* piecewise constant with respect to the partition (when c is large). This is a way to discretize the MDP, but the performance of the final policy will depend on the partition (Bernstein and Shimkin, 2008; Munos and Moore, 2002). Typically, a uniform partition of $\mathcal{S}$ might not be the best choice for all MDPs. For instance some areas of $\mathcal{S}$ with very low optimal value function are usually not encountered in optimal trajectories, hence spending computational power there would be useless. On the contrary, a sharper approximation of the value function in other areas is critical to the performance of the policy.

We propose an algorithm to build the partition adaptively, with a simple greedy heuristic. Starting from a coarse partition, we compute the approximate value function, and then we select one of the $(A_i)_{1\leq i\leq n}$ that

we want to refine, according to some criterion to be described later. Then, we split this cluster into new sub-clusters partitioning it, and replace it by them in the partition. If the clusters are rectangular parallelepipeds in dimension d, a simple splitting strategy is to subdivide it into 2^d smaller parallelepipeds, by a middle cut along each dimension. In a two-dimensional state space, this corresponds to building a quadtree (Finkel and Bentley, 1974). Formally, a cluster A with a soft-indicator w is split into $C_1, ..., C_{2^d}$, such that:

$$A = \bigcup_{j=1}^{2^d} C_j \quad \text{with } \forall i \neq j \in \{1, ..., 2^d\}, C_i \cap C_j = \varnothing. \tag{3.48}$$

Criterion for cluster selection. The efficiency of the partition hinges on the strategy used to select the cluster to be split at each step. We maintain a dictionary $\mathcal{W}$ of soft-indicators associated to a partition $(A_1, ..., A_n)$ and another dictionary $\mathcal{Z}$ with partition $(B_1, ..., B_n)$. Following the idea of matching pursuit (Mallat and Zhang, 1993), a simple heuristic is to split the cluster with highest Bellman error $|TV(s) - V(s)|$. Since two dictionaries are maintained, the origin of this error will be shared between $\mathcal{W}$ and $\mathcal{Z}$, which will lead to a possibly different cluster selected in each dictionary.

We define a grid $G = (s_1, ..., s_p)$ covering $\mathcal{S}$, on which we evaluate the Bellman error. Assuming the $\langle z, Tw \rangle$ are computed exactly, after convergence of reduced value iteration, we obtain fixed points α and β such that:

$$\begin{cases} \beta = K\alpha = Z^\top TW\alpha \\ \alpha = M\beta = W^+ Z^{\top +}\beta. \end{cases} \tag{3.49}$$

Let $V = W\alpha$ and $U = Z^{\top +}\beta$, we get $V = WW^+U$ and $Z^{\top +}Z^\top TV = U$, and then the decomposition:

$$\begin{aligned} V - TV &= (V - Z^{\top +}Z^\top TV) + (Z^{\top +}Z^\top TV - TV) \\ &= (V - U) + (U - TV). \end{aligned} \tag{3.50}$$

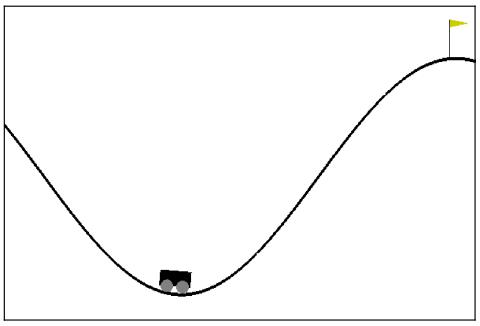
For $s \in G$, $|V(s) - TV(s)| \leq |V(s) - U(s)| + |U(s) - TV(s)|$. The first term is the difference between U and its lower projection on the span of $\mathcal{W}$, the second one between TV and its upper projection on the span of $\mathcal{Z}$. This suggests to:

- select cluster $A \ni s \in \operatorname{argmax}_{s \in G} U(s) - V(s)$ in $\mathcal{W}$,
- select cluster $B \ni s \in \operatorname{argmax}_{s \in G} U(s) - TV(s)$ in $\mathcal{Z}$.

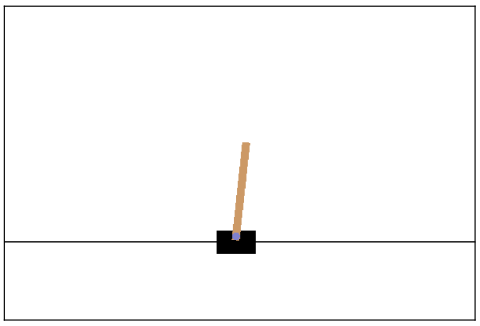
This strategy greedily targets areas of $\mathcal{S}$ where the projection errors should be reduced and locally refines the dictionaries. One could imagine other selection criteria, such as favoring areas with high value function or near the initialization of the trajectories if it is fixed. Alternative strategies include using basis functions depending on a subset of the state variables to capture local lower-dimensional dependencies (Bach, 2019). Furthermore, online methods could be applied to incorporate exploration, especially techniques based on upper-confidence bounds, as done by Bernstein and Shimkin (2008) on a similar problem.

3.7 Experiments

Setting. We test our method on two standard deterministic MDPs from the `gym` library of reinforcement learning environments (Brockman et al., 2016). Both are time-discretizations of control problems, with



(a) An initial state of the `Mountain` environment.



(b) An initial state of the `Cartpole` environment.

Figure 3.2: The two environments used in our experiments. The aim in `Mountain` is to bring the car to the top right corner, although it does not have enough throttle to climb the hill at once. In `Cartpole`, the aim is to keep the pole standing upright – an unstable position – by balancing the cart sideways.

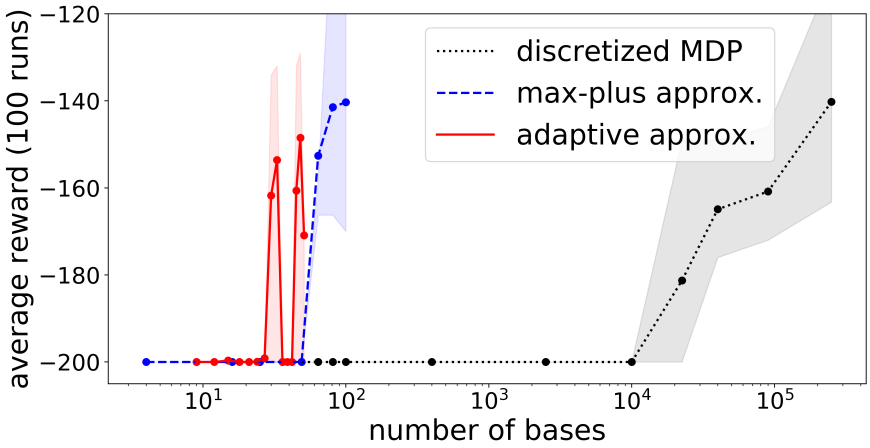


Figure 3.3: Average performance of the three approximation methods on `Mountain` as a function of the number of parameters.

state dimension 2 in `Mountain`, 4 in `Cartpole`. Sample states of the two environments are displayed in Figure 3.2. We test uniform max-plus partitions and the adaptive basis procedure, with respectively $\rho = 5$ and $\rho = 1$ for problems 1 and 2. For comparison, we also run standard value iteration on discretizations of the MDPs. To ensure differentiability, the reward function is slightly smoothed as a sigmoid function for all three methods; γ is set identical across methods ($\gamma = 0.999$ in problem 1, 0.99 in problem 2).

The optimal value function V^* being unknown, the methods cannot be evaluated by $\|V - V^*\|_\infty$. Instead we evaluate the performance of the greedy policy π with respect to V on the task. The standard performance criterion proposed by `gym` is the cumulative reward averaged on 100 consecutive runs. The randomness only comes from the initialization of the trajectories drawn from a Gaussian around equilibrium positions. The results are plotted in Figures 3.3 and 3.4. We give the mean cumulative reward in solid line, as well as the first and third quartiles in shaded colors. The x-axis represents the number of parameters of the value function, that is, either the number of basis functions in the dictionaries $\mathcal{W}$ or $\mathcal{Z}$, or the number of states in the direct discretization of the MDP.

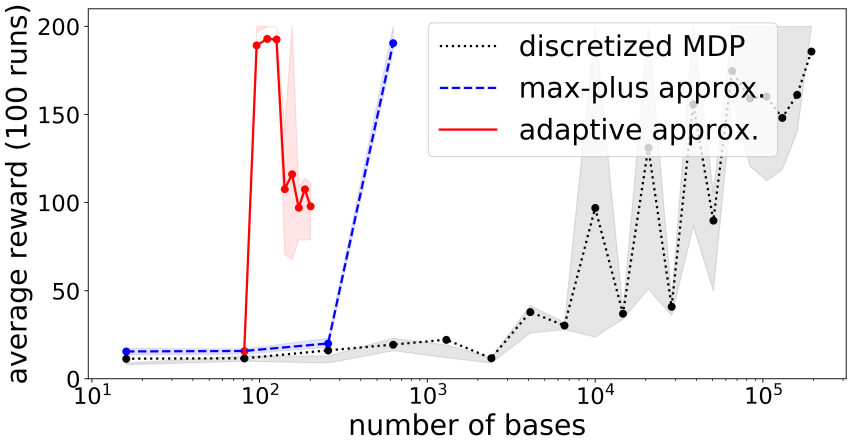


Figure 3.4: Average performance of the three approximation methods on `Cartpole` as a function of the number of parameters.

Results. Value iteration on the discretized MDP requires a very sharp discretization to get an efficient policy. While it is still achievable for such small MDPs, it is not reliable in higher dimension. The max-plus approximation computes *almost* piecewise constant value functions that lead to efficient policies for a much smaller number of parameters. On `Mountain`, the number of parameters is reduced from 10^5 to 10^2 from a direct discretization to the max-plus discretization, for similar performances of the policies. Finally, the adaptive basis selection method further improves the ratio between performance and number of parameters. It provides compact representations of V, faster to compute and leading to faster online evaluations of π during inference. However, after a few steps of greedy selection, the behavior of the adaptive approximation starts to degrade. Empirically, this corresponds to one area of the state space being selected multiple times, hence leading to small clusters concentrated in only one area. As mentioned in Section 3.6, it may be necessary to incorporate an exploration term in the selection criterion to prevent this effect.

In Figure 3.6, we plot the value functions obtained on `Mountain` with the non-adaptive (3.6a) and adaptive (3.6b) methods. As expected, the adaptive method produces a discretization whose mesh is not uniform over the state space. For reference, we also plot in Figure 3.6c the value function obtained with a very fine discretization of the continuous control problem, which is close to the true value function of the problem.

Interpretation. The fact that, on our two examples, the max-plus discretization is more efficient than a simple naive discretization – in terms of number of bases of the value function (see Figures 3.3 and 3.4) – can be interpreted as follows. Imagine a continuous MDP where the reward is uniformly zero on the left half, and one on the right half (see Figure 3.5). Suppose that the actions can move the state to the left or right by a small amount. If the discretization is coarse-scaled, *e.g.*, with two clusters, the naive discretization will result in a trivial dynamics without any movement, because the next state will collapse to the previous one. In contrast, the max-plus discretization will capture one transition between the left and right clusters, by optimizing over the state space. Of course, there is an additional computational cost to this optimization. If the discretization is fine-scaled, the benefits of the max-plus discretization is less obvious, and both methods are expected to perform similarly. Overall, the basis functions incorporate a notion of local proximity between neighboring clusters, which is absent from the naive discretized MDP.

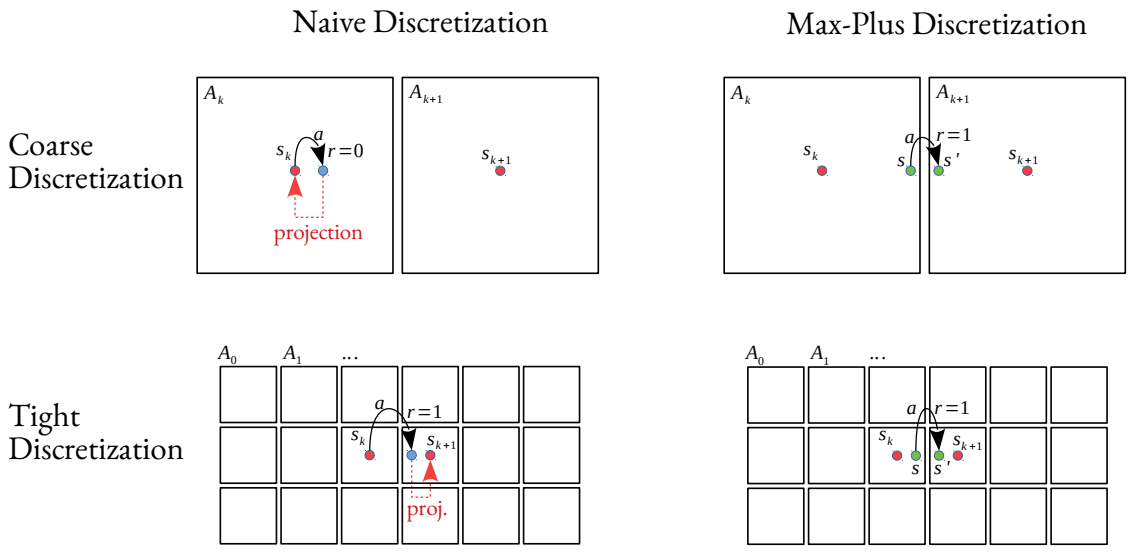
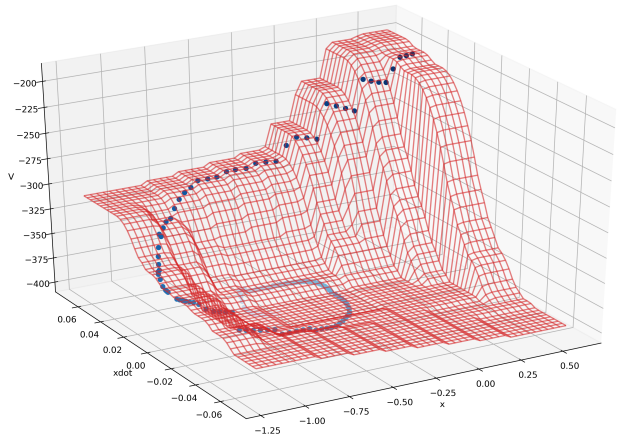


Figure 3.5: Comparison between the naive and the max-plus discretizations, at coarse and tight scales.

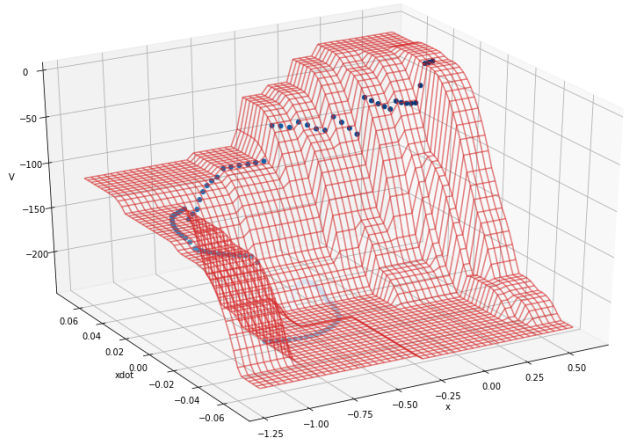
3.8 Conclusion

The max-plus linear approximation method for deterministic continuous-state MDPs with a suitable choice a basis functions provides an intuitive state-discretization. While it is still subject to the curse of dimensionality, the discretization can be adapted to a specific MDP and turns out to be effective in numerical examples. The same approach can be adapted to the Q-function for deterministic MDPs, although the potential benefits are unclear in a model-based setting. In order to make this method applicable to generic reinforcement learning problems, two extensions are needed:

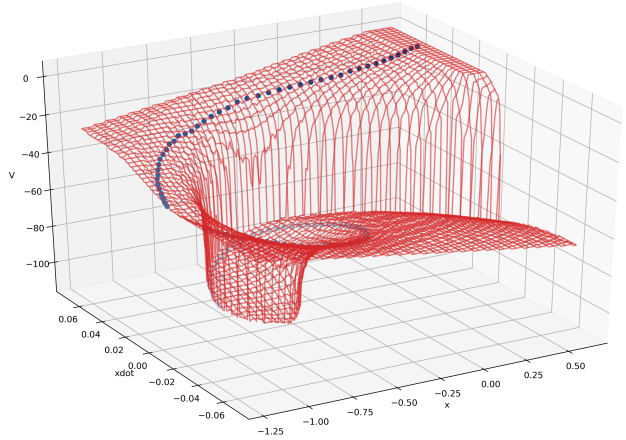
- an extension to the model-free setting, based on observations. A first model-free online approach has been recently proposed by (Gonçalves, 2021);
- an extension to stochastic MDPs. Such an extension has been proposed for stochastic control (in continuous time) by Akian and Fodjo (2017).



(a) Non-adaptive max-plus discretization.



(b) Adaptive max-plus discretization.



(c) Reference value function computed from a fine discretization.

Figure 3.6: Value functions computed on `Mountain`, along with a sample trajectory. The goal is on the upper-right corner (reaching the right hill with sufficient speed), corresponding to larger values of V.

Chapter 4

Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems

Abstract. *A linear quadratic regulator can stabilize a nonlinear dynamical system with a local feedback controller around a linearization point, while minimizing a given performance criterion. An important practical problem is to estimate the region of attraction of such a controller, that is, the region around this point where the controller is certified to be valid. This is especially important in the context of highly nonlinear dynamical systems. In this chapter, we propose two stability certificates that are fast to compute and robust when the first, or second derivatives of the system dynamics are bounded. Associated with an efficient oracle to compute these bounds, this provides a simple stability region estimation algorithm compared to classical approaches of the state of the art. We experimentally validate its application to both polynomial and non-polynomial systems of various dimensions, including standard robotic systems, for estimating regions of attraction around equilibrium points, as well as for trajectory tracking.*

This chapter is based on our work *Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems*, with Justin Carpentier and Francis Bach, published in the European Control Conference, 2021.

Contents

4.1 Introduction

Controlling a robot typically involves a global motion planning to steer the system from an initial position to a target goal, as well as some local feedback corrections to accurately track the planned trajectory. For instance, the combination of rapidly-exploring random trees (LaValle and Kuffner, 2001) and local trajectory stabilization led to a fruitful feedback motion planning algorithm named LQR-trees, proposed by Tedrake et al. (2010). In this algorithm, a locally optimal trajectory is computed between sampled points of the state space. Each trajectory is then locally stabilized with a linear quadratic regulator (LQR). The aim is to design a global controller by covering the whole state space with overlapping funnels, *i.e.*, regions of attraction (ROA) around trajectories. An important subproblem is to estimate such an ROA: a set of initial states that the controlled dynamics brings back to an equilibrium. Crucially, it must be performed efficiently, as it will be called repeatedly to cover a potentially large dimensional state space. Ideally, the estimation must be fast to compute, but not overly conservative.

A controlled dynamical system can be stabilized around an equilibrium point with an adequate closed-loop controller. It is possible to synthesize an optimal feedback controller for some stability criterion (Glover et al., 1990), but a simply available candidate is nothing more than the LQR. We show a simple example of stabilization in Figure 4.1. The stability of a region is commonly assessed with a Lyapunov function, which again can be optimized (Giesl and Hafstein, 2015; Johansen, 2000), or not. This chapter focuses on finding the largest estimate of the ROA for a given controller and a given Lyapunov function, both obtained from LQR. The gold standard technique for this problem is based on sum of squares (SoS) programming and provides high quality estimates. Yet it is limited to polynomial dynamics, and grows computationally heavy in large dimensions, hence limiting its applicability in practice, especially in the context of robotics where fast methods are needed to accurately control and stabilize the motions of the robot such as for legged locomotion (Carpentier and Mansard, 2018b).

Another stake in robotics is robustness with respect to model misspecification or uncertainties. In particular, there can be a shift between the behavior of a simulated robotic system and its physical counterpart (Singh et al., 2018). Robust ROA estimation methods (Chesi, 2004) must account for the uncertainty on the parameters of the dynamics. In particular, we focus on the case where the Jacobian or the Hessian of the dynamics is known to be bounded. This applies to robust control, but also to perfectly known dynamics that are computationally hard to handle. Bounding the Jacobian or Hessian is possible analytically for some simple low-order polynomial systems, or by sampling, taking advantage of automatic differentiation for complicated robotic systems (Giftthaler et al., 2017). Interestingly, the bounds can be computed offline, in parallel, or experimentally with a real physical system. With such information on the dynamics, our goal is to design fast, robust ROA estimation methods, practical in large state dimensions.

Our main contribution is to propose a general ROA estimation framework for non-polynomial systems, which is faster and simpler than SoS-based methods. The chapter is organized as follows. After introducing the

principle of LQR stabilization in Section 4.2, we adapt in Section 4.3 an existing robust stability certificate to systems with entry-wise uncertainty bounds on their Jacobians. In Section 4.4, we present stability certificates for systems with entry-wise bounds on their Hessians, and in Section 4.5, we propose an algorithm adapting robust certificates to systems with varying derivatives. In Section 4.6, we extend the methods to the trajectory tracking problem. In Section 4.7, we compare the robust certificates, as well as those provided by SoS programming, on numerical examples of various dimensions. Finally, in Section 4.8, we summarize the proposed framework. An implementation in Python is available online.

4.2 Preliminaries

We consider a nonlinear time-invariant control system:

$$\dot{x} = f(x, u), \tag{4.1}$$

where $x \in \mathbb{R}^d$, $u \in \mathbb{R}^m$, with $d, m \geq 1$. Assume there exists an equilibrium, without loss of generality at $(x_0, u_0) = (0, 0)$, that is $f(0, 0) = 0$, and that f is differentiable at the origin:

$$f(x, u) = \underbrace{\left.\frac{\partial f}{\partial x}\right|_{0,0}}_{A} x + \underbrace{\left.\frac{\partial f}{\partial u}\right|_{0,0}}_{B} u + o(x) + o(u). \tag{4.2}$$

We assume that the pair (A, B) is controllable. For $Q \succeq 0$, $R \succ 0$ symmetric matrices respectively of size $d \times d$ and $m \times m$, we define the infinite-horizon LQR cost (Liberzon, 2011):

$$J(x) := \int_0^{+\infty} \left(x^\top(t) Q x(t) + u^\top(t) R u(t) \right) \mathrm{d}t, \text{ with } x(0) = x. \tag{4.3}$$

The cost-minimizing controller is known to be:

$$u(x) = -R^{-1} B^\top S x =: -Kx, \tag{4.4}$$

where S is the symmetric positive definite solution of the algebraic Riccati equation (ARE), which exists because (A, B) is controllable:

$$A^\top S + SA - SBR^{-1}B^\top S = -Q. \tag{4.5}$$

Under the closed-loop controller $u(t) = -Kx(t)$, the system is autonomous with closed-loop dynamics:

$$\dot{x} = f(x, -Kx) =: g(x). \tag{4.6}$$

In addition, the optimal cost-to-go $V(x) := x^\top S x$ is used as a Lyapunov function of the nonlinear system. V is a Lyapunov function over a region $\mathcal{R} \subset \mathbb{R}^d$ around 0, if $V(0) = 0$, $V(x) > 0$ in $\mathcal{R} \setminus \{0\}$, and $\dot{V}(x) < 0$ in $\mathcal{R}$ (Slotine and Li, 1991). This certifies that the sublevel sets of V that are included in $\mathcal{R}$ belong to the ROA of the equilibrium point: every trajectory beginning in this set will asymptotically stabilize to 0. In practice, it is convenient to choose $\mathcal{R}$ as a sublevel set of V, to ensure that a trajectory starting in $\mathcal{R}$ remains in $\mathcal{R}$. In Figure 4.2, we show the ROA and the largest sublevel set included in $\mathcal{R}$ on a two-dimensional example.

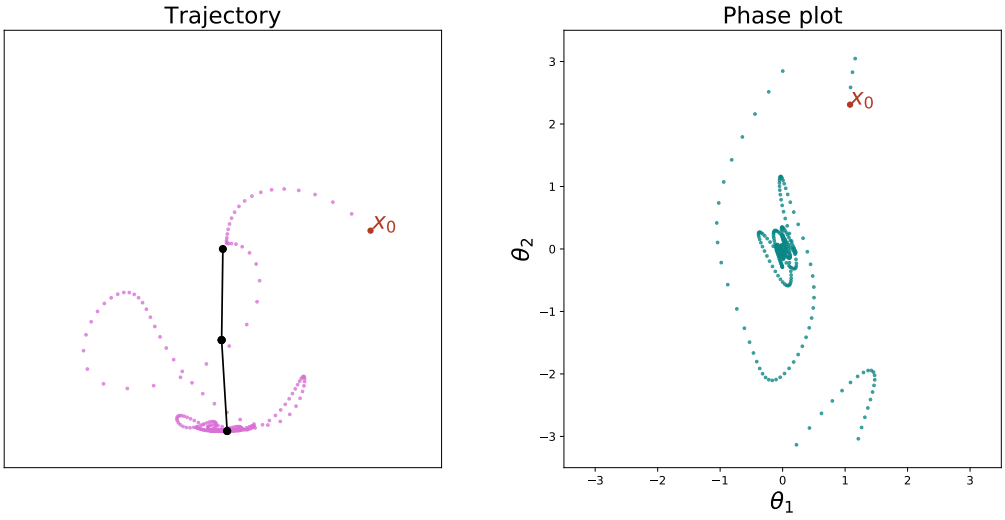


Figure 4.1: The double pendulum is stabilized from x_0 to its bottom position with an LQR controller. Note that since it is a stable equilibrium point, using a controller is not necessary, but this would be the case if we wanted to stabilize the system to the – unstable – top position. Here, we have chosen the bottom position for better readability.

If the dynamics were exactly linear, then one would have:

$$\begin{aligned}\dot{V}(x) &= \nabla V(x) \cdot g(x) = 2x^\top S g(x) = x^\top (SA + A^\top S - 2SBK)x \\ &= x^\top (-Q - SBR^{-1}B^\top S)x < 0, \quad \forall x \neq 0. \end{aligned} \tag{4.7}$$

Hence in the linear case, the ROA is the whole state space. In this work we will consider variations of this situation, and see how defects of linearity will affect this statement.

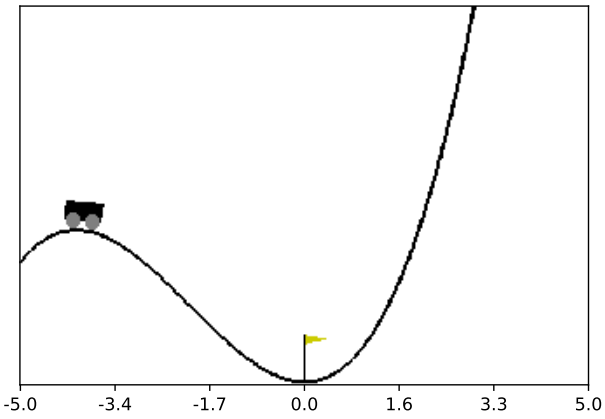
4.3 First-Order Robustness

In this section, we present a robust stability certificate that holds for the class of systems whose Jacobian matrix is bounded by a known quantity.

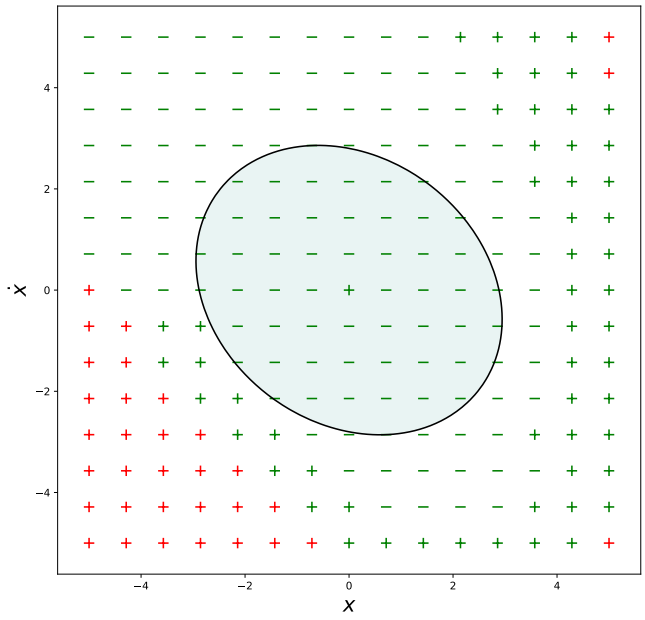
A linear differential inclusion (LDI) (Aubin and Cellina, 1984) is the following set-valued control problem:

$$\dot{x} \in \Omega x, \quad x(0) = x_0, \tag{4.8}$$

where Ω is a convex subset of $\mathbb{R}^{d\times d}$, and $\Omega x := \{Ax,\ A \in \Omega\}$. The asymptotic stability of any dynamical system belonging to the LDI can be expressed as a linear matrix inequality (LMI) for some specific choices of Ω (Boyd et al., 1994). In particular, for $\Omega = \{A_0\}$ we get a linear system, for $\Omega = \text{Conv}(A_1, ..., A_L)$ a polytopic LDI (PLDI). Let C, Δ and E be matrices of compatible dimensions and $\|.\|$ a matrix norm, then $\Omega = \{A_0 + C\Delta E \mid \|\Delta\| \leq 1\}$ represents a norm-bound LDI (NLDI); if in addition Δ must be diagonal, we get a diagonal NLDI (DNLDI).



(a) A modification of the `Mountain` environment from Brockman et al. (2016) (see also Chapter 3). Intuitively, depending on the relative strength of the car compared to gravity, stabilization to 0 will be possible except at the left-hand side.



(b) Phase plot (x and $\dot{x}$ are the initial position and velocity of the car) showing which initial points can be stabilized to 0. The points plotted in green are effectively stabilized to 0 in a simulated trajectory. Those in red are not: they escape to the left. The labels "-" denote points for which a given quadratic function V is decreasing along the whole trajectory, contrary to those labeled "+". The points labeled "-" are a subset of the green points: those for which V is a Lyapunov function. Finally, the light blue ellipse is the largest sublevel set of V where this condition holds. All methods considered in this chapter are concerned with finding this maximal ellipsoid.

Figure 4.2: Study of the stability of different initial points on a simple two-dimensional system.

The asymptotic stability of an LDI around 0, *i.e.*, for all initial conditions, the trajectories converge to 0, can be certified by a Lyapunov function of the form $V(x) = x^\top P x$. This amounts to finding:

$$P \succ 0 \text{ such that } \quad A^\top P + PA \prec 0, \ \forall A \in \Omega. \tag{4.9}$$

This problem reduces to an LMI and is classically solved by interior-point methods (Nesterov and Nemirovskii, 1994) for general choices of Ω. For example, in a PLDI, it reduces to a finitely constrained LMI, with one constraint at each of the vertices of the polytope Ω.

LDIs are used to model uncertainty in linear systems. Any differentiable dynamical system with an equilibrium at the origin and with bounded Jacobian (including the closed loop system in equation (4.6)) belongs to a suitable LDI, written as an uncertain linear system:

$$\dot{x} = A(x)x, \quad A(x) \in \Omega. \tag{4.10}$$

Ω is a convex set of matrices that accounts for nonlinearities, uncertainties or time-variations of the dynamics. In particular, Ω can bound the deviation of a nonlinear system $\dot{x} = g(x)$ from its linearization $\dot{x} = J_g(0)x$. This is similar to the problem considered by Topcu and Packard (2007), except that the perturbations lie in a closed convex set instead of a semialgebraic set.

For $i, j \in \{1, ..., d\}$, $x \in \mathbb{R}^d$, let $\delta_{ij}(x) := A(x)_{ij} - (A_0)_{ij}$, the entrywise deviations of the Jacobian of the dynamics from a given matrix A_0. Suppose we are given individual upper bounds on each deviation:

$$\forall i, j \in \{1, ..., d\}, \quad v_{ij} := \sup_{x \in \mathbb{R}^d} |\delta_{ij}(x)|. \tag{4.11}$$

Such bounds can be computed in closed form in some simple cases, or estimated by sampling, as will be discussed in Section 4.5.2. Stability is readily studied (Boyd et al., 1994) if Ω is a convex hull (PLDI) or a matrix ball (NLDI), which we now specify for our problem. Entrywise bounds can be fitted in both settings. Yet the description of the corresponding PLDI is intractable in large dimension: the number of vertices required to describe Ω scales as 2^d. This is why we opt for the NLDI.

The following description of Ω with a DNLDI has polynomial length. Let $\mathbf{1}_d := (1 \ ... \ 1)$, $\mathbf{0}_d := (0 \ ... \ 0)$,

$$\Delta := \text{Diag}\left(\frac{\delta_{11}}{v_{11}}, ..., \frac{\delta_{1d}}{v_{1d}}, \ \ ... \ \ , \frac{\delta_{d1}}{v_{d1}}, ..., \frac{\delta_{dd}}{v_{dd}}\right) \in \mathbb{R}^{d^2 \times d^2}, \tag{4.12}$$

$$C := \begin{bmatrix} \mathbf{1}_d & \mathbf{0}_d & \\ \mathbf{0}_d & \ddots & \\ & & \mathbf{1}_d \end{bmatrix} = I_d \otimes \mathbf{1}_d \in \mathbb{R}^{d \times d^2}, \tag{4.13}$$

$E := \begin{bmatrix} E_1 & ... & E_d \end{bmatrix}^\top \in \mathbb{R}^{d^2 \times d}$, with $E_i = \text{Diag}(v_{i1}, ..., v_{id})$. Hence $\|\Delta\|_2 = \sqrt{\lambda_{\max}(\Delta^\top \Delta)} = \sigma_{\max}(\Delta) \leq 1$, and the system belongs to the DNLDI defined by

$$\Omega = \{A_0 + C\Delta E \mid \|\Delta\| \leq 1, \ \Delta \text{ diagonal}\}. \tag{4.14}$$

Checking the asymptotic stability of a DNLDI is an LMI feasibility problem derived by applying the S-procedure (Boyd et al., 1994), resulting in the following proposition.

Proposition 4. *Let $\dot{x} = A(x)x$ an uncertain linear system with entrywise bounded Jacobian. A sufficient condition for its global asymptotic stability at 0 is the feasibility of the following LMI, for A_0, C, E defined as above:*

$$\textit{Find } P \succ 0 \in \mathbb{R}^{d\times d},\ \Lambda \succeq 0 \in \mathbb{R}^{d^2\times d^2} \textit{ diagonal such that:}$$
$$\begin{bmatrix} A_0^\top P + PA_0 + E^\top \Lambda E & PC \\ C^\top P & -\Lambda \end{bmatrix} \prec 0. \tag{4.15}$$

One may optimize both P and Λ to obtain a Lyapunov function, or use a fixed predefined value of P, *e.g.*, S from the LQR, to check if $V(x) = x^\top Px$ is a valid Lyapunov function.

4.4 Second-Order Robustness

Let us derive robust stability certificates like in the previous section, except that now, they hold for a class of systems whose Hessian tensor is bounded by a known quantity.

4.4.1 Condition on the Sublevel Sets

Let $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, such that each φ_k is twice continuously differentiable with bounded Hessian on a closed ball $\mathcal{B}$ centered around 0. Then, using Taylor's formula, for any $x \in \mathbb{R}^d$, there exists a symmetric matrix $H^k(x)$ such that for all $k \in \{1, ..., d\}$:

$$\varphi_k(x) = \varphi_k(0) + \nabla\varphi_k(0)^\top x + \frac{1}{2}x^\top H^k(x)x, \tag{4.16}$$
$$\text{with } H^k_{ij}(x) = 2\int_0^1 (1-t)\frac{\partial^2\varphi_k}{\partial x_i \partial x_j}(tx)\mathrm{d}t.$$

Hence $\forall i, j, k, \forall x \in \mathcal{B},\ |H^k_{ij}(x)| \leq \max_{y\in\mathcal{B}}\left|\frac{\partial^2\varphi_k}{\partial x_i\partial x_j}(y)\right|$. Note that this is also true if $\mathcal{B}$ is an ellipsoid around 0. This applies to the function g of equation (4.6):

$$g_k(x) = f_k(x, -Kx) = (A - BK)_{k\cdot}x + \frac{1}{2}x^\top H^k(x)x, \tag{4.17}$$

where $X_{k\cdot}$ denotes the k-th row-vector of a matrix X. The time derivative of the candidate Lyapunov function is:

$$\begin{aligned}\dot{V}(x) &= 2x^\top S\left((A-BK)x + \frac{1}{2}(x^\top H^k(x)x)_{k\in\{1,\dots,d\}}\right)\\ &= x^\top(-Q - SBR^{-1}B^\top S + \sum_{k=1}^d (S_{k\cdot}x)H^k(x))x.\end{aligned} \tag{4.18}$$

Let $\mathcal{B}_\rho := \{x \mid x^\top Sx \leq \rho\}$ for $\rho > 0$, a sublevel set of V. A sufficient condition for $\mathcal{B}_\rho$ to be an ROA around 0 is that $-Q - SBR^{-1}B^\top S + \sum_k (S_{k\cdot}x)H^k(x) \prec 0$ for all $x \in \mathcal{B}_\rho$. Let $M := Q + SBR^{-1}B^\top S \succ 0$, the condition is equivalent to:

$$\forall x \in \mathcal{B}_\rho, \quad \sum_{k=1}^d (S_{k\cdot}x)\tilde{H}^k(x) \prec I_d, \tag{4.19}$$

where $\tilde{H}^k(x) := M^{-1/2}H^k(x)M^{-1/2}$. We denote by $\tilde{H}(x)$ the tensor composed of the matrices $\tilde{H}^k(x)$, for $k \in \{1, ..., d\}$.

The goal is to find the largest ρ such that condition (4.19) holds, which will in turn prove that $\mathcal{B}_\rho$ is an ROA around 0. To simplify this problem, we will decouple the two dependencies in x. On the one hand, the contribution of $S_k.x$ will be bounded by two different bounds that we present below. On the other hand, the tensor $\tilde{H}(x)$ is bounded globally, independently from ρ. Of course, this is not always possible in general, and a tighter analysis of the Hessian with local bounds depending on ρ will be discussed in Section 4.5. For now, assume that we are given an oracle on the magnitude $e_i^\top \tilde{H}^k(x)e_j$ (e_i being the i-th unit vector) of $\tilde{H}(x)$ along d^2 directions for each matrix $\tilde{H}^k(x)$, of the form:

$$\forall x, \tilde{H}(x) \in \Xi := \{T \in \mathbb{R}^{d^3} | \forall i, j, k, |T^k_{ij}| \leq u^k_{ij},\ (T^k)^\top = T^k\}, \tag{4.20}$$

for some $d \times d \times d$ tensor U of nonnegative real numbers u^k_{ij}, with $(U^k)^\top = U$ for all $k \in \{1, ..., d\}$.

A relaxation of condition (4.19) is then:

$$\sup_{x^\top Sx \leq \rho} \sup_{T \in \Xi} \lambda_{\max}\left(\sum_{k=1}^{d}(S_k.x)T^k\right) < 1. \tag{4.21}$$

With a simple change of variable and rescaling, the largest ρ fulfilling the above condition is then given by:

$$\rho = \frac{1}{\lambda^2}, \text{ where } \lambda := \sup_{\|y\|_2 \leq 1} \sup_{T \in \Xi} \lambda_{\max}\left(\sum_{k=1}^{d}(S_{k.}^{1/2}y)T^k\right). \tag{4.22}$$

4.4.2 Two Upper Bounds on λ

The first bound is based on the following fact:

$$\sup_{\|y\|_2 \leq 1} \sup_{T \in \Xi} \|\sum_{k=1}^{d}(S_{k.}^{1/2}y)T^k\|_2 \leq \sup_{\|y\|_2 \leq 1} \sum_{k=1}^{d} |S_{k.}^{1/2}y| \sup_{T^k \in \Xi^k} \|T^k\|_2, \tag{4.23}$$

where Ξ^k is the projection of Ξ onto its k-th coordinate subspace.

Let Z be a matrix with rows $Z_{k.} := \left(\sup_{T \in \Xi^k} \|T^k\|_2\right) S_{k.}^{1/2}$, then

$$\lambda \leq \sup_{\|y\|_2 \leq 1} \|Zy\|_1 \leq \sqrt{d} \sup_{\|y\|_2 \leq 1} \|Zy\|_2 = \sqrt{d}\|Z\|_2, \tag{4.24}$$

where $\|.\|_2$ denotes both the Euclidean norm and the corresponding matrix induced norm. With equation (4.22), this guarantees that $\mathcal{B}_{\rho_b}$ is an ROA for

$$\rho_b := \frac{1}{d\|DS^{1/2}\|_2^2}, \text{ with } D = \text{Diag}\Big(\big(\sup_{T^k \in \Xi^k} \|T^k\|_2\big)_k\Big). \tag{4.25}$$

The following lemma explains how to compute the entries of D.

Lemma 1. *Let V be a nonnegative symmetric $d \times d$ matrix with entries (v_{ij}). Let ξ be the set of symmetric $d \times d$ matrices A such that for all $i, j \in \{1, ..., d\}$, $|A_{ij}| \leq v_{ij}$. Then:*

$$\max_{A \in \xi} \|A\|_2 = \|V\|_2. \tag{4.26}$$

Proof. Since ξ is centered around 0, we only look for the largest eigenvalue:

$$\sup_{\|x\|_2\leq 1,\ A\in\xi} x^\top Ax = \sup_{\|x\|_2\leq 1} \sup_{A\in\xi} \sum_i a_{ii}x_i^2 + 2\sum_{i<j} a_{ij}x_ix_j. \tag{4.27}$$

Maximizing with respect to A, we get $a_{ii} = v_{ii}$, and

$$\forall i<j, \quad a_{ij} = \begin{cases} v_{ij} & \text{if } x_ix_j \geq 0 \\ -v_{ij} & \text{else.} \end{cases} \tag{4.28}$$

And then, $x^\top Ax = \sum_i v_{ii}x_i^2 + 2\sum_{i<j} v_{ij}|x_ix_j| = |x|^\top V|x|$. The full problem becomes:

$$\sup_{\|x\|_2\leq 1,\ x\succeq 0} x^\top Vx. \tag{4.29}$$

The Perron-Frobenius theorem ensures that for any nonnegative square matrix, there exists a nonnegative real eigenvalue with at least one nonnegative eigenvector. Any other eigenvalue's modulus is smaller than this eigenvalue. V being symmetric, all its eigenvalues are real, hence the result. □

Remark. If the bounds on the entries of A are not centered around 0, it is possible to write $A = A' + \bar{A}$, where the entries of $\bar{A}$ are symmetrically bounded, and use $\|A\|_2 \leq \|A'\|_2 + \|\bar{A}\|_2$.

The lemma leads to another bound on λ. If $T \in \Xi$:

$$\forall i,j,k,\ |\sum_k (S_{k\cdot}^{1/2}y)T_{ij}^k| \leq \sum_k |S_{k\cdot}^{1/2}y|u_{ij}^k \leq \sum_k \|S^{1/2\top}_{k\cdot}\| \cdot \|y\|u_{ij}^k. \tag{4.30}$$

Applying the previous result to the matrix whose entries are the middle term in the inequality above yields:

$$\lambda \leq \lambda_{\max}\left(\sum_k \sqrt{S_{k\cdot}S^{-1}S_{k\cdot}^\top}U^k\right) = \lambda_{\max}\left(\sum_k \sqrt{S_{kk}}U^k\right) =: \lambda_a. \tag{4.31}$$

The following theorem states the two stability certificates derived above.

Theorem 1. *Consider the control system and the matrices S, K and M defined in Sections 4.2 and 4.4. Assume that the closed-loop system $x \mapsto g(x) = f(x, -Kx)$ is twice continuously differentiable and the following condition holds:*

$$\forall x \in \mathbb{R}^d, \forall i,j,k \in \{1,...,d\}, \left|\left[M^{-1/2}\nabla^2 g_k(x) M^{-1/2}\right]_{ij}\right| \leq u_{ij}^k, \tag{4.32}$$

where for each k, the matrix U^k with entries $(u_{ij}^k)_{ij}$ is symmetric and nonnegative. Then $\mathcal{B}_{\rho_a}$ and $\mathcal{B}_{\rho_b}$ are two ROAs of the closed-loop system, for $\rho_a = 1/\lambda_a^2$, $\rho_b = 1/\lambda_b^2$ and

$$\lambda_a = \lambda_{\max}\left(\sum_k \sqrt{S_{kk}}U^k\right), \tag{4.33}$$

$$\lambda_b = \sqrt{d}\|DS^{1/2}\|_2, \text{ with } D = \text{Diag}\left(\|U^k\|_2\right)_k. \tag{4.34}$$

This theorem provides certified ROAs from a given Lyapunov function $V(x) = x^\top Sx$, which comes from the LQR. The elliptical shape of the ROAs is fixed, as opposed to Proposition 4, where the Lyapunov function $x^\top Px$ can be optimized. In the rest of the chapter, we will consider P to be fixed in the LMI (4.15). There does not seem to be a straightforward extension of our second-order certificates with an optimized Lyapunov function.

4.5 Iterative Algorithm

4.5.1 Stability Certificates

The bounds introduced in Sections 4.3 and 4.4 readily give robust stability certificates that hold for a whole class of dynamics with suitably bounded derivatives. It is also possible to apply the same methods to a single known dynamics, if its derivatives can be bounded efficiently. In general the bounds on the derivatives depend on where they are computed: the larger the region, the larger the bounds. But such bounds must be computed on a sufficiently large region containing the sublevel set where the system stability is asserted.

With the notations of the two previous sections, and given equations (4.15,4.25,4.31), we define three stability certificates:

$$\mathcal{C}_1 : (S, \rho_{up}, \Omega) \to \rho_{up} \mathbf{1}_{\text{LMI (4.15) is feasible}} \tag{4.35}$$

$$\mathcal{C}_2^{a,b} : (S, \rho_{up}, \Xi) \to \min(\rho_{a,b}, \rho_{up}), \tag{4.36}$$

meaning that $\{x^\top S x \leq \mathcal{C}_1(S, \rho_{up}, \Omega)\}$ is an ROA if the derivatives of the dynamics are bounded by Ω (resp. Ξ) and if ρ_{up} is an upper bound on ρ used to compute Ω (resp. Ξ).

4.5.2 Oracle on the Derivatives

Suppose that we have an oracle $\mathcal{O}$ computing, on a domain $\mathcal{D}$, a bound $\mathcal{O}(\mathcal{D})$ on the derivatives, corresponding to sets Ω or Ξ above. Our methods compute $\rho = \mathcal{C}(S, \rho_{up}, \mathcal{O}(\mathcal{D}))$. Then $\mathcal{B}_\rho$ is an ROA if the whole trajectory to 0 stays inside $\mathcal{D}$ (else the assumptions on the derivatives would be violated). A simple way to ensure that is to choose $\mathcal{D}$ as a sublevel set of V containing $\mathcal{B}_\rho$, *i.e.*, $\mathcal{B}_{\rho_{up}}$ for $\rho_{up} \geq \rho$.

For a quadratic dynamical system, each entry of the Hessian is constant and each entry of the Jacobian is an affine function. For any $c \in \mathbb{R}^d$, the supremum of a linear function $x \mapsto c^\top x$ on an ellipsoid can be computed in closed-form, as follows:

$$\sup_{x^\top S x \leq \rho_{up}} c^\top x = \sqrt{\rho_{up}}\, \|S^{-1/2} c\|_2. \tag{4.37}$$

For a third-order polynomial system, the entries of its Hessian are affine, hence the previous formula can be applied, and those of the Jacobian are polynomials of degree two. The following formula gives an exact upper bound for a quadratic monomial over an ellipsoid and naturally extends to polynomials of degree two after an affine change of variable:

$$\sup_{x^\top S x \leq \rho_{up}} x^\top J x = \rho_{up}\, \lambda_{\max}(S^{-1/2} J S^{-1/2}). \tag{4.38}$$

In large dimension, manually identifying each coefficient of the derivatives of a second or third order polynomial dynamics might be tedious: the Hessian tensor indeed contains d^3 entries. One solution is to define the polynomial dynamics f with symbolic expressions and to obtain the derivatives with a computed algebra system. Another one is to sample derivatives at a few but different points with automatic differentiation (Paszke et al., 2019), to fit a low order polynomial model, and then to maximize it in closed form.

For generic dynamics, one can sample derivatives, *e.g.*, by automatic differentiation, using analytical derivatives (Carpentier and Mansard, 2018a) for rigid body dynamics, or from direct physical measurements on

the system. Of course, bounding the samples only provides lower-estimates of the oracle, possibly resulting in over-optimistic stability certificates. Hence extra caution must be taken to ensure sufficient precision of the oracles. In particular, samples can be collected offline or in parallel in order to mitigate the computation times. Besides, maximization by sampling suffers from the curse of dimensionality, yet efficiency improvements can be expected with Bayesian optimization (Mockus, 2012) or other global optimization tools.

4.5.3 Algorithm

A simple ROA estimation algorithm (see Algorithm 2) consists in iteratively bounding the derivatives and producing stability certificates, *i.e.*, alternating calls of $\mathcal{O}$ and $\mathcal{C}$. ρ_0 is an initial upper bound on the size of the ROA. Each step of the loop provides a certificate that $\mathcal{B}_\rho$ is an ROA, and this region grows at each iteration, the sequence of ρs being nondecreasing. The number of iterations before the algorithm stops depends on both the initial guess ρ_0 and the step size η. In our experiments, we typically require from 10 up to 20 iterations.

Algorithm 2 Adaptive stability certificates

Input: S, $\mathcal{C}()$, $\mathcal{O}()$, $\rho_0 > 0$, $\eta \in (0, 1)$
Output: An ROA certificate on $\{x \mid x^\top S x \leq \rho\}$

$\rho_{up} \leftarrow \rho_0$
repeat
  $U \leftarrow \mathcal{O}(\mathcal{B}_{\rho_{up}})$
  $\rho \leftarrow \mathcal{C}(S, \rho_{up}, U)$
  $\rho_{up} \leftarrow \eta \rho_{up}$
until $\rho \geq \rho_{up}$
return ρ

4.6 Trajectory Tracking

The certificates and the algorithm presented in the previous sections are applied around the equilibrium point of a dynamical system. They can be extended to the more general problem of trajectory tracking, as described hereafter.

Let $(x_0(t), u_0(t))$, for $t \in [0, t_f]$ be a reference trajectory with final state $x_f := x_0(t_f)$. For a nearby trajectory $(x(t), u(t))$, let $\bar{x}(t) := x(t) - x_0(t)$, $\bar{u}(t) := u(t) - u_0(t)$. The linearized dynamics reads:

$$\dot{\bar{x}}(t) = A(t)\bar{x}(t) + B(t)\bar{u}(t) + o(\bar{x}(t)) + o(\bar{u}(t)). \tag{4.39}$$

Let $\mathcal{B}_f$ a target region $\{x \mid (x - x_f)^\top S_f (x - x_f) \leq 1\}$, for some $S_f \succeq 0$. We define the finite-horizon LQR problem (Liberzon, 2011) with the following tracking cost:

$$\int_0^{t_f} (\bar{x}(t)^\top Q \bar{x}(t) + \bar{u}(t)^\top R \bar{u}(t)) \mathrm{d}t + \bar{x}^\top(t_f) S_f \bar{x}(t_f). \tag{4.40}$$

For $t \in [0, t_f]$, the optimal cost-to-go is $V(x, t) = \bar{x}^\top S(t) \bar{x}$, $S(t)$ being the solution of the Riccati differential algebraic equation (RDE):

$$\dot{S} = -Q + SBR^{-1}B^\top S - SA - A^\top S, \quad S(t_f) = S_f, \tag{4.41}$$

with controller $\bar{u}(t) = -K(t)\bar{x}(t) := -R^{-1}B^\top(t)S(t)\bar{x}(t)$.

We want to estimate the time-varying region (also called "funnel" by Tedrake et al. (2010))

$$\mathcal{B}(t) := \{x \mid F(x,t) \in \mathcal{B}_f\}\,, \tag{4.42}$$

where $F(x,t)$ is the integrated closed-loop dynamics with control $u(.)$ from t to t_f. In particular, we have $\mathcal{B}(t_f) = \mathcal{B}_f$. $\mathcal{B}(t)$ is a region where applying $u(t) = u_0(t) + \bar{u}(t)$ will make the trajectory reach $\mathcal{B}(t_f)$ after time t_f. If in addition $\mathcal{B}(t_f)$ is included in an ROA around 0, the trajectory will finally reach 0 in finite time.

We consider regions $\mathcal{B}(t) := \{x \mid 0 \leq V(x,t) \leq \rho(t)\}$. A sufficient condition for $\mathcal{B}(t)$ to be a funnel is (Tobenkin et al., 2011):

$$V(x,t) \geq 0,\ \forall x \in \mathcal{B}(t) \text{ and } \dot{V}(x,t) \leq \dot{\rho}(t),\ \forall x \in \partial\mathcal{B}(t). \tag{4.43}$$

We drop some occurrences of the time variable t to simplify the notations. $S(t)$ being a positive definite matrix (Liberzon, 2011) for any $t \in [0, t_f]$, the first condition holds, the second one is:

$$\dot{V}(x,t) = 2\bar{x}^\top S\dot{\bar{x}} + \bar{x}^\top \dot{S}\bar{x} \leq \dot{\rho},\ \forall x \in \left\{x \mid \bar{x}^\top S\bar{x} = \rho\right\}. \tag{4.44}$$

Assume that the closed-loop system is an LDI $\dot{\bar{x}} = \tilde{A}(t,x)\bar{x}$ in $\{x \mid \bar{x}^\top S\bar{x} = \rho\}$, with $\tilde{A} \in \Omega(\rho)$, for a given set $\Omega(\rho)$. Then, a sufficient condition is:

$$\forall \tilde{A} \in \Omega(\rho), \quad \tilde{A}^\top S + S\tilde{A} + \dot{S} - \frac{\dot{\rho}}{\rho}S \preceq 0. \tag{4.45}$$

This can be fit into the LDI framework presented in Section 4.3, just by shifting the set $\Omega(\rho)$ to the set

$$\tilde{\Omega}(\rho,\dot{\rho}) := \{\tilde{A} + \frac{1}{2}S^{-1}\dot{S} - \frac{1}{2}\frac{\dot{\rho}}{\rho}I_d \mid \tilde{A} \in \Omega(\rho)\}. \tag{4.46}$$

We now deal with the case where the closed-loop system is known up to order two in $\{x \mid \bar{x}^\top S\bar{x} = \rho\}$, say $\dot{\bar{x}} = (A - BK)\bar{x} + \frac{1}{2}\bar{x}^\top H(t,x)\bar{x}$, with $H(t,x) \in \Xi(\rho)$. Using that $S(.)$ is a solution of equation (4.41), we obtain the following sufficient condition: $\forall\, y$ such that $\|y\|_2 = 1,\ \forall\, H \in \Xi(\rho)$,

$$-Q - SBR^{-1}B^\top S - \frac{\dot{\rho}}{\rho}S + \sqrt{\rho}\sum_{k=1}^{d}(S^{1/2}_{k.}y)H^k \preceq 0\,. \tag{4.47}$$

If $N(\rho,\dot{\rho}) = Q + SBR^{-1}B^\top S + \frac{\dot{\rho}}{\rho}S \succ 0$, let $\tilde{\Xi}(\rho,\dot{\rho})$ the shifted set $\{N^{-1/2}HN^{-1/2} \mid H \in \Xi(\rho)\}$, we must check that:

$$\forall\, y \text{ s.t. } \|y\|_2 = 1,\ \forall\, \tilde{H} \in \tilde{\Xi}(\rho,\dot{\rho}),\ \sqrt{\rho}\sum_{k=1}^{d}(S^{1/2}_{k.}y)\tilde{H}^k \preceq I_d\,. \tag{4.48}$$

Under such conditions, $\rho(.)$ is built backwards in time through backward integration, starting from $\rho(t_f) = 1$. At each time step, given ρ, a greedy strategy is to choose $\dot{\rho}$ as the smallest possible value such that the sufficient condition is enforced. Since $\rho(.)$ is computed backwards, this maximizes $\rho(t - \mathrm{d}t)$, hence locally the funnel's volume. Also, for both first and second order cases, the sufficient condition is monotonically more restrictive as $\dot{\rho}$ decreases. A simple algorithm is to start with a large positive $\dot{\rho}$, compute the set $\tilde{\Omega}(\rho,\dot{\rho})$ or $\tilde{\Xi}(\rho,\dot{\rho})$, check that the sufficient condition holds, and progressively decrease $\dot{\rho}$ until it no longer does (possibly with $\dot{\rho} < 0$ if $N \succ 0$ is still enforced, when applicable).

4.7 Numerical Experiments

4.7.1 Definition of the Systems and Implementation Details

The code to reproduce the experiments is available online[1]. The first two systems, an electrical oscillator and a floating satellite with commanded torques, are taken from the `Matlab` material of Tobenkin et al. (2011). The third one is an underactuated double pendulum, with the actuated joint between the two arms (also called "acrobot" by Sutton (1996)). The last one corresponds to the UR5 robotic arm from *Universal Robots*[2], with 6 actuated joints. The dynamics of these dynamical systems are described hereafter.

Vanderpol. $d = 2$, $m = 0$ (unactuated), $x_0 = \mathbf{0}_2^\top$, $Q = I_2$. The dynamics is a polynomial of degree 3:

$$\forall x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x) = (-x_2,\ x_1 + x_2(x_1^2 - 1))^\top.$$

Satellite. $d = 6$, $m = 3$, $(x_0, u_0) = (\mathbf{0}_6^\top, \mathbf{0}_3^\top)$, $Q = I_6$, $R = 10 \times I_3$, the dynamics is a polynomial of degree 3. Let $J = \mathrm{Diag}(5, 3, 2)$. For $x = (\omega^\top, \sigma^\top)^\top \in \mathbb{R}^6$, with $\omega, \sigma \in \mathbb{R}^3$, $f(x, u) = (\dot{\omega}^\top, \dot{\sigma}^\top)^\top$,

$$\begin{aligned}
\dot{\omega} &= J^{-1}(u - \omega \times J\omega) \\
\dot{\sigma} &= \frac{1}{4}\left((1 - \|\sigma\|^2)I_3 + 2\sigma\sigma^\top - 2\begin{bmatrix} 0 & \sigma_3 & \sigma_2 \\ \sigma_3 & 0 & \sigma_1 \\ \sigma_2 & \sigma_1 & 0 \end{bmatrix}\right)\omega.
\end{aligned}$$

Pendulum. $d = 4$, $m = 1$, $u_0 = 0$, $x_0 = \mathbf{0}_4^\top$ (for the bottom position, see Figure 4.1) or $x_0 = (\pi, \pi, 0, 0)^\top$ (for the top position), $Q = I_4$, $R = 1$. Let $g = 9.8$, $\ell = 0.5$ and $\mu = 1$. For $x = (\theta_1, \theta_2, p_1, p_2)^\top$, $f(x, u)$ is defined by:

$$\begin{aligned}
\dot{\theta}_1 &= \frac{6}{\mu\ell^2}\frac{2p_1 - 3\cos(\theta_1 - \theta_2)p_2}{16 - 9\cos^2(\theta_1 - \theta_2)} \\
\dot{\theta}_2 &= \frac{6}{\mu\ell^2}\frac{8p_2 - 3\cos(\theta_1 - \theta_2)p_1}{16 - 9\cos^2(\theta_1 - \theta_2)} \\
\dot{p}_1 &= -\frac{\mu\ell^2}{2}\left(\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + \frac{3g}{\ell}\sin\theta_1\right) \\
\dot{p}_2 &= -\frac{\mu\ell^2}{2}\left(-\dot{\theta}_1\dot{\theta}_2\sin(\theta_1 - \theta_2) + \frac{g}{\ell}\sin\theta_1\right) + u.
\end{aligned}$$

Robotic arm. Here $d = 12$, $m = 6$, $x_0 = (q_0^\top, \mathbf{0}_6)^\top$, with q_0 the following initial configuration, displayed in Figure 4.3:

$$q_0 = (0, -\pi/5, -3\pi/5, 0, 0, 0),$$

$Q = I_{12}$, $R = I_6$. u_0 is such that $f(x_0, u_0) = 0$ and is computed by the recursive Newton-Euler algorithm (RNEA) implemented in the C++ library `Pinocchio` (Carpentier et al., 2019). The forward dynamics $f(x, u)$ is computed *via* the articulated body algorithm (ABA). We refer to Section 2.1 in Chapter 2 for more details.

[1] `www.github.com/eloiseberthier/Fast-Robust-ROA`
[2] `www.universal-robots.com/products/ur5-robot`

Table 4.1: Radius and volume of the certified ROA for the different methods, relative to the values obtained by sampling for reference.

Dynamics	$\mathcal{C}_1$		$\mathcal{C}_2^a$		$\mathcal{C}_2^b$		**SoS**		**sampling**	
	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s
Vanderpol	0.20	0.20	0.14	0.14	0.10	0.10	**1**	**1**	1	1
Satellite	2.9×10^{-2}	2.6×10^{-5}	9.3×10^{-2}	9.4×10^{-4}	7.9×10^{-2}	5.7×10^{-4}	**0.93**	**0.82**	1	1
Pend. (bot.)	3.2×10^{-2}	1.1×10^{-3}	3.5×10^{-2}	1.2×10^{-3}	$\mathbf{4.2\times10^{-2}}$	$\mathbf{1.9\times10^{-3}}$	1.4×10^{-2}	2.0×10^{-4}	1	1
Pend. (top)	5.1×10^{-3}	2.6×10^{-5}	4.5×10^{-2}	2.0×10^{-3}	$\mathbf{4.7\times10^{-2}}$	$\mathbf{2.2\times10^{-3}}$	N.A.	N.A.	1	1
Robot	2.4×10^{-3}	1.8×10^{-16}	7.1×10^{-3}	1.5×10^{-13}	$\mathbf{1.5\times10^{-2}}$	$\mathbf{1.2\times10^{-11}}$	N.A.	N.A.	1	1

Table 4.2: CPU time (s) per iteration, except for SoS (total time).

Dynamics	$\mathcal{O}+\mathcal{C}_1$	$\mathcal{O}+\mathcal{C}_2^a$	$\mathcal{O}+\mathcal{C}_2^b$	**SoS**
Vanderpol	1.8×10^{-3}	1.1×10^{-4}	1.6×10^{-4}	0.05
Satellite	1.2	0.17	0.17	32
Pend. (bot.)	2.3	15	15	132
Robot	2.3	32	33	N.A.

The software used for the SoS-based certificates is adapted from the `Matlab` material of Tobenkin et al. (2011). The oracles on the derivatives are computed either in closed form, for *Vanderpol* and the Hessian of *Satellite*, using formulas (4.37) and (4.38), or by sampling p derivatives. Using automatic differentiation in `PyTorch` (Paszke et al., 2019), we sample $p = 10^4$ Jacobians for *Satellite*, $p = 10^3$ Jacobians and Hessians for *Pendulum*. For *Robot*, $p = 5 \times 10^4$ and the Jacobians of the dynamics are computed analytically (Carpentier and Mansard, 2018a), and we use finite differences on the first partial derivatives to approximate the Hessians. It is important to notice at this stage that more advanced methods to efficiently compute these Hessians could improve the whole computation time of our methods, for instance by code-generating the second-order derivatives computed by automatic differentiation. Yet, the proposed solution already provides competitive timings.

4.7.2 Results

The performances of the certificates are compared in Table 4.1, both in terms of radius of $\mathcal{B}_\rho$ and volume $v \propto \rho^{d/2}/\sqrt{|S|}$, the latter exacerbating differences in large dimensions. The volume, divided by the volume of the state space, is roughly the inverse of the number of ROAs that would have covered it. All the values in the table are divided by the ground truth ρ_s, the maximal ρ such that $\forall x^\top S x \leq \rho$, $\dot{V}(x) < 0$, estimated by sampling a very large number of points. Apart from SoS on the first two problems, all methods are very far from estimating the true maximal ROA.

For the SoS method on *Pendulum*, because the dynamics is non polynomial, we substitute the odd function f by its Taylor expansion around the equilibrium, truncated at order $n = 7$. The result is sensitive to the order: for $n = 2$, f is linear hence $\rho = +\infty$, whereas ρ decreases for higher orders. It is unclear which one to choose, and the results are no longer certified. At the top position, an utterly unstable position, SoS fails to provide a positive ρ, regardless of $n \geq 3$.

Table 4.2 reports the corresponding CPU running times on a standard laptop. The code, in Python, is not

optimized, except the SoS method and the LMI solver for $\mathcal{C}_1$ which are in `Matlab`. Our methods are much lighter than SoS, yet one must keep in mind that Algorithm 2 typically calls the oracle and the certificate 10 times. Nonetheless, this allows to tackle systems of larger dimensions, like *Robot*. If the oracle uses sampling, this dominates the running time. Figure 4.4 compares the running times of bounding the derivatives for $\mathcal{C}_1$ and $\mathcal{C}_2^a$, depending on the number of samples p, on *Satellite*. At fixed p, it is of course longer to sample Hessians than Jacobians. The sampling oracle overestimates ρ, but this tends to stabilize for reasonable values of p, as seen for ρ_2^a which can also be computed using a closed-form oracle.

We also experiment trajectory tracking of a given reference trajectory of *Vanderpol*, with $x_f = (-1, -1)^\top$, $t_f = 2$. The target region $\mathcal{B}_f = \{x \mid \bar{x}^\top S_f \bar{x} \leq 1\}$ is the largest ellipsoid included in $\mathcal{R}$, an ROA around 0 computed by SoS. In Figure 4.6, the state-space is in green, $\mathcal{R}$ in light gray and $\mathcal{B}_f$ in red. The funnel $\mathcal{B}(t)$, in gray, is computed backwards, with one or two iterations of the SoS-based algorithm of Tobenkin et al. (2011), and with the methods of Section 4.6. Figure 4.5 shows our certificates lead to competitive values of $\rho(t)$, with faster computations.

4.8 Implementation Summary

The complete ROA estimation and trajectory tracking frameworks are summarized respectively in Figure 4.7 and 4.8. Each building block used in the diagrams is detailed below.

The first one computes the LQR as in Section 4.2.

• **Static LQR**: $(Q, R, x_0, u_0, f) \to (S, K)$.

$A = \frac{\partial f}{\partial x}(x_0, u_0)$, $B = \frac{\partial f}{\partial u}(x_0, u_0)$, S is the positive definite solution of $A^\top S + SA - SBR^{-1}B^\top S = -Q$, and $K = R^{-1}B^\top S$.

The following block computes the LQR for one time step of trajectory tracking and is detailed in Section 4.6.

• **Dynamic LQR**:

$$(S(t+\tau), Q, R, x_0(t), u_0(t), f) \to (S(t), K(t)).$$

Let $\bar{S} = S(t+\tau)$, then $S(t) = \bar{S} - \tau\dot{S}$, with $A = \frac{\partial f}{\partial x}(x_0(t), u_0(t))$, $B = \frac{\partial f}{\partial u}(x_0(t), u_0(t))$,

$$\dot{S} = -Q - \bar{S}A - A^\top \bar{S} + \bar{S}BR^{-1}B^\top \bar{S}, \text{ and } K(t) = R^{-1}B^\top S(t).$$

The next two blocks compute bounds on the derivatives of the dynamics. They can be implemented arbitrarily.

• **First-order oracle**: $(\rho_{up}, S, K, x_0, u_0, f) \to (A_0, V)$.

$$V_{ij} := \sup_{x^\top S x \leq \rho_{up}} |J_{ij}(x) - (A_0)_{ij}|,$$

where J is the Jacobian of $x \mapsto f(x_0 + x, u_0 - Kx)$. A default choice for A_0 is $J(0) = A - BK$.

• **Second-order oracle**: $(\rho_{up}, S, K, M, x_0, u_0, f) \to U$.

$$U_{ij}^k := \sup_{x^\top S x \leq \rho_{up}} \left[M^{-1/2} H^k(x) M^{-1/2}\right]_{ij},$$

where H is the Hessian of $x \mapsto f(x_0 + x, u_0 - Kx)$.

The next two blocks compute stability certificates, as detailed in Sections 4.3 and 4.4.

• **First-order certificate**:

$$(\rho_{up}, S, A_0, V) \to \rho_1 = \rho_{up} \mathbf{1}_{\text{LMI is feasible}}.$$

Let C, E defined as in section 4.3. The LMI feasibility problem is to find $\Lambda \succeq 0 \in \mathbb{R}^{d^2 \times d^2}$ diagonal such that:

$$\begin{bmatrix} A_0^\top S + S A_0 + E^\top \Lambda E & SC \\ C^\top S & -\Lambda \end{bmatrix} \prec 0.$$

• **Second-order certificate**: $(\rho_{up}, S, U) \to \rho_{a,b} = \frac{1}{\lambda_{a,b}^2}$.

$$\lambda_a = \lambda_{\max} \left(\sum_k \sqrt{S_{k.} S^{-1} S_{k.}^\top} U^k \right),$$

$$\lambda_b = \sqrt{d} \| D S^{-1/2} \|_2, \text{ with } D = \text{Diag} \left(\| U^k \|_2 \right)_k.$$

The last two blocks are used in Section 4.6.

• **First-order shift**: $(A_0, \dot{\rho}, \rho(t+\tau), S(t), \dot{S}(t)) \to \tilde{A}_0$.

$\tilde{A}_0 = A_0 + \frac{1}{2} S^{-1} \dot{S} - \frac{1}{2} \frac{\dot{\rho}}{\rho} I_d$, where $\dot{S}$ is given by the RDE (equation (4.41)).

• **Second-order shift**:

$$(M, \dot{\rho}, \rho(t+\tau), S(t)) \to \tilde{M} = M + \frac{\dot{\rho}}{\rho} S.$$

4.9 Conclusion

The stability certificates presented in this chapter are both fast to compute, and robust over a class of bounded-derivatives dynamics. They readily extend to the trajectory tracking problem, with a linear complexity in the number of time steps. Such certificates can be easily implemented and enable handling non-polynomial, large dimensional control systems that were previously out of reach. The complexity is transferred from the certificate to a derivative-bounding oracle, which can be estimated efficiently in some cases, including rigid body dynamic systems in robotics. The certificates for trajectory tracking can in turn be integrated into the LQR-trees framework for global motion planning. They are more conservative than competing methods, yet faster, hence repeating calls to these certificates around numerous different trajectories, as done in the LQR-trees algorithm, could be more efficient overall. Providing empirical evidence or counter-evidence for this trade-off phenomenon in real-world control systems would be an interesting avenue for future research.

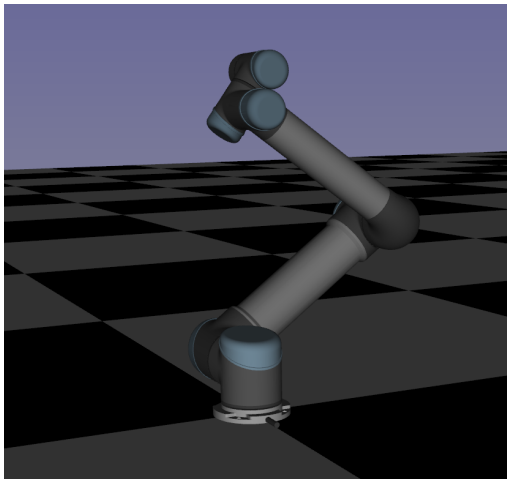


Figure 4.3: Configuration q_0 to be stabilized for the UR5 robotic arm.

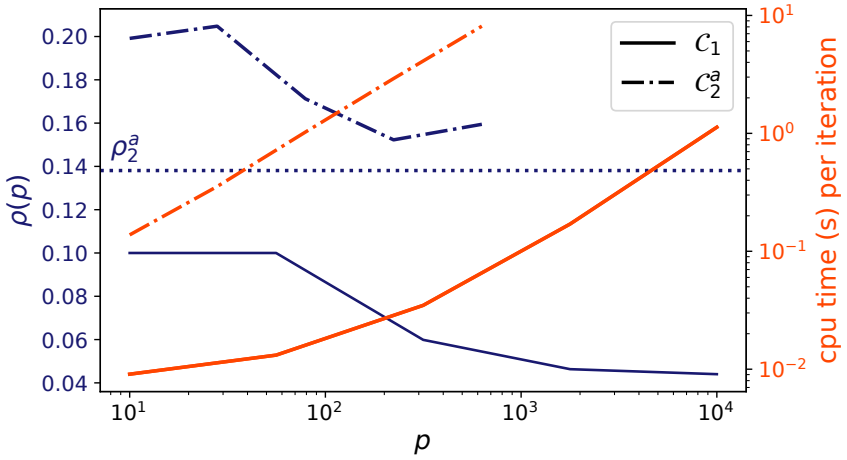


Figure 4.4: Results of $\mathcal{C}_1$, $\mathcal{C}_2^a$, and cpu time on *Satellite*, depending on p.

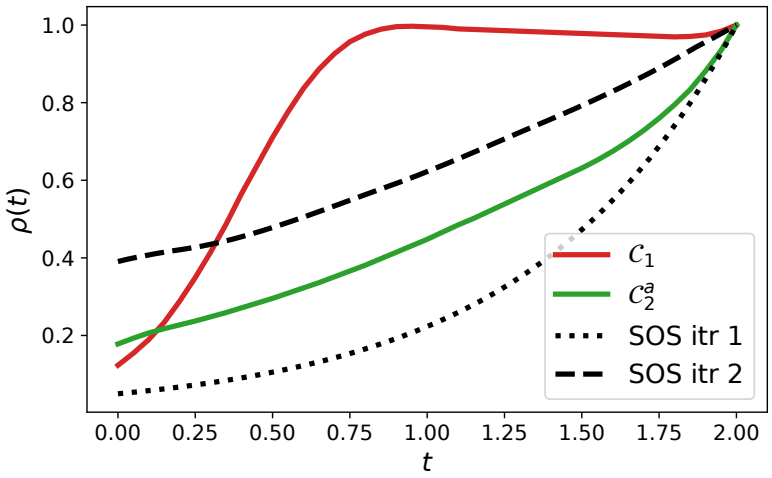


Figure 4.5: $\rho(t)$ with different certificates, around a trajectory of *Vanderpol*. The total CPU time is 7s for two iterations of SoS, roughly 1s for $\mathcal{C}_1$, $\mathcal{C}_2^a$.

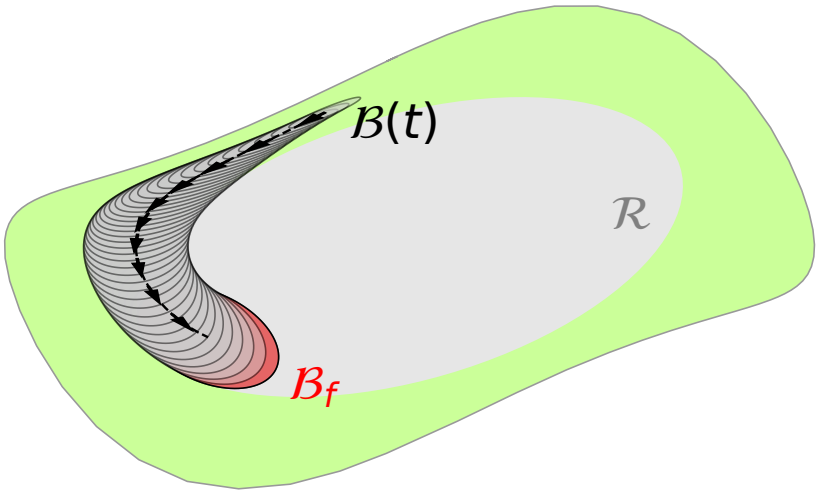


Figure 4.6: A funnel $\mathcal{B}(t)$ around a trajectory of *Vanderpol*, obtained with $\mathcal{C}_1$. The state-space is in green, $\mathcal{R}$ in light gray is an ROA around 0, and $\mathcal{B}_f$ in red is the target region. The reference trajectory is displayed with arrows.

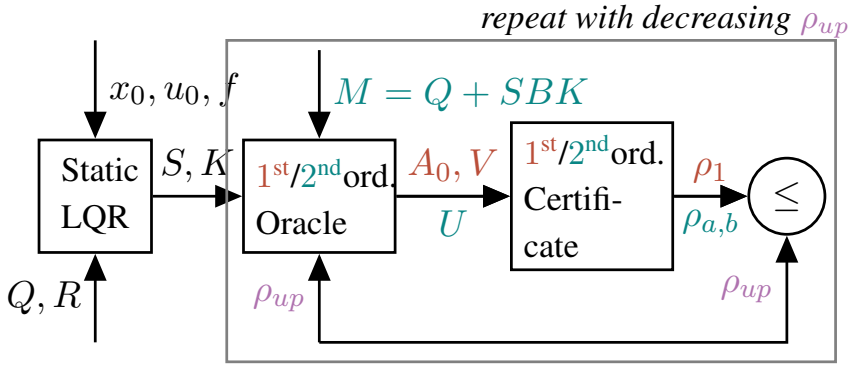


Figure 4.7: ROA estimation algorithm. Elements specific to the 1^{st} order method are in red, to the 2^{nd} order in blue. Framed steps are repeated until $\rho \geq \rho_{up}$.

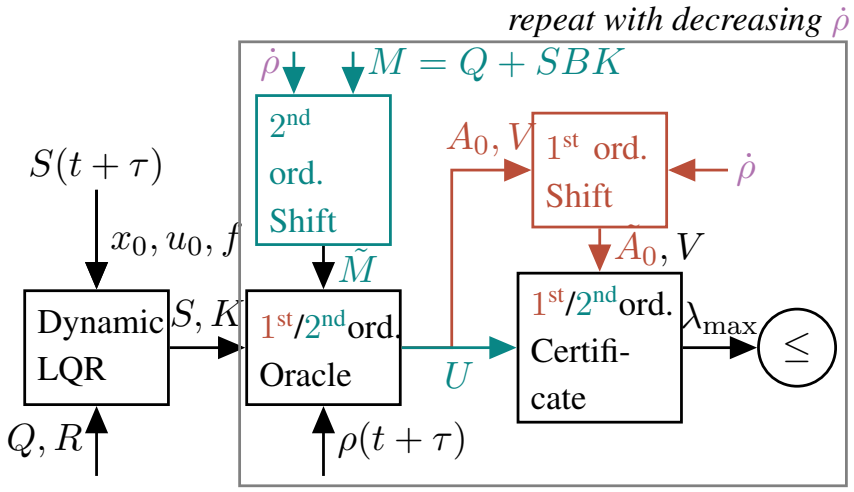


Figure 4.8: Trajectory tracking algorithm for one time-step. Framed steps are repeated while equations (4.45) for the 1^{st} order, or (4.48) for the 2^{nd} order, hold.

Chapter 5

Infinite-Dimensional Sums-of-Squares for Optimal Control

Abstract. *We introduce an approximation method to solve an optimal control problem* via *the Lagrange dual of its weak formulation. This method applies to problems with an unknown, non-necessarily polynomial dynamics, accessed through samples, akin to model-free reinforcement learning. It is based on a sum-of-squares representation of the Hamiltonian, and extends a previous method from polynomial optimization to the generic case of smooth problems. Such a representation is infinite-dimensional and relies on a particular space of functions – a reproducing kernel Hilbert space – chosen to fit the structure of the control problem. After subsampling, it leads to a practical method that amounts to solving a semi-definite program. We illustrate our approach by a numerical application on a low-dimensional control problem.*

This chapter is based on our work *Infinite-Dimensional Sums-of-Squares for Optimal Control*, with Justin Carpentier, Alessandro Rudi and Francis Bach, accepted for publication in the Conference on Decision and Control, 2022.

Contents

5.1 Introduction

The continuous-time optimal control problem (OCP) is a generic formalism modeling a wide variety of nonlinear systems and optimization criteria, with countless industrial applications, including aerospace (Trélat, 2012) or robotics (Murray et al., 2017). Developing efficient numerical methods for solving such a general problem is a daunting task, especially for high-dimensional systems. Among current methods, *indirect methods* exploit optimality criteria derived from Pontryagin's maximum principle and give precise results but need to be initialized properly, while the less accurate *direct methods* reformulate the problem as a nonlinear program without specific initialization requirements (Trélat, 2005, Chapter 9).

In this chapter, we focus on a direct method that computes the optimal value function of the problem as the maximal subsolution of the Hamilton-Jacobi-Bellman (HJB) equation. It is described in Hernández-Hernández et al. (1996); Lasserre et al. (2008) and is obtained by taking the dual of the *weak formulation* of the OCP, involving occupation measures (Vinter, 1993). In Lasserre et al. (2008), the numerical resolution of this formulation is based on polynomial optimization (Lasserre, 2015), and hence applies to semi-algebraic dynamics, constraints and cost functions, with a possibly costly extension to smooth functions, involving a hierarchy of semi-definite programs (SDPs).

Our main contribution is to extend the numerical method of Lasserre et al. (2008) to non-polynomial, smooth OCPs, when the dynamics and costs are unknown and only accessed through a finite number of samples. In this sense, our approach is *sample-based*, or data-driven (Kutz et al., 2016). As a side benefit, it is gradient-free, hence directly applicable to systems where the dynamics is not known analytically, let alone polynomial, like soft robots (Della Santina et al., 2021) or contact interactions (Brogliato, 1999), just to name a few. To this end, we consider a space of smooth functions, called a reproducing kernel Hilbert space (RKHS) (Aronszajn, 1950), and use representations of non-negative functions in this space with sums-of-squares (Marteau-Ferey et al., 2020). This is detailed in Sections 5.2 and 5.3. In Section 5.4, we notably prove that this representation is exact in two particular cases: the time-invariant linear quadratic regulator (LQR) and smooth control-affine systems. This results in a practical numerical method derived in Section 5.5, that requires to solve an SDP to approximate the optimal value function of the OCP. Finally, in Section 5.6 we illustrate the practical application of this method to a simple two-dimensional OCP, with a comparison to a sample-based baseline.

5.2 Background

First, we introduce the three building blocks that are then combined in Section 5.3 to design our approximation method.

5.2.1 Formulation of OCP with Maximal Subsolutions of HJB

Let $\mathcal{X}$ and $\mathcal{U}$ be compact subsets of $\mathbb{R}^d$ and $\mathbb{R}^p$, for integers $d, p \geq 1$, and assume that $\mathcal{U}$ is convex. We define the dynamics $f : [0,T] \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}^d$, the running cost $L : [0,T] \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, and the terminal cost $M : \mathcal{X} \to \mathbb{R}$ occurring at the fixed terminal time $T > 0$. In addition, we assume that optimal state trajectories remain in the compact set $\mathcal{X}$, and that sampling points from $\mathcal{X}$ and $\mathcal{U}$ is easy. We do not consider problems with explicit path-constraints of the form $g(t,x,u) \geq 0$ which are left as future work.

We assume the existence of a smooth function $V^* \in C^1(\mathcal{X}_T)$, meaning that V^* is smoothly differentiable on $\mathcal{X}_T := [0,T] \times \mathcal{X}$, that is a solution of the HJB equation: $\forall (t,x) \in \mathcal{X}_T$,

$$\begin{aligned} &\frac{\partial V^*}{\partial t}(t,x) + \inf_{u \in \mathcal{U}} \Big(L(t,x,u) + \nabla V^*(t,x)^\top f(t,x,u) \Big) = 0 \\ &V^*(T,x) = M(x), \end{aligned} \tag{5.1}$$

where ∇V^* refers to the gradient of V^* w.r.t. x. For $t \in [0,T]$, let $\mathcal{U}_t$ the set of admissible controls such that $\forall s \in [t,T], (x(s),u(s)) \in \mathcal{X} \times \mathcal{U}$. Then V^* is the value function (Liberzon, 2011) of the following OCP: $\forall (t_0,x_0) \in \mathcal{X}_T$,

$$\begin{aligned} &V^*(t_0,x_0) = \inf_{u \in \mathcal{U}_{t_0}} \int_{t_0}^{T} L(t,x(t),u(t))\mathrm{d}t + M(x(T)) \\ &\forall t \in [t_0,T],\ \dot{x}(t) = f(t,x(t),u(t)), \quad x(0) = x_0. \end{aligned} \tag{5.2}$$

Let μ_0 be a probability measure on $\mathcal{X}$. We are interested in the value of the stochastic initial point problem, where x_0 is drawn according to μ_0, that is, $\mathbb{E}_{x_0 \sim \mu_0}\left[V^*(0,x_0)\right] = \int V^*(0,x_0)\mathrm{d}\mu_0(x_0)$.

In this chapter, instead of directly looking for solutions of the HJB equation, we will focus on the following alternative problem (P), namely, finding a maximal subsolution of HJB:

$$\begin{aligned} &\sup_{V \in C^1(\mathcal{X}_T)} \int V(0,x_0)\mathrm{d}\mu_0(x_0) \\ &\forall (t,x,u),\ \frac{\partial V}{\partial t}(t,x) + L(t,x,u) + \nabla V(t,x)^\top f(t,x,u) \geq 0 \\ &\forall x,\ V(T,x) \leq M(x). \end{aligned} \tag{P}$$

This is the dual of the *weak formulation* of the OCP with occupation measures, which is a linear program over the space of measures (Kamoutsi et al., 2017; Lasserre, 2010; Vinter, 1993). Moreover, subsolutions of HJB also play a key role in the theory of viscosity solutions (Crandall and Lions, 1983) of partial differential equations, as well as for approximating reachable sets through dissipativity conditions (Jones and Peet, 2019). The first constraint in (P) is the positivity of a certain Hamiltonian associated to V, namely:

$$H(t,x,u) := \frac{\partial V}{\partial t}(t,x) + L(t,x,u) + \nabla V(t,x)^\top f(t,x,u). \tag{5.3}$$

If $V = V^*$ is the optimal value function, then the optimal controller $u = u^*(t,x)$ minimizes the positivity constraint and for all $(t,x) \in \mathcal{X}_T$, $H^*(t,x,u^*(t,x)) = 0$.

Our goal is to find an approximate solution V of (P). Under some additional assumptions, (P) is equivalent to the OCP. Such regularity and convexity assumptions were first studied by Vinter (1993) and are detailed

in Lasserre et al. (2008). We refer to Chapter 2, Section 2.2.3 for a more detailed discussion on this point. In this particular case, the value of problem (P) coincides with the one of the stochastic initial point problem:

$$\sup P = \mathbb{E}_{x_0 \sim \mu_0} \left[V^*(0, x_0) \right]. \tag{5.4}$$

We always assume that this is the case, and in particular such conditions are met by the numerical example of Section 5.6.

5.2.2 Parameterization of the Value Function

A first difficulty in problem (P) is searching V in the infinite-dimensional set $C^1(\mathcal{X}_T)$. One option is to search V in a finitely-parameterized set $\mathcal{F}_\Theta$. A common practice, notably in approximate dynamic programming and reinforcement learning (Sutton and Barto, 2018, Chapter 9), is to use a linear approximation of V, with a feature vector $\psi(t, x) \in \mathbb{R}^m$ and a parameter θ in a convex subset Θ of $\mathbb{R}^m$, for $m \geq 1$.

Since we assumed that V^* is a solution of the HJB equation, we can restrict that search space in (P) to functions V such that $V(T, .) = M(.)$. Then we assume that the parameterization is such that for any θ, $V_\theta(T, .) = M(.)$, so that we can remove the explicit constraint in (P). Hence our parameterized set is:

$$\mathcal{F}_\Theta := \{(t, x) \mapsto V_\theta(t, x) = \theta^\top \psi(t, x) + M(x) \mid \theta \in \Theta\}, \tag{5.5}$$

with ψ such that $\psi(T, .) = 0$. To simplify the evaluations of ∇V_θ and $\frac{\partial V_\theta}{\partial t}$, it is convenient (but not necessary) to use a separable feature vector $\psi(t, x) = \kappa(t)\varphi(x)$, with $\kappa(T) = 0$.

5.2.3 Representing Non-Negative Functions as Sum-of-Squares

Problem (P) is constrained by a dense set of inequalities indexed by $(t, x, u) \in [0, T] \times \mathcal{X} \times \mathcal{U}$, which cannot be directly handled by numerical algorithms. Hence we look for a finite – possibly approximate – representation of the non-negative function $(t, x, u) \mapsto H(t, x, u) \geq 0$.

If f, L, M are semi-algebraic functions, one way is to use sum-of-squares (SoS) polynomials (Lasserre, 2015), *i.e.*, to represent $H(t, x, u)$ as the sum of the squares of polynomials of a given degree. This is a sufficient but not necessary condition for being a non-negative polynomial. This technique has been applied to problem (P) by Lasserre et al. (2008), although it is presented in its dual version using the method of moments (Henrion, 2014). In any case, this representation is not exact in general and gives a lower approximation of (P). To numerically solve the problem, one needs to build a hierarchy of SDPs obtained by this SoS representation with polynomials of increasing degree r. Under generic conditions, this hierarchy converges to the value of (P), and recent results have brought explicit convergence rates (Baldi and Mourrain, 2021; Korda et al., 2017). Yet, the size of the SDP at rank r is defined by the number of monomials of degree less than r in the dimension of (t, x, u), which is $\binom{d+p+1+r}{r}$, a quantity growing exponentially with r.

In this chapter, we opt for another option inspired by recently-introduced machine learning techniques (Rudi et al., 2020): representing a non-negative function as a SoS in a reproducing kernel Hilbert space (RKHS). Hereafter, we briefly define an RKHS and mention its main properties, and refer to Paulsen and Raghupathi (2016) for a thorough description. Consider a set E, a function $k : E \times E \to \mathbb{R}$ is a positive definite kernel if $\forall n \in \mathbb{N}, y_1, \ldots, y_n \in E$, the matrix $K := (k(y_i, y_j))_{i,j=1}^n$ is positive semi-definite. Associated to a positive definite kernel k, there exists a unique RKHS $\mathcal{H}$, a Hilbert space of functions $E \to \mathbb{R}$, with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, such that the following properties hold (the second one is the so-called "reproducing property"):

- $\forall y \in E, k_y := k(y, \cdot) \in \mathcal{H}$;
- $\forall g \in \mathcal{H}, \forall y \in E, \langle g, k_y \rangle_{\mathcal{H}} = g(y)$.

In addition, there exists a feature map $\Phi : E \to \mathcal{H}$, possibly infinite-dimensional, defined by $\Phi(y) = k_y$, which maps a point in E to a function in $\mathcal{H}$, and in particular we have $k(y, y') = \langle \Phi(y), \Phi(y') \rangle_{\mathcal{H}}$. Conversely, any feature map defines an RKHS associated to the former kernel.

Here we mention two classical kernels from the RKHS literature, which we use later to build our function representations. Assume that E is a subset of $\mathbb{R}^\ell$, for $\ell \geq 1$. The polynomial kernel of degree r is defined on $E \times E$ by $k(y, y') = (1 + y^\top y')^r$, and the corresponding embedding $\Phi(y)$ is the vector of $\binom{\ell+r}{r}$ multivariate monomials of degree less than r. In this case $\mathcal{H}$ is finite-dimensional. The exponential kernel is defined by $k(y, y') = \exp(-\|y - y'\|_2/\sigma)$, with $\sigma > 0$. If E is bounded and has locally Lipschitz boundary, the corresponding RKHS is the Sobolev space of functions whose weak-derivatives up to order $s = \ell/2 + 1/2$ are square-integrable (Berlinet and Thomas-Agnan, 2011).

Functions that are SoS in an RKHS $\mathcal{H}$ can be represented using an infinite-dimensional positive semi-definite operator (Rudi et al., 2020, Corollary 1). Indeed, assume that a function $h \in \mathcal{H}$ is written as a sum-of-squares of functions $h_j \in \mathcal{H}$:

$$\forall y \in E, \quad h(y) = \sum_{j=1}^{q} h_j(y)^2. \tag{5.6}$$

For any $v, w \in \mathcal{H}$, we have:

$$\langle w, (v \otimes v)w \rangle = \langle w, vv^* w \rangle = \mathrm{Tr}(w^* vv^* w) = (\langle v, w \rangle)^2, \tag{5.7}$$

where $\otimes$ denotes the outer product. Because of the reproducing property: $\forall j \in \{1, ..., q\}, y \in E$,

$$h_j(y)^2 = (\langle \Phi(y), h_j \rangle_{\mathcal{H}})^2 = \langle \Phi(y), (h_j \otimes h_j)\Phi(y) \rangle. \tag{5.8}$$

Consequently:

$$h(y) = \langle \Phi(y), \mathcal{A}\Phi(y) \rangle_{\mathcal{H}}, \tag{5.9}$$

with $\mathcal{A} := \sum_{j=1}^{q} h_j \otimes h_j \in \mathbb{S}_+(\mathcal{H})$ and $\mathcal{A}$ has rank at most q, where $\mathbb{S}_+(\mathcal{H})$ is the set of bounded self-adjoint positive semi-definite operators on $\mathcal{H}$.

In Marteau-Ferey et al. (2020), this SoS representation in an RKHS is used to model non-negative functions, *e.g.*, for signal processing or statistics applications. In some cases, *e.g.*, in Sobolev spaces as we will see hereafter, the representation is exact in the sense that all non-negative functions can be written as a SoS in $\mathcal{H}$, whereas the polynomial SoS representation is tight only for a restricted class of polynomials (Lasserre, 2010). Besides, SoS polynomials are a particular case of what has just been described, if k is the polynomial kernel. In the rest of the chapter, we will extend the method of Lasserre et al. (2008) from polynomials to any RKHS, at the expense of possibly infinite-dimensional representations.

5.3 Dense Set of Inequality Constraints

In this section, we start by providing a basic relaxation of problem (P) which is naturally compatible with a sample-based approach, and a motivation for preferring a SoS representation of the non-negativity constraints in (P). Then we present the resulting problem and its main features.

5.3.1 Relaxed Formulation by Subsampling

A straightforward relaxation of (P) is obtained by finitely subsampling the non-negativity constraints. Let us sample values of $(t^{(i)}, x^{(i)}, u^{(i)})_{i\in I}$ in $[0,T] \times \mathcal{X} \times \mathcal{U}$, with I a finite set of cardinality $n \geq 1$. For simplicity, assume that μ_0 is the mean of n_0 Diracs at points $\{x_0^{(i)}\}_{i\in\{1,\dots,n_0\}}$. Besides, let us use a linear parameterization of V as described in Section 5.2.2, with $\Theta = \mathbb{R}^m$. We then obtain a linear program, with a possibly unbounded solution in the overparameterized setting ($m \gg n$). To circumvent this effect, we add a quadratic regularizer on θ with parameter $\lambda_\theta \geq 0$, and obtain the following problem:

$$\begin{aligned} \sup_{\theta\in\mathbb{R}^m} \ & \frac{1}{n_0}\sum_{k=1}^{n_0} \theta^\top \psi(0, x_0^{(k)}) + M(x_0^{(k)}) - \lambda_\theta \|\theta\|_2^2 \\ \forall i \in I, \quad & \theta^\top \frac{\partial \psi}{\partial t}(t^{(i)}, x^{(i)}) + L(t^{(i)}, x^{(i)}, u^{(i)}) \\ & + \Big(\theta^\top \nabla\psi(t^{(i)}, x^{(i)}) + \nabla M(x^{(i)})\Big)^\top f(t^{(i)}, x^{(i)}, u^{(i)}) \geq 0. \end{aligned} \tag{LP}$$

Although this is not exactly a linear program if $\lambda_\theta > 0$, it can still can be solved easily by standard solvers, and we will refer to it as the LP problem. A similar finite-dimensional LP formulation has been proposed by Gaitsgory and Quincampoix (2009) for discounted infinite-horizon control problems. It is part of a long series of LP formulations for optimal control (see, *e.g.*, Gaitsgory et al. (2017) and references therein), for dynamic programming and more recently for reinforcement learning (Lu et al., 2021a).

This problem will be used as a baseline in Section 5.6, to be compared with the SoS formulation below. It is a relaxation that gives an upper-bound on (P), but relating the number of samples n to the quality of the approximation and ultimately to the performance of the controller is a challenging problem, even for LQR (Dean et al., 2020). Yet in the example below, this can be evaluated explicitly.

Example 1. Let $g : \mathbb{R}^p \to \mathbb{R}$ be a smooth function with a unique minimizer u^*. With $T = 1$, $L : (t, x, u) \mapsto g(u)$, $f = 0$ and $M = 0$, then $V^*(t, x) = (1 - t)g(u^*)$ and solving the OCP is essentially equivalent to finding the global minimizer of g. If V is parameterized by $V_\theta(t, x) = \theta(1 - t)$, the LP formulation with $\lambda_\theta = 0$ writes:

$$\sup\ \theta \quad \text{such that} \quad \forall i \in I,\ -\theta + g(u_i) \geq 0, \tag{5.10}$$

which is readily solved by $\theta = \min\{g(u_i)\}_{i\in I}$. In general, this method requires $O(\varepsilon^{-p})$ samples to approximate $g(u^*)$ with precision ε (Novak, 2006), and yet when g is smooth, this is not an optimal way to perform zero-th order optimization. Indeed, Rudi et al. (2020) use a SoS representation of $g - \theta$ to solve this exact problem, and alleviate the curse of dimensionality when g is sufficiently smooth: the number of samples reduces to $O(\varepsilon^{-p/s})$ for $g \in C^s(\mathbb{R}^p)$. In the rest of this chapter, we propose to use the same approach and to generalize it to any OCP. We expect similar benefits when H is smooth.

5.3.2 Strengthened Formulation by SoS Representation

Consider an RKHS $\mathcal{H}$ of real-valued functions on $[0,T] \times \mathcal{X} \times \mathcal{U}$, with positive definite kernel k, and $\Phi : E \to \mathcal{H}$ the corresponding embedding. We use a SoS representation in $\mathcal{H}$, or "kernel SoS", for the

constraint $H \geq 0$ in (P):

$$\sup_{\substack{V \in C^1(\mathfrak{X}_T), \\ \mathcal{A} \in \mathbb{S}_+(\mathcal{H})}} \int V(0, x_0) \mathrm{d}\mu_0(x_0)$$
$$\forall (t,x,u),\ H(t,x,u) = \langle \Phi(t,x,u), \mathcal{A}\Phi(t,x,u) \rangle. \tag{KSOS}$$

This is a strengthening of the constraint in (P), since being SoS is stronger than being non-negative. So in general, (KSOS) is a lower-approximation of (P). However, in certain cases, (KSOS) can be equivalent to (P), as we will prove in Section 5.4. A sufficient condition is the existence of $\mathcal{A} \in \mathbb{S}_+(\mathcal{H})$ such that, at the optimal V^*:

$$\forall (t,x,u),\ H^*(t,x,u) = \langle \Phi(t,x,u), \mathcal{A}\Phi(t,x,u) \rangle. \tag{5.11}$$

5.4 Tight Sum-of-Squares Representations

We study the tightness of problem (KSOS) in two particular cases: the time-invariant LQR and smooth value functions.

5.4.1 Case 1: Infinite-Horizon Time-Invariant LQR

First we look at a very simple OCP, where every quantity can be computed almost in closed form, and with infinite-horizon so that there is no dependence in t. Let $f(x,u) = A_0 x + B_0 u$, for $A_0 \in \mathbb{R}^{d \times d}$, $B_0 \in \mathbb{R}^{d \times p}$, with (A_0, B_0) controllable, $L(x,u) = x^\top Q_0 x + u^\top R_0 u$, $Q_0 \in \mathbb{S}_+(\mathbb{R}^{d \times d})$, $R_0 \in \mathbb{S}_+(\mathbb{R}^{p \times p})$, $R_0 \succ 0$. The optimal value function is $V^*(x) = x^\top S_0 x$, where S_0 is the unique positive semi-definite solution of the algebraic Riccati equation:

$$0 = -Q_0 - A_0^\top S_0 - S_0 A_0 + S_0 B_0 R_0^{-1} B_0^\top S_0. \tag{5.12}$$

The optimal controller is $u^*(x) = -R_0^{-1} B_0^\top S_0 x =: -K_0 x$.

The Hamiltonian is:

$$\begin{aligned} H^*(x,u) &= x^\top Q_0 x + u^\top R_0 u + 2x^\top S_0 (A_0 x + B_0 u) \\ &= u^\top R_0 u + x^\top S_0 B_0 u + u^\top B_0^\top S_0 x + x^\top S_0 B_0 K_0 x. \end{aligned} \tag{5.13}$$

This is a SoS of degree-one polynomials in (x,u):

$$\begin{aligned} H^*(x,u) &= (u + K_0 x)^\top R_0 (u + K_0 x) \\ &= \begin{pmatrix} x^\top & u^\top \end{pmatrix} \begin{pmatrix} K_0^\top \\ I_p \end{pmatrix} R_0 \begin{pmatrix} K_0 & I_p \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \\ &= \sum_{j=1}^{p} [q_j(x,u)]^2, \end{aligned} \tag{5.14}$$

with $q_j(x,u) := [R_0^{1/2}]_j . \begin{pmatrix} K_0 & I_p \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix}$.

Therefore, an infinite-horizon, time-invariant LQR with unknown parameters can be equivalently expressed by:

$$\sup_{V \in C^1(\mathcal{X}),\ N \succeq 0} \int V(x_0) \mathrm{d}\mu_0(x_0)$$
$$\forall (x,u), \quad L(x,u) + \nabla V(x)^\top f(x,u) = \begin{pmatrix} x \\ u \end{pmatrix}^\top N \begin{pmatrix} x \\ u \end{pmatrix}. \tag{5.15}$$

In the next section, we prove that similar SoS constructions exist for sufficiently smooth OCPs, possibly with an infinite-dimensional embedding (*v.s.* a $(d+p)$-dimensional one here).

5.4.2 Sum-of-Squares Decomposition with Smooth Functions

We show that, for smooth and control-affine OCPs, H^* is a SoS of smooth functions.

Let $\Omega_2 := \mathrm{Int}(\mathcal{U})$, $\Omega_1 := \mathrm{Int}\{(t,x) \in \mathcal{X}_T \mid \mathrm{argmin}_{u \in \mathcal{U}} H^*(t,x,u) \subset \Omega_2\}$, and $\Omega := \Omega_1 \times \Omega_2$.

Theorem 6. *Let $s \in \mathbb{N}$, $s \geq 1$. Assume that:*

• *f is control-affine: $\forall (t,x,u) \in [0,T] \times \mathcal{X} \times \mathcal{U}$,*

$$f(t,x,u) = g(t,x) + B(t,x)u. \tag{5.16}$$

• *For all $(t,x) \in \Omega_1$, $u \mapsto L(t,x,u)$ is twice differentiable on Ω_2 and strongly convex:*

$$\nabla_u^2 L(t,x,u) \succcurlyeq \rho I, \tag{5.17}$$

for some $\rho > 0$, and $(t,x,u) \mapsto \nabla_u^2 L(t,x,u) \in C^s(\Omega)$.

• $(t,x,u) \mapsto \nabla_u L(t,x,u) + B(t,x)^\top \nabla_x V^*(t,x) \in C^s(\Omega)$.

Then there exist p functions $(w_j)_{1 \leq j \leq p} \in C^s(\Omega)$ such that:

$$\forall (t,x,u) \in \Omega, \quad H^*(t,x,u) = \sum_{j=1}^{p} w_j(t,x,u)^2. \tag{5.18}$$

Proof. Consider the Hamiltonian with f control-affine:

$$H^*(t,x,u) = \nabla V^*(t,x)^\top g(t,x) + \frac{\partial V^*}{\partial t}(t,x) + L(t,x,u) + \nabla V^*(t,x)^\top B(t,x)u. \tag{5.19}$$

Pontryagin's maximum principle states that:

$$\forall (t,x) \in \Omega_1, \quad \inf_{u \in \mathcal{U}} H^*(t,x,u) = 0. \tag{5.20}$$

Since L is strongly convex in u, the minimizer is unique and we call it $u^*(t,x)$. By definitions of Ω_1 and Ω_2, we have a mapping $u^* : \Omega_1 \to \Omega_2$ and it is characterized by:

$$\nabla_u H^*(t,x,u) = \nabla_u L(t,x,u^*(t,x)) + B(t,x)^\top \nabla_x V^*(t,x) = 0.$$

Since $(t,x,u) \mapsto \nabla_u^2 L(t,x,u)$ is continuous on Ω and invertible, and $(t,x,u) \mapsto \nabla_u H^*(t,x,u) \in C^s(\Omega)$, then the implicit function theorem ensures that $u^* \in C^s(\Omega_1)$ (see Schwartz (1981), Chapter 8, Theorems 25 & 31).

For $(t,x,u) \in \Omega$, we use Taylor's formula around $u^*(t,x)$:

$$\begin{aligned} H^*(t,x,u) = H^*(t,x,u^*(t,x)) \\ + \nabla_u H^*(t,x,u^*(t,x))^\top (u - u^*(t,x)) \\ + (u - u^*(t,x))^\top R(t,x,u)(u - u^*(t,x)), \end{aligned} \tag{5.21}$$

with

$$R(t,x,u) := \int_0^1 (1-\tau)\nabla_u^2 H^*(t,x,(1-\tau)u^*(t,x) + \tau u)\mathrm{d}\tau. \tag{5.22}$$

Since by definition of u^*, we have $H^*(t,x,u^*(t,x)) = 0$, $\nabla_u H^*(t,x,u^*(t,x)) = 0$, and in addition, $\nabla_u^2 H^*(t,x,\cdot) = \nabla_u^2 L(t,x,\cdot)$, then we have:

$$H^*(t,x,u) = (u - u^*(t,x))^\top R(t,x,u)(u - u^*(t,x)), \text{ and} \tag{5.23}$$

$$R(t,x,u) = \int_0^1 (1-\tau)\nabla_u^2 L(t,x,(1-\tau)u^*(t,x) + \tau u)\mathrm{d}\tau \succcurlyeq \frac{\rho}{2} I. \tag{5.24}$$

For $(t,x,u) \in \Omega$, $R(t,x,u)$ has a positive-definite square root $\sqrt{R(t,x,u)}$.

Also, $\forall \tau \in [0,1]$, $(1-\tau)u^*(t,x) + \tau u \in \Omega_2$ because $\mathrm{Int}(\mathcal{U})$ is convex like $\mathcal{U}$.

Since $\forall i,j$, $\frac{\partial^2 L}{\partial u_i \partial u_j} \in C^s(\Omega)$, $u^* \in C^s(\Omega_1)$, and $\sqrt{\cdot}$ is C^∞ on $\{M \mid M^\top = M, M \succcurlyeq \frac{\rho}{2} I\}$, then $r_{i,j} : (t,x,u) \mapsto e_i^\top \sqrt{R(t,x,u)} e_j \in C^s(\Omega)$, and we have the decomposition:

$$H^*(t,x,u) = \sum_{i=1}^p w_i(t,x,u)^2, \quad \text{with} \tag{5.25}$$

$$\begin{aligned} w_i(t,x,u) &:= \sqrt{R(t,x,u)}_{i.} (u - u^*(t,x)) \\ &= \sum_{j=1}^p r_{i,j}(t,x,u) \left(e_j^\top (u - u^*(t,x)) \right), \end{aligned} \tag{5.26}$$

and each $w_i \in C^s(\Omega)$. One can easily check that for a time-varying LQR, we recover this explicit SOS decomposition with $R(t,x,u) = R$, $u^*(t,x) = -K(t)x$, $K(t) = R^{-1}B^\top(t)S(t)$, and $S(.)$ the positive definite solution of the Riccati differential equation. □

This result motivates the use of exponential kernels, inducing a Sobolev space RKHS, to represent the non-negativity constraints in (P) for smooth OCPs. When $s > d/2 + 1$, by applying a technique similar to the one used under Corollary 2 of Rudi et al. (2020), it is possible to obtain a SoS representation in terms of the exponential kernel. Then (KSOS) is equivalent to (P).

5.4.3 Stochastic Smoothing of the Optimal Value Function

However in general, V^* is not necessarily smooth (*e.g.*, minimal time problems), nor is u^* (*e.g.*, *bang-bang* controllers that are not even continuous), even with differentiable dynamics and cost functions. Here, we provide a generic technique to give some regularity to V^*. For $\eta > 0$, we can define a perturbed version of the control system (Fleming and Rishel, 2012), where the state is a random variable X_t:

$$\mathrm{d}X_t = f(t, X_t, u_t)\mathrm{d}t + (2\eta)^{1/2}\mathrm{d}B_t \,, \tag{5.27}$$

where B_t is a standard Brownian motion independent of X_t. We define the optimal value function, with $X_0 = x_0$:

$$V^\eta(t_0, x_0) = \inf_{u \in \mathcal{U}} \mathbb{E}\left[\int_{t_0}^{T} L(t, X_t, u_t)\mathrm{d}t + M(X_T)\right]. \tag{5.28}$$

V^η is the unique $C^{1,2}(\mathcal{X}_T)$ (C^1 in t, C^2 in x) solution (Fleming and Rishel, 2012) of the following regularized HJB equation: $\forall (t,x) \in \mathcal{X}_T$,

$$\begin{aligned}\inf_{u \in \mathcal{U}} \{L(t,x,u) &+ \nabla V^\eta(t,x)^\top f(t,x,u)\} \\ &+ \frac{\partial V^\eta}{\partial t}(t,x) + \eta \Delta V^\eta(t,x) = 0 \,,\end{aligned} \tag{5.29}$$

with $V^\eta(T,x) = M(x)$. ΔV^η refers to the Laplacian with respect to x. Contrary to HJB, the solutions are at least $C^{1,2}(\mathcal{X}_T)$ because this is a quasilinear parabolic partial differential equation (Lieberman, 1996). The regularization $\eta\Delta V$ is a vanishing viscosity term, and the optimal controller is still in $\operatorname{argmin}_u L + \nabla V^{\eta\top} f$. Generically, V^η converges to V^* as $\eta \to 0$, following a reasoning similar to the theory of viscosity solutions (Crandall and Lions, 1983).

5.5 SDP Formulation and its Numerical Resolution

5.5.1 Finite-Dimensional Formulation via Subsampling

Similarly to the (LP) formulation that relaxes (P), we will now derive a relaxation of problem (KSOS), which is another relaxation of problem (P) if the SoS representation of H^* is tight. Going through (KSOS) as an intermediate step will help to exploit the structure of (P). Using a parameterization of V in $\mathcal{F}_\Theta$ with $\Theta = \mathbb{R}^m$, and a set of sampled points $(t^{(i)}, x^{(i)}, u^{(i)})_{i \in I}$ in $[0,T] \times \mathcal{X} \times \mathcal{U}$, with $|I| = n$, we obtain:

$$\begin{aligned}\sup_{\mathcal{A} \in \mathbb{S}_+(\mathcal{H}), \theta \in \Theta} \quad & c^\top\theta - \lambda_\theta\|\theta\|_2^2 - \lambda \operatorname{Tr}(\mathcal{A}) + C \\ \text{such that} \quad & \forall i \in \{1,\dots,n\}, \\ & b_i + a_i^\top\theta = \langle \Phi(t^{(i)}, x^{(i)}, u^{(i)}), \mathcal{A}\Phi(t^{(i)}, x^{(i)}, u^{(i)})\rangle,\end{aligned} \tag{5.30}$$

with $c := \sum_{i=1}^n \mu_0^{(i)} \psi(0, x^{(i)})$, $C := \sum_i \mu_0^{(i)} M(x^{(i)})$,

$$b_i := L(t^{(i)}, x^{(i)}, u^{(i)}) + \nabla M(x^{(i)})^\top f(t^{(i)}, x^{(i)}, u^{(i)}) + \eta \Delta M(x^{(i)}), \tag{5.31}$$

and

$$a_i := J_\psi(t^{(i)}, x^{(i)}) f(t^{(i)}, x^{(i)}, u^{(i)}) + \frac{\partial \psi}{\partial t}(t^{(i)}, x^{(i)}) + \eta \Delta \psi(t^{(i)}, x^{(i)}), \tag{5.32}$$

where J_ψ denotes the Jacobian matrix of ψ with respect to x only. Note that we integrate the stochastic smoothing process in this formulation, with parameter η that can be eventually set to 0.

The regularization parameter $\lambda > 0$ controls the trace of the infinite-dimensional operator $\mathcal{A}$, and allows for subsampling to provably recover the non-subsampled program when n tends to infinity, and λ goes to zero at the proper rate (see Rudi et al. (2020) for the precise dependence). In the limit $\lambda \to 0$, we recover the LP formulation where we assume nothing about the SoS representation of H^* in $\mathcal{H}$.

Both the operator $\mathcal{A}$ and the $\Phi(t^{(i)}, x^{(i)}, u^{(i)})$ can be infinite-dimensional, depending on the RKHS $\mathcal{H}$. Yet, following Rudi et al. (2020), we can reformulate the problem equivalently in finite dimension. Using the representer theorem in Marteau-Ferey et al. (2020), one can prove that $\mathcal{A}$ can be sought in the form: for $D \in \mathbb{R}^{n \times n}$, $D \succeq 0$,

$$\mathcal{A} = \sum_{i,j=1}^{n} D_{ij} \Phi(t^{(i)}, x^{(i)}, u^{(i)}) \otimes \Phi(t^{(j)}, x^{(j)}, u^{(j)}). \tag{5.33}$$

Simple computations detailed in Rudi et al. (2020) show that:

$$\begin{cases} \forall i,\ \langle \Phi(t^{(i)}, x^{(i)}, u^{(i)}), \mathcal{A} \Phi(t^{(i)}, x^{(i)}, u^{(i)}) \rangle = [KDK]_{ii}\,, \\ \mathrm{Tr}(\mathcal{A}) = \mathrm{Tr}(DK), \end{cases} \tag{5.34}$$

where K is the kernel matrix with entry (i,j) equal to $k\left((t^{(i)}, x^{(i)}, u^{(i)}), (t^{(j)}, x^{(j)}, u^{(j)})\right)$. Assume that $K \succ 0$. We denote by $K = R^\top R$ the Cholesky decomposition of K, with R an invertible upper-triangular matrix.

Let $B := RDR^\top$ and for $1 \leq i \leq n$, $\Phi_i := R_{\cdot i}$. Then:

$$\begin{cases} \mathrm{Tr}(B) = \mathrm{Tr}(DK) = \mathrm{Tr}(\mathcal{A}), \\ [KDK]_{ii} = \left[R^\top B R\right]_{ii} = \Phi_i^\top B \Phi_i. \end{cases} \tag{5.35}$$

The problem can now be reformulated as a finite-dimensional SDP over the positive semi-definite matrix $B \in \mathbb{R}^{n \times n}$:

$$\begin{array}{ll} \sup_{B \succcurlyeq 0, \theta \in \mathbb{R}^m} & c^\top \theta - \lambda_\theta \|\theta\|_2^2 - \lambda \,\mathrm{Tr}(B) + C \\ \text{such that} & \forall i \in \{1, \ldots, n\},\ b_i + a_i^\top \theta = (\Phi_i)^\top B \Phi_i\,. \end{array} \tag{5.36}$$

An important question is to estimate the number of subsampled inequalities sufficient to ensure that (5.36) $\Leftrightarrow$ (KSOS). If nothing is assumed on the structure of H, as in the LP method, this number is infinite. In contrast, the kernel SoS representation can reduce it or make it finite. If H is a polynomial of degree $2r$, k is the polynomial kernel of degree r, then $n \geq 2r$ distinct sampled points are enough to interpolate H, and (5.36) $\Leftrightarrow$ (KSOS). Another example is global optimization of smooth functions (see *Example 1*) with the exponential kernel. We refer to Rudi et al. (2020) for the analysis of the convergence rates, with a lower dependence in the dimension for the kernel SoS when compared to direct inequality subsampling (corresponding to the LP approach).

5.5.2 Interior Point Method with the Damped Newton Method

Problem (5.36) can be readily solved by any off-the-shelf SDP solver. However, for large n, this quickly becomes too computationally demanding. Here, we propose a numerical scheme based on the one proposed by Rudi et al. (2020) that scales better with the number of subsamples n. First, we introduce a slack variable $\delta \in \mathbb{R}^n$ allowing the constraints to be slightly violated (*e.g.*, because $\mathcal{F}_\Theta$ is not a perfect model), controlled by a large parameter $\gamma > 0$. Second, we introduce a log-barrier term controlled by a small $\varepsilon > 0$, useful to form the dual of the SDP. We obtain the following problem:

$$\begin{aligned} \sup_{\substack{B \succcurlyeq 0, \\ \theta, \delta}} \; & c^\top \theta - \lambda \operatorname{Tr}(B) - \lambda_\theta \|\theta\|_2^2 - \gamma \|\delta\|^2 + \varepsilon \log\det B + C \\ \text{such that} \; & \forall i \in \{1, \dots, n\},\ b_i + a_i^\top \theta = (\Phi_i)^\top B \Phi_i + \delta_i. \end{aligned} \tag{5.37}$$

The Lagrange dual of this problem reads:

$$\begin{aligned} \inf_{\alpha \in \mathbb{R}^n} \; & \sum_{i=1}^n \alpha_i b_i + \frac{1}{4\lambda_\theta} \sum_{j=1}^m \left(c_j + \sum_{i=1}^n \alpha_i a_{ij} \right)^2 \\ & - \varepsilon \log\det U(\alpha) + \frac{1}{4\gamma} \|\alpha\|_2^2 + \varepsilon n \log(\varepsilon / e) + C, \end{aligned} \tag{5.38}$$

where $U(\alpha) := \lambda I_n + \Phi^\top \operatorname{Diag}(\alpha) \Phi$, and $\Phi := R^\top$ is the matrix with rows $(\Phi_i)_{1 \le i \le n}$. Let us call the objective $F(\alpha)$.

Since F/ε is self-concordant (Boyd and Vandenberghe, 2004), like in Rudi et al. (2020), we propose to use damped Newton iterations (Nemirovski, 2004) on F/ε:

$$\alpha \leftarrow \alpha + \frac{1}{1 + \lambda(\alpha)} \Delta\alpha \,, \tag{5.39}$$

where $\Delta(\alpha) := -[F''(\alpha)]^{-1} F'(\alpha)$ is the Newton direction and $\lambda(\alpha) := \sqrt{\Delta\alpha^\top F''(\alpha) \Delta\alpha / \varepsilon}$ is the Newton decrement. The gradient and Hessian of F are computed by:

$$\frac{\partial F}{\partial \alpha_i} = b_i + \frac{1}{2\lambda_\theta} \sum_{j=1}^m a_{ij} \left(c_j + \sum_{k=1}^n a_{kj} \alpha_k \right) + \frac{1}{2\gamma} \alpha_i - \varepsilon \, \Phi_i^\top U(\alpha)^{-1} \Phi_i. \tag{5.40}$$

$$\frac{\partial^2 F}{\partial \alpha_i \partial \alpha_j} = \frac{1}{2\lambda_\theta} \sum_{k=1}^n a_{ik} a_{jk} + \varepsilon \left[\Phi_i^\top U(\alpha)^{-1} \Phi_j \right]^2 + \frac{\mathbf{1}_{i=j}}{2\gamma}. \tag{5.41}$$

At optimum, the value function is recovered by

$$\theta^\star = \frac{1}{2\lambda_\theta} \left(\sum_{i=1}^n \alpha_i^\star a_i + c \right), \tag{5.42}$$

and the dual variable $\alpha^\star$ plays a role similar to an occupation measure (Vinter, 1993), although it is not necessarily non-negative. To improve numerical stability in the experiments hereafter, we used an homotopy heuristics that progressively decreases the parameters λ_θ and ε. Moreover, parallel implementations are possible because no singular value decomposition is needed, only matrix operations and system inversions, with a computational complexity of $O(n^3)$ per iteration.

5.6 Numerical Example

In this section, we apply the kernel SoS method along with the basic LP method, on a two-dimensional control problem, namely the double integrator with finite horizon.

Setting. The problem is an LQR, as in Section 5.4.1, but with finite-horizon $T = 1$, $d = 2$, $p = 1$, $M(x) = \|x\|_2^2$,

$$A_0 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \; B_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \; Q_0 = I_2, \; R_0 = 0.1 \; . \tag{5.43}$$

The optimal value function and controller are $V^*(t,x) = x^\top S(t) x$,

$$u^*(t,x) = -R_0^{-1} B_0^\top S(t) x =: -K(t)x, \tag{5.44}$$

where $S(.)$ is the positive semi-definite solution of $S(T) = I_2$ and:

$$\dot{S}(t) = -Q_0 - A_0^\top S(t) - S(t) A_0 + S(t) B_0 R_0^{-1} B_0^\top S(t). \tag{5.45}$$

Parameterization of V. Let $V_\theta(t,x) = \theta^\top \psi(t,x) + M(x)$, where each entry of ψ is a product of basis functions on $\mathcal{X}$ and $[0,T]$. Let $\varphi(x) := (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)^\top$, because we know V^* is quadratic in x. For κ on $[0,T]$, we only know that it is a smooth function, so we use an approximate basis of the Sobolev space of functions with squared integrable derivatives: a sequence of sines and cosines with decreasing periods beginning with $2T$ to avoid constraining $V(0,.) = V(T,.)$, and $\kappa(T) = 0$, ensures that $V(T,.) = M(.)$:

$$\kappa(t) := \left(\frac{1}{\omega} \sin\left(\frac{\omega\pi}{2} \frac{t-T}{T} \right) \right)_{1 \le \omega \le m_t}^\top . \tag{5.46}$$

Finally, $\psi_{i+6j}(t,x) := \varphi_i(x)\kappa_j(t)$, and $\theta \in \mathbb{R}^m$, $m = 6m_t$. We choose $m_t = 10$, for which the performance of the policy of the projection of V^* on $\mathcal{F}_\Theta$ is almost perfect.

Evaluation. We give two criteria to evaluate the quality of an approximation V. First, the distance to V^*: $\|\bar{V} - \bar{V}^*\|^2$, where $\bar{V}$ is the vector of its values on a regular grid on $[0,T] \times [-1,1]^2$ with $10 \times 10 \times 10$ points. Second, the cost of the policy on a 10×10 regular grid of initial points.

Sampling. The set of samples $(t^{(i)}, x^{(i)}, u^{(i)})_{i \in I}$ is built as follows. The $x^{(i)}$ are n_x points in $[-1,1]^2$ generated by the Sobol sequence (Sobol', 1967), the $(u^{(i)})_{1 \le i \le n_u}$ are on a uniform grid on $[-10,10]$ and the $(t^{(i)})_{1 \le i \le n_t}$ on $[0,T]$. The sample set is the Cartesian product of the three previous ones, and has $n = n_t n_x n_u$ elements. We also use the same samples as initial points $(t_0^{(i)}, x_0^{(i)})$ in the objective function of problem (P). Note that we have replaced it with $\sum_{i=1}^n V(t^{(i)}, x^{(i)})/n$, as we found it more efficient in our experiments to optimize over V at intermediate time steps rather than at t_0 only. Indeed, we ultimately evaluate our approximation by the accuracy of V on the whole $\mathcal{X}_T$ and not only on $\{t_0\} \times \mathcal{X}$. In a discrete states and actions setting, this effect is analyzed by De Farias and Van Roy (2003), where μ_0 is denoted as "state-relevance weights".

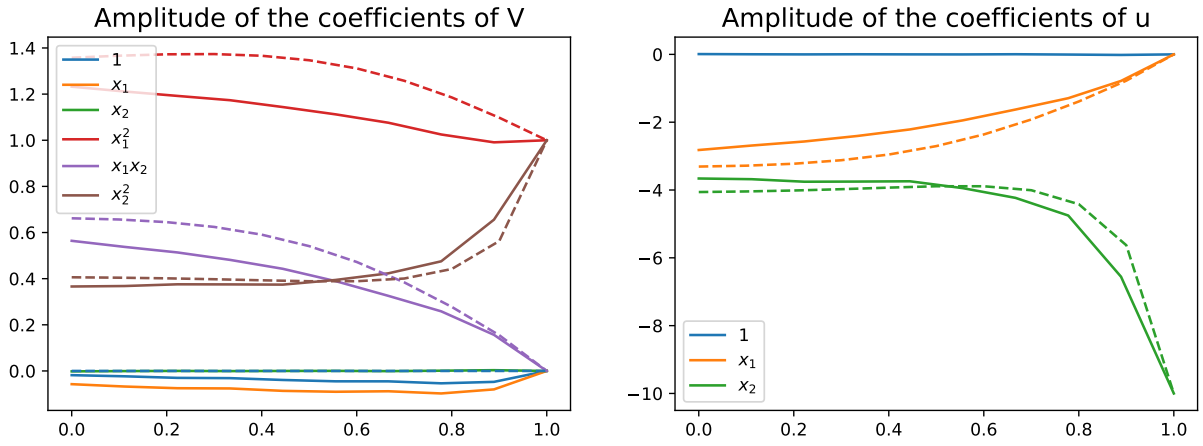


Figure 5.1: Variation through time of the coefficients of the value function in the basis of monomials of degree less than two, and of the corresponding controller. The plain lines denote the results obtained with the kernel SoS method, for $n_x = n_u = 10$, $\lambda_\beta = 10^{-2}$, $\lambda_B = 0.1$, $\gamma = 10^{-2}$, $\varepsilon = 10^{-4}$. The dotted lines are the coefficients of the true value function V^*.

Methods. We compare three methods: the LP, the guided SoS and the kernel SoS. The LP method is detailed in Section 5.3.1, and as for the kernel SoS method, we add a slackness parameter on the constraints, with a penalization controlled by $\gamma > 0$ ($\gamma \to \infty$ recovers the original LP).

The guided SoS method is the same as problem (5.36), except that the embeddings Φ_i of the samples are replaced by vectors Ψ_i of fixed dimension, which are computed explicitly, without a kernel. Motivated by the fact that:

$$H^*(t,x,u) = (u + K(t)x)^\top R(u + K(t)x), \tag{5.47}$$

we choose the embedding vectors as follows:

$$\Psi_i := \left(\frac{u^{(i)}}{10},\ x^{(i)},\ \left(\omega^{-1} \sin\left(\frac{\omega\pi}{2}(t^{(i)}/T - 1) \right) x^{(i)} \right)_{1 \le \omega \le q_t} \right)^\top, \tag{5.48}$$

where the last q_t scalar terms (without the vector x) approximately model $K(.)$ as a smooth function of t. For computational efficiency, we choose $q_t = 5$ and we checked that this basis can approximate the entries of $K(t)$ well. Then we solve an SDP of size $(p + q_t d) \times (p + q_t d) = 11 \times 11$ instead of $n \times n$ for the kernel version.

The kernel SoS method is as described in the previous sections, with the following kernel:

$$k((t,x,u),(t',x',u')) = \langle u, u' \rangle / 100 + \langle x, x' \rangle \times \exp(-|t - t'|). \tag{5.49}$$

This kernel is also designed to match the shape of H^*, with a smooth term in t modelled by the exponential kernel. The matrix K can be singular, so we replace it by $K + 10^{-8} I_n$.

Results. We compare the performance of the three methods to a baseline: the projection of V^* on $\mathcal{F}_\Theta$, which is a proxy for the best performance to expect with a fixed $\mathcal{F}_\Theta$. We keep the best set of hyper-parameters

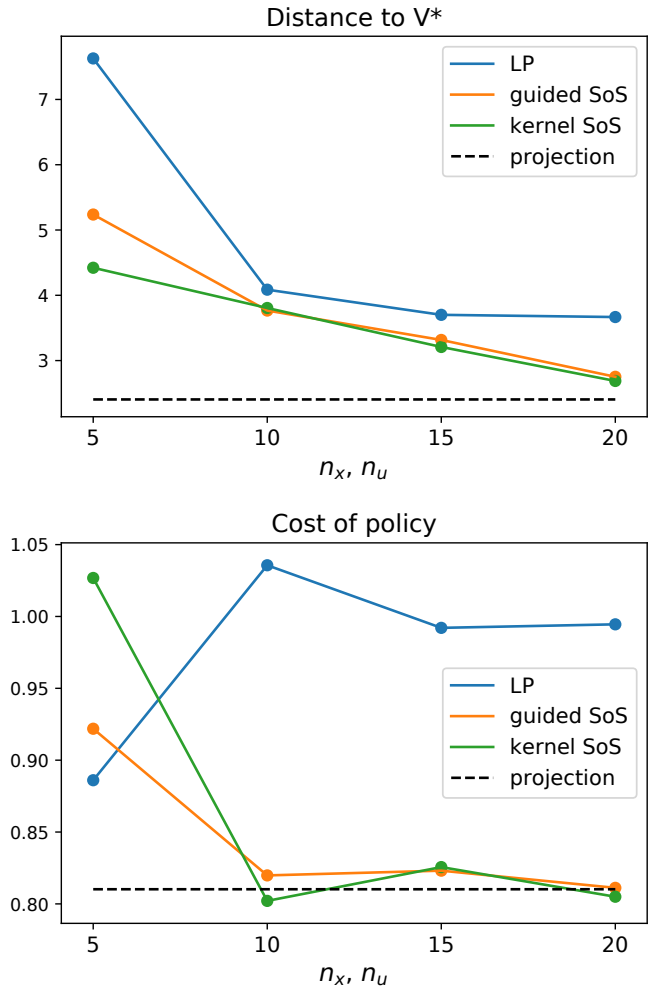


Figure 5.2: Comparison of the performances of the value function and the policy of the three methods, as a function of the number of samples $n_x = n_u$.

after a grid search on (λ_θ, γ) for the LP method, and on $(\lambda_\theta, \gamma, \lambda)$ for the two others, with $\varepsilon = 10^{-4}$. We keep $\eta = 0$ and $n_t = 20$ in all the experiments, and a varying number $n_x = n_u$ of sample points. For example, with $n_x = n_u = 20$, the dual variable α of the largest problem here has dimension $n = 8000$, and solving the numerical problem written in Python takes a few minutes on a standard laptop. In Figure 5.1, we plot the value function and controller obtained with $n_x = n_u = 10$ points.

The results are presented in Figure 5.2. The guided and kernel SoS methods perform similar, and better than the LP: they better exploit a fixed number of samples than the LP. Note that the kernel SoS tends to the LP when λ tends to 0, hence using a positive λ improves the results. We believe that the design of a kernel adapted to prior knowledge on the problem is crucial to benefit from this effect. Finally, the kernel SoS has the same performance as the guided SoS, but it is computationally more expensive as soon as $n > 11$. Yet the kernel version extends way beyond such fixed finite-dimensional embeddings, to infinite-dimensional embeddings represented by any positive definite kernel, including the exponential kernel, the polynomial kernel and many others.

5.7 Conclusion

The kernel SoS approximation method generalizes the polynomial SoS method for OCPs. Like the simple LP method, it is *black-box*, or sample-based, in the sense that it is based only on function evaluations of the dynamics and loss, without requiring to compute any gradients. Moreover, it enables to exploit prior knowledge on the structure of an OCP, by choosing an appropriate kernel. The problem reduces to an SDP, whose size can be computationally limiting, but parallel implementations are possible. There are several sources of approximations in this method: the parameterization $\mathcal{F}_\Theta$ of V might not be exact, the SoS representation of H^* is not exact in general (although we have proved it is in a few particular cases), and we subsample a finite number of constraints. In particular, estimating the effect of subsampling in such a way that the method gives a certified lower-bound on the OCP, like in the method of Lasserre et al. (2008), could be addressed in future work by extending the technique of Woodworth et al. (2022). For all these reasons, the method will probably not reach high precision solutions, but can be used to initialize direct shooting methods, and returns an approximate solution even with very few samples. Furthermore, we believe it is possible to extend the method to also account for state constraints, similarly to Lasserre et al. (2008). One could also parameterize the value function directly in an RKHS. Another interesting extension is to apply the method to Markov decision processes, where we could deal with states that are more complex objects (graphs, sequences, trajectories,...), or even infinite-dimensional objects like in soft robotics (Della Santina et al., 2021), with appropriate kernels.

Chapter 6

A Non-asymptotic Analysis of Non-parametric Temporal-Difference Learning

Abstract. *Temporal-difference learning is a popular algorithm for policy evaluation. In this chapter, we study the convergence of the regularized non-parametric TD(0) algorithm, in both the independent and Markovian observation settings. In particular, when TD is performed in a universal reproducing kernel Hilbert space (RKHS), we prove convergence of the averaged iterates to the optimal value function, even when it does not belong to the RKHS. We provide explicit convergence rates that depend on a source condition relating the regularity of the optimal value function to the RKHS. We illustrate this convergence numerically on a simple continuous-state Markov reward process.*

This chapter is based on our work *A Non-asymptotic Analysis of Non-parametric Temporal-Difference Learning*, with Ziad Kobeissi and Francis Bach, accepted for publication in the Conference on Neural Information Processing Systems (NeurIPS), 2022.

Contents

6.1 Introduction

One of the main ingredients of reinforcement learning (RL) is the ability to estimate the long-term effect on future rewards of employing a given policy. This building block, known as policy evaluation, already contains crucial features of more complex RL algorithms, such as SARSA or Q-learning (Sutton and Barto, 2018). Temporal-difference learning (TD), proposed by Sutton (1988), is among the simplest algorithms for policy evaluation. The estimation of the performance of the policy is made through a value function. It is updated *online*, after each new observation of a couple composed of a state transition and a reward.

Although the formulation of TD is quite natural, its theoretical analysis has proved more challenging, as it combines two difficulties. The first one is that TD *bootstraps*, in the sense that it uses its previous – possibly inaccurate – predictions to correct its next predictions, because it does not have access to a fixed ground truth. The second difficulty is that the observations are produced along a trajectory following a fixed policy (*on-policy*), hence they are correlated, which calls for more involved stochastic approximation tools compared to independent identically distributed (*i.i.d.*) samples. Moreover, using *off-policy* samples, produced by a different policy than the one being evaluated, can make the algorithm diverge (Boyan and Moore, 1994). Off-policy sampling is out of our scope in this chapter.

A third element which is not inherent to TD further complicates the plot: function approximation. While TD was originally proposed in a tabular setting, its large-scale applicability has been greatly extended by its combination with parametric function approximation (Bradtke and Barto, 1996). This enables the use of any linear or non-linear function approximation method to model the value function, including neural networks. However, one can exhibit unstable diverging behaviors of TD even with simple non-linear approximation schemes (Tsitsiklis and Van Roy, 1997). This combination of difficulties (even with linear function approximation) has been coined the "deadly triad" by Sutton (2015). We argue that convergence can be obtained even with non-linear function approximation, by making use of the non-parametric formalism of reproducing kernel Hilbert spaces (RKHS), involving linear approximation in infinite-dimension. Studying this case could bring us closer to understanding what happens with other universal approximators used in practice, like neural networks, in the wake of the recent study of TD with a one-hidden layer neural network by Cai et al. (2019).

6.1.1 Contributions

We study the policy evaluation algorithm TD(0) in the non-parametric case, first when the observations are sampled *i.i.d.* from the invariant distribution of the Markov chain resulting from the evaluated policy, and then when they are collected from a trajectory of the Markov chain with geometric mixing. In that sense we follow a similar outline as the analysis of Bhandari et al. (2018) which is dedicated to the linear case.

The non-parametric formulation of TD closes the gap between the original tabular formulation and the parametric formulation which involves semi-gradients. It allows the use of classical tools and theory from kernel methods (Cristianini and Shawe-Taylor, 2004). In particular, we highlight the central role of infinite-dimensional covariance operators (Bach, 2022; Baker, 1973) which already appear in the analysis of other non-parametric algorithms, like least-squares regression. We study a regularized variant of TD, a widely used way of dealing with misspecification in regression. Importantly, when the regularized TD approximation is run on an infinite-dimensional RKHS which is dense in the space of square-integrable functions, then there is no approximation error and the algorithm converges to the true value function. More precisely, we provide a proof of convergence in expectation of TD without approximation error, even when the true value function does not belong to the RKHS, under a weaker source condition. Furthermore, we give non-asymptotic convergence rates related to this source condition, which measures the regularity of the true value function relative to the RKHS, *e.g.*, its smoothness if the RKHS is a Sobolev space (Novak et al., 2018).

Note that using a universal kernel (Micchelli et al., 2006) to obtain convergence of TD to the true value function is also interesting from a theoretical point of view. Indeed it exempts us from a possibly tedious study of the approximation (or projection) error on a given basis, and simply removes an error term which in general scales linearly with the horizon of the Markov reward process (Mou et al., 2020; Yu and Bertsekas, 2010).

In the rest of this section, we review the related literature. In Section 6.2, we present the algorithm, along with generic results and notations. In Section 6.3, we analyze a simplified version of the algorithm, namely population TD in continuous time. This allows to catch the main features of the analysis, while postponing the technicalities related to stochastic approximation. Section 6.4 is dedicated to the analysis of non-parametric TD with *i.i.d.* observations, while Section 6.5 consists in a similar analysis for correlated observations sampled from a geometrically mixing Markov chain. Finally, in Section 6.6, we present simple numerical simulations illustrating the convergence results and the role of the main parameters.

6.1.2 Related Literature

Temporal-difference learning. The TD algorithm was introduced in its tabular version by Sutton (1988), with a first convergence result for linearly independent features, later extended to dependent features by Dayan (1992). Further stochastic approximation results were proposed by Jaakkola et al. (1993) for the tabular case, and by Schapire and Warmuth (1996) for the linear approximation case. An exact analysis of the behavior of tabular TD was recently carried out by Hu and Syed (2019), using the framework of Markov jump linear systems. Tsitsiklis and Van Roy (1997) provided a thorough asymptotic analysis of TD with linear function approximation, while failure cases were already known (Baird, 1995). A non-asymptotic analysis was later proposed by Lakshminarayanan and Szepesvari (2018) in the *i.i.d.* sampling case with constant step size, concurrently to another approach extending to Markov sampling by Bhandari et al. (2018). Other problem-dependent bounds for linear TD were derived around the same period (Dalal et al., 2018; Srikant and Ying, 2019), along with an analysis of variance-reduced TD (Korda and La, 2015; Xu et al., 2020). All of the

analyses mentioned above focus either on the tabular or on the linear *parametric* TD algorithm. A recent work by Duan et al. (2021) deals with the batch counterpart of non-parametric TD, namely the least-squares TD algorithm (LSTD), but they rather focus on the analysis of the statistical estimation error. Importantly, LSTD only requires offline computations and is not related to stochastic approximation. Most closely related to our work is the non-parametric regularized TD setting studied by Koppel et al. (2020). However, their analysis is limited to the case where the optimal value function belongs to the RKHS. This is not sufficient to get rid of the approximation error term. Also, we will show later that regularization is not necessary in this case. Furthermore, their analysis is restricted to the *i.i.d.* setting, for which we will require fewer regularity assumptions. Finally, let us mention the recent work by Cai et al. (2019) concerning TD with function approximation using a one-hidden layer neural network with finite-width, called "neural TD". Since finite-width neural networks are not universal approximators, there is an approximation error, which vanishes in the infinite-width limit if the value function belongs to a particular function space.

Kernel methods in RL. To tackle large-dimensional problems, kernel methods have been combined with various RL algorithms, including approximate dynamic programming (Barreto et al., 2011; Bhat et al., 2012; Grünewälder et al., 2012; Ormoneit and Sen, 2002), policy evaluation (Dai et al., 2017), policy iteration (Farahmand et al., 2016), LSTD (Duan et al., 2021), the linear programming formulation of RL (Dietterich and Wang, 2001), upper confidence bound (Domingues et al., 2021), or fitted Q-iteration (Long et al., 2021). Such kernel methods often come along with practical ways to reduce the computational complexity that grows with the number of observed transitions and rewards (Barreto et al., 2016; Koppel et al., 2020).

Stochastic approximation. The analysis of TD requires tools from stochastic approximation (Benveniste et al., 1990), among which the ODE method (Borkar and Meyn, 2000). Such tools are primarily designed for finite-dimensional problems. Stochastic gradient descent (SGD) (Bottou et al., 2018) is a specific instance of stochastic approximation that has received extensive attention for supervised learning. In particular, the role of regularization of SGD for least-squares regression has been studied (Caponnetto and De Vito, 2007; Cucker and Zhou, 2007), as well as the effect of of sampling data from a Markov chain (Nagaraj et al., 2020). Finally, we use a formalism which is close to the analyses by Berthier et al. (2020); Dieuleveut and Bach (2016); Pillaud-Vivien et al. (2018a) of non-parametric SGD for least squares regression.

6.2 Problem Formulation and Generic Results

6.2.1 The Non-parametric TD(0) Algorithm

We consider a Markov reward process (MRP), *i.e.*, a Markov chain with a reward associated to each state. This is what results from keeping the policy fixed in a Markov decision process (MDP) for policy evaluation. We consider MRPs in discrete-time, not necessarily with a countable state space $\mathcal{X}$. Specifically, we use the formalism of Markov chains on a measurable state space, also called Harris chains, which unifies discrete- and continuous-state Markov chains. Formally, let $\mathcal{X} \subset \mathbb{R}^d$ a measurable set associated with the σ-algebra $\mathcal{A}$ of Lebesgue measurable sets. Let $(x_n)_{n\geq 1}$ a time-homogeneous Markov chain with Markov kernel κ. A Markov kernel (Klenke, 2013; Reiss, 2012) is a mapping $\kappa : \mathcal{X} \times \mathcal{A} \to [0, 1]$ that has the following two properties:

1. for every $x \in \mathcal{X}$, $\kappa(x, \cdot)$ is a probability measure on $\mathcal{A}$;

2. for every $A \in \mathcal{A}$, $\kappa(\cdot, A)$ is $\mathcal{A}$-measurable.

If $\mathcal{X}$ is a countable set, κ is represented by a transition matrix Q such that $Q_{i,j} := \mathbb{P}(j|i) = \kappa(i, \{j\})$, for any $i, j \in \mathcal{X}$.

We define a reward function $r : \mathcal{X} \to \mathbb{R}$ uniformly bounded by $R < \infty$, and a discount factor $\gamma \in [0, 1)$. The aim of policy evaluation is to compute the value function of the MRP:

$$\forall x \in \mathcal{X}, \quad V^*(x) = \mathbb{E}\Big[\sum_{n=0}^{+\infty} \gamma^n r(x_n) \;\Big|\; x_0 = x\Big], \tag{6.1}$$

where the $(x_n)_{n\geq 1}$ are drawn from the Markov chain. A probability distribution $p : \mathcal{A} \to \mathbb{R}$ is a stationary distribution for κ if for all $A \in \mathcal{A}$,

$$p(A) = \int_{\mathcal{X}} \kappa(x, A) p(\mathrm{d}x). \tag{6.2}$$

The existence and uniqueness of a stationary distribution p, along with the convergence of the Markov chain to p in total variation, is ensured by ergodicity conditions. A sufficient condition is that the Markov chain is Harris ergodic, *i.e.*, it has a regeneration set, and is aperiodic and positively recurrent (see Asmussen (2003) and Durrett (2019) for an exposition of Harris chains). For discrete-state Markov chains, ergodicity conditions can be expressed somewhat more simply, and any aperiodic and positive recurrent Markov chain has a unique invariant distribution. Throughout this chapter, we assume that p is the unique invariant distribution of the Markov chain, and that it has full support on $\mathcal{X}$. Only in Section 6.5, we will in addition assume that the Markov chain is geometrically mixing.

We define $L^2(p)$, the set of squared integrable functions $f : \mathcal{X} \to \mathbb{R}$ with respect to p, with the norm

$$\|f\|^2_{L^2(p)} = \int_{\mathcal{X}} f(x)^2 p(\mathrm{d}x) < +\infty. \tag{6.3}$$

We also consider a reproducing kernel Hilbert space $\mathcal{H}$ of $\mathcal{A}$-measurable functions, associated to a positive-definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For all $x \in \mathcal{X}$, we use the notation $\Phi(x) := K(x, \cdot)$ for the mapping of x in $\mathcal{H}$, and $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ for the inner product in $\mathcal{H}$ (we sometimes drop the index). We assume that $M_{\mathcal{H}} := \sup_{x \in \mathcal{X}} K(x, x)$ is finite, which implies that $\mathcal{H} \subset L^2(p)$. More precisely, the $\mathcal{H}$-norm controls the $L^2(p)$-norm: any sequence converging in $\mathcal{H}$ thus converges in $L^2(p)$. Indeed, if $f \in \mathcal{H}$:

$$\|f\|^2_{L^2(p)} = \int f(x)^2 \mathrm{d}p(x) = \int \langle f, \Phi(x) \rangle^2_{\mathcal{H}} \mathrm{d}p(x) \leq \|f\|^2_{\mathcal{H}} \int \|\Phi(x)\|^2_{\mathcal{H}} \mathrm{d}p(x) \leq M_{\mathcal{H}} \|f\|^2_{\mathcal{H}}. \tag{6.4}$$

We also assume that $r \in L^2(p)$. The non-parametric TD(0) algorithm in the RKHS $\mathcal{H}$ is defined as follows (Koppel et al., 2020; Ormoneit and Sen, 2002). Draw a sequence $(x_n)_{n\geq 0}$ according to the Markov chain with initial distribution p, and collect the corresponding rewards $(r(x_n))_{n\geq 0}$. Define a sequence of non-negative step sizes $(\rho_n)_{n\geq 1}$. We build recursively a sequence of approximate value functions $(V_n)_{n\geq 0}$ in $L^2(p)$. Throughout the chapter, we take $V_0 = 0$ for simplicity, but note that all the results can be adapted to the case $V_0 \in \mathcal{H}$. For $n \geq 1$:

$$\forall y \in \mathcal{X}, \quad V_n(y) = V_{n-1}(y) + \rho_n \Big[r(x_n) + \gamma V_{n-1}(x'_n) - V_{n-1}(x_n) \Big] K(x_n, y), \tag{6.5}$$

where $x'_n := x_{n+1}$. The term in brackets is called a temporal-difference. Equivalently, in the RKHS:

$$V_n = V_{n-1} + \rho_n \Big[r(x_n) + \gamma V_{n-1}(x'_n) - V_{n-1}(x_n) \Big] \Phi(x_n). \tag{6.6}$$

This update has a running time complexity of $O(n^2)$, which can be improved to $O(n)$, *e.g.*, using Nyström approximation or random features (Halko et al., 2011). More details on the implementation are given in Appendix 6.B.2. This non-parametric formulation is a natural extension of the tabular TD algorithm. Indeed, if $\mathcal{X}$ is a countable set and $K(x,y) = \mathbf{1}_{x=y}$ is a Dirac kernel – a valid positive-definite kernel – then we exactly recover tabular TD: the update rule (6.5) becomes, after observing a transition $(i, i', r_i) := (x_n, x'_n, r(x_n))$:

$$V_n(i) = V_{n-1}(i) + \rho_n \Big[r_i + \gamma V_{n-1}(i') - V_{n-1}(i) \Big], \quad \text{and } \forall j \neq i,\ V_n(j) = V_{n-1}(j). \tag{6.7}$$

This also covers the *semi-gradient* formulation of TD for linear function approximation (Sutton and Barto, 2018). Suppose $\mathcal{H}$ has finite dimension d, then V_n can be identified to $\xi_n \in \mathbb{R}^d$, and we are searching for an approximation of the form $V_n(x) = \xi_n^\top \Phi(x)$. Then (6.6) becomes:

$$\xi_n = \xi_{n-1} + \rho_n \Big[r(x_n) + \gamma V_{n-1}(x'_n) - V_{n-1}(x_n) \Big] \nabla_\xi V_n(x_n). \tag{6.8}$$

Since $V_0 \in \mathcal{H}$, all the iterates V_n are in the RKHS, in particular $V_n \in \operatorname{span}\{\Phi(x_k)\}_{1 \leq k \leq n}$. Consequently, if the sequence (V_n) converges in the topology induced by the $L^2(p)$-norm, it converges in $\overline{\mathcal{H}}$, the closure of $\mathcal{H}$ with respect to the $L^2(p)$-norm. In particular, for a dense RKHS and because p has full support on $\mathcal{X}$, $\overline{\mathcal{H}} = L^2(p)$, but in general it only holds that $\overline{\mathcal{H}} \subset L^2(p)$.

To understand the behavior of the algorithm, we will first consider the *population* version (also called *mean-path* by Bhandari et al. (2018)) of the algorithm: set $V_0 = 0$ and for $n \geq 1$:

$$V_n = V_{n-1} + \rho_n \mathbb{E}_{(x,x') \sim q} \left[\left(r(x) + \gamma V_{n-1}(x') - V_{n-1}(x) \right) \Phi(x) \right], \tag{6.9}$$

where the expectation is taken with respect to $q(\mathrm{d}x, \mathrm{d}x') := p(\mathrm{d}x)\kappa(x, \mathrm{d}x')$. Since $V_{n-1} \in \mathcal{H}$, the reproducing property holds: $V_{n-1}(x) = \langle V_{n-1}, \Phi(x) \rangle_{\mathcal{H}}$. Hence the update is affine and reads:

$$V_n = V_{n-1} + \rho_n (A V_{n-1} + b), \tag{6.10}$$

with $A := \mathbb{E}_q \left[\gamma \Phi(x) \otimes \Phi(x') - \Phi(x) \otimes \Phi(x) \right]$ and $b := \mathbb{E}_p \left[r(x) \Phi(x) \right]$, where $\otimes$ denotes the outer product in $\mathcal{H}$ defined by $g \otimes h : f \mapsto \langle f, h \rangle_{\mathcal{H}} g$.

6.2.2 Covariance Operators

Assume that the expectations $\Sigma := \mathbb{E}_p[\Phi(x) \otimes \Phi(x)]$ and $\Sigma_1 := \mathbb{E}_q[\Phi(x) \otimes \Phi(x')]$ are well-defined. Σ and Σ_1 are the uncentered auto-covariance operators of order 0 and 1 of the Markov process $(x_n)_{n \geq 1}$, under the invariant distribution p. They are operators from $\mathcal{H}$ to $\mathcal{H}$, such that, for all $f, g \in \mathcal{H}$, using the reproducing property:

$$\begin{aligned} \mathbb{E}_p[f(x)g(x)] &= \mathbb{E}_p[\langle f, \Phi(x) \rangle_{\mathcal{H}} \langle g, \Phi(x) \rangle_{\mathcal{H}}] = \langle f, \mathbb{E}_p[\langle g, \Phi(x) \rangle_{\mathcal{H}} \Phi(x)] \rangle_{\mathcal{H}} = \langle f, \Sigma g \rangle_{\mathcal{H}} \\ \mathbb{E}_q[f(x)g(x')] &= \mathbb{E}_q[\langle f, \Phi(x) \rangle_{\mathcal{H}} \langle g, \Phi(x') \rangle_{\mathcal{H}}] = \langle f, \mathbb{E}_p[\langle g, \Phi(x') \rangle_{\mathcal{H}} \Phi(x)] \rangle_{\mathcal{H}} = \langle f, \Sigma_1 g \rangle_{\mathcal{H}}. \end{aligned} \tag{6.11}$$

In particular, for all $y \in \mathcal{X}$ and $f \in \mathcal{H}$, $(\Sigma f)(y) = \langle \Phi(y), \Sigma f \rangle_{\mathcal{H}} = \mathbb{E}_p[f(x)K(x,y)]$ and similarly, $(\Sigma_1 f)(y) = \mathbb{E}_q[f(x')K(x,y)]$. Following Dieuleveut and Bach (2016), Σ and Σ_1 can therefore be extended to operators Σ^e and Σ_1^e from $L^2(p)$ to $L^2(p)$ defined by:

$$\begin{aligned} \Sigma^e &: f \mapsto \int_{\mathcal{X}} f(x) \Phi(x) p(\mathrm{d}x), \text{ such that } \forall y \in \mathcal{X},\ (\Sigma^e f)(y) = \mathbb{E}_p[f(x)K(x,y)] \\ \Sigma_1^e &: f \mapsto \iint_{\mathcal{X}^2} f(x') \Phi(x) q(\mathrm{d}x, \mathrm{d}x'), \text{ such that } \forall y \in \mathcal{X},\ (\Sigma_1^e f)(y) = \mathbb{E}_q[f(x')K(x,y)]. \end{aligned} \tag{6.12}$$

These two operators are the building blocks of the TD iteration (6.9). In particular, $A = \gamma\Sigma_1 - \Sigma$ and $b = \Sigma^e r$, the latter being valid for $r \in L^2(p)$. With a slight abuse of notation, we denote simply as Σ, Σ_1 the extended operators. Furthermore, $\mathrm{Im}(\Sigma) \subset \mathcal{H}$ and $\Sigma^{1/2}$ is an isometry from $L^2(p)$ to $\mathcal{H}$ (Dieuleveut and Bach, 2016):

$$\forall f \in \overline{\mathcal{H}}, \quad \|f\|_{L^2(p)} = \|\Sigma^{1/2} f\|_{\mathcal{H}}. \tag{6.13}$$

The fact that p is a stationary distribution for κ implies a particular constraint linking Σ and Σ_1:

Lemma 1. *There exists a unique bounded linear operator $\tilde{\Sigma}_1 : \mathcal{H} \to \mathcal{H}$ such that $\Sigma_1 = \Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}$ on $\overline{\mathcal{H}}$, and $\|\tilde{\Sigma}_1\|_{\mathrm{op}} \leq 1$ ($\|\cdot\|_{\mathrm{op}}$ is the $\mathcal{H}$-operator norm).*

This results from an application of (Baker, 1973, Theorem 1), valid on $\mathcal{H}$ and extended by continuity to $\overline{\mathcal{H}}$. See also Fukumizu et al. (2004) for an exposition of cross-covariance operators specifically in an RKHS. In finite dimension, this is retrieved with generic results on positive semi-definite (PSD) matrices. Specifically, if $\mathcal{H} \subset \mathbb{R}^m$, the uncentered covariance matrix of the random variable $(\Phi(x), \Phi(x'))$, when $(x, x') \sim q$ is:

$$\begin{pmatrix} \Sigma & \Sigma_1 \\ \Sigma_1^\top & \Sigma \end{pmatrix} \succeq 0. \tag{6.14}$$

Using a classical condition on block matrices (Bhatia, 2013, Proposition 1.3.2), this matrix is PSD if and only if there exists a matrix $\tilde{\Sigma}_1$ such that $\|\tilde{\Sigma}_1\|_{\mathrm{op}} \leq 1$ and $\Sigma_1 = \Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}$ ($\|\cdot\|_{\mathrm{op}}$ is also the spectral norm in this case). This corresponds to the fact that the Schur complement of a PSD block matrix is also PSD.

Assumptions on Σ and V^*. We assume that $x \mapsto K(x, x)$ is uniformly bounded by $M_{\mathcal{H}}$. Therefore, the eigenvalues of Σ are upper-bounded. However, unlike Tsitsiklis and Van Roy (1997) and Bhandari et al. (2018), we do not assume them to be lower-bounded, *i.e.*, $\Sigma \succeq 0$ is not invertible in general. We will formulate our convergence results for two sets of assumptions. The first one recovers known results from Bhandari et al. (2018) for linear function approximation. The second one assumes that V^* verifies a *source condition* (Dieuleveut, 2017, Chapter 1):

(A1) $V^* \in \mathcal{H}$, $\mathcal{H}$ is finite-dimensional and Σ has full-rank;

(A2) $V^* \in \Sigma^{\theta/2}(\mathcal{H})$ for some $\theta \in (-1, 1]$ (and consequently, $\|\Sigma^{-\theta/2}V^*\|_{\mathcal{H}} < +\infty$), and $\overline{\mathcal{H}} = L^2(p)$ (*i.e.*, K is a universal kernel).

In **(A1)**, $\mathcal{H}$ is finite-dimensional because Σ cannot be simultaneously compact ($x \mapsto K(x, x)$ being uniformly bounded) and invertible in infinite-dimension (Cheney, 2001). Recalling the isometry property (6.13), the case $\theta = -1$ always holds in **(A2)** because $V^* \in L^2(p)$ (which we prove in the next subsection). The case $\theta = 0$ is equivalent to $V^* \in \mathcal{H}$. For $\theta > 0$, it must be interpreted as: $\|\Sigma^{-\theta/2}V^*\|^2_{\mathcal{H}} := \inf\{\|V\|^2_{\mathcal{H}} \mid V \text{ s.t. } V^* = \Sigma^{\theta/2}V\}$, with $\|\Sigma^{-\theta/2}V^*\|_{\mathcal{H}} = +\infty$ if $V^* \notin \Sigma^{\theta/2}(\mathcal{H})$. Using a universal approximation removes the need for a projection operator on $\overline{\mathcal{H}}$, as typically used for finite-dimensional function approximation, and hence there will be no projection error (Tsitsiklis and Van Roy, 1997).

6.2.3 Non-Expansiveness of the Bellman Operator

It is known that the value function V^* of the MRP is a fixed point of the Bellman operator T. We define two operators P and $T : L^2(p) \to L^2(p)$ by, for $V \in L^2(p)$, $PV(x) = \mathbb{E}_{x' \sim \kappa(x,\cdot)}V(x')$ and $TV(x) =$

$r(x) + \gamma PV(x)$. Both operators can be expressed in terms of Σ and Σ_1. For $V \in L^2(p)$:

$$\begin{cases} \Sigma PV = \mathbb{E}_p[\Phi(x)(PV)(x)] = \mathbb{E}_q[\Phi(x)V(x')] = \Sigma_1 V \\ \Sigma TV = \Sigma r + \gamma \Sigma_1 V. \end{cases} \tag{6.15}$$

Lemma 2. *For any $V \in L^2(p)$: $\|PV\|_{L^2(p)} \leq \|V\|_{L^2(p)}$.*

This is a direct reformulation of (Tsitsiklis and Van Roy, 1997, Lemma 1), the proof of which is given in Appendix 6.A.1. As stressed by Tsitsiklis and Van Roy (1997), this strongly relies on the fact that p is a stationary distribution of the Markov chain. It implies that T is a γ-contraction mapping on $L^2(p)$ and has as unique fixed point V^*. One can check that if Σ is non-singular, Lemma 2 is exactly equivalent to $\|\Sigma^{-1/2}\Sigma_1\Sigma^{-1/2}\|_{\text{op}} \leq 1$, that is, Lemma 1. Moreover, using Lemma 2, we obtain

$$\|V^*\|_{L^2(p)} \leq \|r\|_{L^2(p)}/(1-\gamma), \tag{6.16}$$

and $V^* \in L^2(p)$.

6.3 Analysis of a Continuous-Time Version of the Population TD Algorithm

Before considering regularized TD with stochastic samples, we look at simplified versions of the algorithm that momentarily remove the difficulties related to stochastic approximation. Specifically, we consider the population version of TD to capture a "mean" behavior, and a continuous-time algorithm to avoid choosing step sizes. Instead, we focus on the role of the regularization parameter.

6.3.1 Existence of a Fixed Point for Regularized TD

For $\lambda \geq 0$, let us consider the regularized population recursion:

$$V_n = V_{n-1} + \rho_n(\Sigma r + (\gamma\Sigma_1 - \Sigma - \lambda I)V_{n-1}). \tag{6.17}$$

If the TD iterations converge, their limit will be a solution of the *regularized* fixed-point equation:

$$\Sigma r + (\gamma\Sigma_1 - \Sigma - \lambda I)V = 0. \tag{6.18}$$

Proposition 1. *If $\lambda > 0$, $\gamma\Sigma_1 - \Sigma - \lambda I$ is non-singular on $\mathcal{H}$ and equation (6.18) admits a unique solution V_λ^* in $L^2(p)$, defined by $V_\lambda^* = (\gamma\Sigma_1 - \Sigma - \lambda I)^{-1}\Sigma r$. Furthermore, $V_\lambda^* \in \mathcal{H}$ and:*

$$\|V_\lambda^*\|_{\mathcal{H}} \leq \frac{\|\Sigma r\|_{\mathcal{H}}}{\lambda} \leq \frac{\sqrt{M_{\mathcal{H}}}\|r\|_{L^2(p)}}{\lambda}. \tag{6.19}$$

The proof is in Appendix 6.A.2. Hence, for $\lambda > 0$, the $\mathcal{H}$-norm of V_λ^* is always bounded, unlike $\|V^*\|_{\mathcal{H}}$.

6.3.2 Convergence of the Regularized Fixed Point to the Optimal Value Function

Recalling that $V^* \in L^2(p)$, it satisfies the relation $TV^* = V^*$, implying that $\Sigma TV^* = \Sigma V^*$, *i.e.*,

$$\Sigma r + (\gamma\Sigma_1 - \Sigma)V^* = 0. \tag{6.20}$$

This *unregularized* fixed point equation possibly has other solutions, but if K is a universal kernel, as assumed by **(A2)**, then Σ is injective (Steinwart, 2001) and V^* is the unique solution. Let us recall that **(A2)** does not imply that V^* has a bounded $\mathcal{H}$-norm. However, we can control the $L^2(p)$-norm of $V_\lambda^* - V^*$ when λ is small using the *source condition* **(A2)**.

Proposition 2. *Assume that $\lambda > 0$ and assumption (A2). Then:*

$$\|V_\lambda^* - V^*\|_{L^2(p)}^2 \leq \frac{\lambda^{\theta+1}}{(1-\gamma)^2} \|\Sigma^{-\theta/2} V^*\|_{\mathcal{H}}^2. \tag{6.21}$$

The proof in Appendix 6.A.2 is inspired by similar results (Caponnetto and De Vito, 2007; Cucker and Zhou, 2007) in the context of ridge regression (recovered for $\gamma = 0$). Note that only $\|V_\lambda^* - V^*\|_{L^2(p)}$ is controlled, not $\|V_\lambda^* - V^*\|_{\mathcal{H}}$. Consequently, we obtain the convergence of V_λ^* to V^* in $L^2(p)$-norm when $\lambda \to 0$: the higher θ is, the faster the rate of convergence.

For universal Mercer kernels (Cucker and Smale, 2002), if we drop the source condition **(A2)**, using only the fact that $V^* \in L^2(p)$ – corresponding to $\theta = -1$ in **(A2)** – we can still prove that V_λ^* converges to V^* in $L^2(p)$-norm when $\lambda \to 0$, but without an explicit rate. We recall that a Mercer kernel is a continuous kernel over a compact set (Dieuleveut and Bach, 2016).

Corollary 1. *Assume that K is a universal Mercer kernel, and that $V^* \in L^2(p)$ (which holds as soon as $r \in L^2(p)$, see Section 6.2.3), then:*

$$\|V_\lambda^* - V^*\|_{L^2(p)} \xrightarrow[\lambda \to 0^+]{} 0\,. \tag{6.22}$$

The proof is given in Appendix 6.A.2.

6.3.3 Convergence of Continuous-Time Population TD

Following the ordinary differential equation (ODE) method (Borkar and Meyn, 2000), we study the continuous-time counterpart of the population iteration (6.17). At least formally, this consists in defining $\widetilde{V}_t = V_{n(t)}$ for t and $n(t)$ satisfying $t = \sum_{i=1}^{n(t)} \rho_i$, and letting ρ_i tend to 0 for any $i \geq 1$, where $V_{n(t)}$ is defined by recursion using (6.17). With a slight abuse of notation, we use the notation V_t instead of $\widetilde{V}_t$. We then obtain the following ODE in $\mathcal{H}$: $V_0 = 0$ and for $t \geq 0$:

$$\frac{\mathrm{d}V_t}{\mathrm{d}t} = (A - \lambda I)V_t + b. \tag{6.23}$$

We can exhibit a Lyapunov function for this dynamical system, see (Slotine and Li, 1991). This implies that V_t converges to V_λ^* when t tends to infinity, where V_λ^* is defined in Proposition 1. More precisely, for $\beta \in \{-1, 0\}$, we define W^β, the Lyapunov function, by $W^\beta(t) := \|\Sigma^{-\beta/2}(V_t - V_\lambda^*)\|_{\mathcal{H}}^2$ (please note that β's role in W^β is an index, not a power). $W^0(t)$ strictly decreases with t as follows:

Lemma 3 (Descent Lemma)**.** *For $\lambda > 0$, for all $t \geq 0$, the following holds:*

$$\frac{\mathrm{d}W^0(t)}{\mathrm{d}t} \leq -2(1-\gamma)W^{-1}(t) - 2\lambda W^0(t)\,. \tag{6.24}$$

The proof (see Appendix 6.A.2) mainly relies on the contraction property of the Bellman operator as expressed in Lemma 2. We can then deduce the convergence of the ODE (6.23) to V_λ^*.

Proposition 3. *Under assumption (A1), the solution V_t of the ODE (6.23) with $\lambda = 0$ is such that:*

$$\text{For } T > 0, \quad \|\overline{V}_T - V^*\|^2_{L^2(p)} \leq \frac{1}{2(1-\gamma)} \frac{\|V^*\|^2_{\mathcal{H}}}{T}, \tag{6.25}$$

where $\overline{V}_T$ is the Polyak-Ruppert average (Polyak and Juditsky, 1992) of V_t, defined by

$$\overline{V}_T := \frac{1}{T} \int_0^T V_t \mathrm{d}t. \tag{6.26}$$

Under assumption (A2), the solution V_t of the ODE (6.23) with $\lambda > 0$ is such that:

$$\text{For } T \geq 0, \quad \|V_T - V^*_\lambda\|^2_{\mathcal{H}} \leq \|V^*_\lambda\|^2_{\mathcal{H}} e^{-2\lambda T}. \tag{6.27}$$

Under (A1), we recover the same $O(1/T)$ convergence rate as Bhandari et al. (2018). We focus on (A2), where we get a fast convergence to V^*_λ in $\mathcal{H}$-norm (stronger than $L^2(p)$). However, we are rather interested in convergence to V^*. Proposition 2 quantifies how far V^*_λ is from V^*. Indeed, the error decomposes as:

$$\|V_T - V^*\|^2_{L^2(p)} \leq 2M_{\mathcal{H}} \|V_T - V^*_\lambda\|^2_{\mathcal{H}} + 2\|V^*_\lambda - V^*\|^2_{L^2(p)}. \tag{6.28}$$

Combining Propositions 1, 2, 3 shows a trade-off on λ:

$$\|V_T - V^*\|^2_{L^2(p)} = O\left(e^{-2\lambda T}/\lambda^2 + \lambda^{\theta+1}\right). \tag{6.29}$$

Taking $\lambda = (3+\theta)\log T/(2T)$ balances the terms up to logarithmic factors:

$$\|V_T - V^*\|^2_{L^2(p)} = \tilde{O}\left(T^{-1-\theta}\right), \tag{6.30}$$

where $\tilde{O}(g(n)) := O(g(n)\log(n)^\ell)$, for some $\ell \in \mathbb{R}$. In particular, for $\theta = 0$, *i.e.*, $V^* \in \mathcal{H}$, we recover a convergence rate $\tilde{O}(1/T)$: up to logarithmic factors, it is the same as the unregularized case with averaging, assuming (A1). In this case, regularization brings no benefits.

6.4 Stochastic TD with *i.i.d.* Sampling

We now consider stochastic TD iterations (6.6), where the couples $(x_n, x'_n)_{n\geq 1}$ are sampled *i.i.d.* from the distribution $q(\mathrm{d}x, \mathrm{d}x') = p(\mathrm{d}x)\kappa(x, \mathrm{d}x')$. Such *i.i.d.* samples can be obtained by running the Markov chain until it has mixed so that $x_n \sim p$, collecting a couple (x_n, x'_n), and restarting.

With $A_n := \gamma\Phi(x_n) \otimes \Phi(x'_n) - \Phi(x_n) \otimes \Phi(x_n)$ and $b_n := r(x_n)\Phi(x_n)$, we study the recursion:

$$V_n = V_{n-1} + \rho_n((A_n - \lambda I)V_{n-1} + b_n). \tag{6.31}$$

In particular, $\mathbb{E}_q[A_n] = A$, $\mathbb{E}_p[b_n] = b$, and A_n and b_n are independent of the past $(V_k)_{k<n}$.

For $\beta \in \{0, 1\}$, let $W^\beta_n := \|\Sigma^{-\beta/2}(V_n - V^*_\lambda)\|^2_{\mathcal{H}}$. Adapting the proof of Lemma 3, we exhibit a similar decreasing behavior of W^0_n in expectation, hence showing that $\mathbb{E}[\|V_n - V^*_\lambda\|^2_{\mathcal{H}}] \to 0$ for well-chosen step sizes ρ_n. Finally, λ is chosen to balance $\mathbb{E}[\|V_n - V^*_\lambda\|^2_{L^2(p)}]$ and $\|V^*_\lambda - V^*\|^2_{L^2(p)}$. We define $V^{(e)}_n$ and $V^{(t)}_n$ as the exponentially-weighted and the tail-averaged n-th iterates respectively:

$$V^{(e)}_n := \frac{\sum_{k=1}^n (1-\rho\lambda)^{n-k} V_{k-1}}{\sum_{k=1}^n (1-\rho\lambda)^{n-k}} \quad \text{and} \quad V^{(t)}_n := \frac{1}{n - \lfloor n/2 \rfloor + 1} \sum_{k=\lfloor n/2 \rfloor}^n V_{k-1}. \tag{6.32}$$

Theorem 7. *Let $n \geq 9$. Under assumption* **(A2)** *with $-1 < \theta \leq 1$, there exists a positive real number $\underline{\lambda}_\theta$ independent of n such that, for $\lambda_0 \geq \underline{\lambda}_\theta$,*

(a) Using $\lambda = \lambda_0 n^{-\frac{1}{3+\theta}}$ and a constant step size $\rho = \frac{\log n}{\lambda n}$, then:

$$\mathbb{E}\left[\|V_n - V^*\|^2_{L^2(p)}\right] = O\left((\log n) n^{-\frac{1+\theta}{3+\theta}}\right). \tag{6.33}$$

(b) Using $\lambda = \lambda_0 n^{-\frac{1}{2+\theta}}$ and a constant step size $\rho = \frac{\log n}{\lambda n}$, then:

$$\mathbb{E}\left[\|V_n^{(e)} - V^*\|^2_{L^2(p)}\right] = O\left((\log n) n^{-\frac{1+\theta}{2+\theta}}\right). \tag{6.34}$$

(c) Using $\lambda = \lambda_0 n^{-\frac{1}{2+\theta}}$ and a constant step size $\rho = \frac{2\log n}{\lambda n}$ for the first $\lfloor n/2 \rfloor - 1$ iterates and then a decreasing step size $\rho_k = \frac{1}{\lambda k}$, then:

$$\mathbb{E}\left[\|V_n^{(t)} - V^*\|^2_{L^2(p)}\right] = O\left((\log n) n^{-\frac{1+\theta}{2+\theta}}\right). \tag{6.35}$$

A similar exponentially-weighted averaging scheme as in (b) has been used by Défossez and Bach (2017) to study constant step size SGD. When $\gamma = 0$, the rates can be compared to existing results for SGD. For example, for $\theta \in [0, 1]$, Tarres and Yao (2014) prove almost sure convergence for regularized least-mean-squares without averaging at rate $O(n^{-\frac{1+\theta}{2+\theta}})$. The dependence in θ is similar to what we obtain. Moreover, under assumption **(A1)**, we recover the same convergence rate as Bhandari et al. (2018):

Proposition 4. *Under assumption* **(A1)**, *there exists an $n_0 > 0$ such that, for any $n \geq n_0$, using a constant step size $\rho = 1/\sqrt{n}$ and $\lambda = 0$, leads to:*

$$\mathbb{E}\|\overline{V}_n - V^*\|^2_{L^2(p)} \leq O(1/\sqrt{n}).$$

Finally, our bounds have a polynomial dependence in the horizon $1/(1-\gamma)$ of the MRP. The proofs are given in Appendix 6.A.3.

Remark. Because the Bellman operator is also a contraction mapping in L^∞-norm, this analysis in L^2-norm might be adapted to the L^∞-norm, using a modified Lyapunov function to study the ODE, *e.g.*, following Borkar and Soumyanatha (1997). The stochastic case could be handled using the smoothing technique recently developed by Chen et al. (2020).

6.5 Stochastic TD with Markovian Sampling

We now consider the truly *online* TD algorithm, where the samples are produced by a Markov chain. In particular, there is now a correlation between the current samples (x_n, x'_n) and the previous iterate V_{n-1}. To control it, we assume that the Markov chain mixes at uniform geometric rate:

$$\textbf{(A3)} \qquad \exists m > 0,\ \mu \in (0,1) \ \text{ s.t. } \sup_{x \in \mathcal{X}} d_{TV}\left(\mathbb{P}(x_n \in \cdot | x_0 = x), p\right) \leq m\mu^n, \tag{6.36}$$

where d_{TV} denotes the total variation distance. This is always verified for irreducible, aperiodic finite Markov chains (Levin and Peres, 2017). Note that the uniform mixing assumption might be relaxed by a weaker drift condition using the technique developed by Durmus et al. (2021) for linear TD, although its extension to the infinite-dimensional setting is not straightforward, and out of our scope. We give an example of continuous-state Markov chain with geometric mixing in Section 6.6. Furthermore, following Bhandari et al. (2018), in our analysis we need to control the magnitude of the iterates almost surely. To do so, we add a projection step at each TD iteration:

$$V_n = \Pi_B[V_{n-1} + \rho_n((A_n - \lambda I)V_{n-1} + b_n)], \tag{6.37}$$

where Π_B is the projection on the $\mathcal{H}$ ball of radius $B > 0$. If $\|V_\lambda^*\|_{\mathcal{H}} \leq B$, the convergence of the method is preserved. In the following theorem, we consider two regimes with different rates of convergence. In the first one, we assume like Bhandari et al. (2018) that we are given an oracle B upper-bounding $\|V_\lambda^*\|_{\mathcal{H}}$, with B independent of λ. In the second one, we use the bound of Proposition 1, but this will affect the convergence rate since in this case $B = O(1/\lambda)$.

Theorem 8. *Assuming* **(A2)** *and that the samples are produced by a Markov chain with uniform geometric mixing* **(A3)**, *the projected TD iterations (6.37) are such that:*

(i) Using $\lambda = n^{-\frac{1}{2+\theta}}$, a constant step size $\rho = \frac{\log n}{2\lambda n}$, and using a projection radius B independent of λ provided by an oracle and such that $\|V_\lambda^*\|_{\mathcal{H}} \leq B$, then:

$$\mathbb{E}\left[\|V_n^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left(\frac{(\log n)^2 n^{-\frac{1+\theta}{2+\theta}}}{\log(1/\mu)}\right). \tag{6.38}$$

(ii) Using $\lambda = n^{-\frac{1}{4+\theta}}$, $\rho = \frac{\log n}{2\lambda n}$, and the projection radius $B = \sqrt{M_{\mathcal{H}}}\|r\|_{L^2(p)}/\lambda$, then:

$$\mathbb{E}\left[\|V_n^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left(\frac{(\log n)^2 n^{-\frac{1+\theta}{4+\theta}}}{\log(1/\mu)}\right), \tag{6.39}$$

with $V_n^{(e)} = \sum_{k=1}^n (1-2\rho\lambda)^{n-k} V_{k-1} / \sum_{j=1}^n (1-2\rho\lambda)^{n-j}$.

When an oracle is given for B (*i.e.*, in setting (i)), we recover the same rate as for *i.i.d.* sampling, up to a multiplicative factor $\log(n)/\log(1/\mu)$ which represents the mixing time of the Markov chain. If no oracle is provided (*i.e.*, in setting (ii)), the convergence will be slower because the bound B is of order $1/\lambda$. Note that the slight changes in the definitions of ρ, λ, $V^{(e)}$, and the absence of constraint on λ, as compared to Theorem 7, are implied by the boundedness of the iterates. Following a similar study for SGD (Nagaraj et al., 2020), we might compare these rates to those of a naive algorithm which we call "τ-Skip-TD", for some $\tau \geq 1$, where only one every τ samples from the Markov chain is used to make TD updates:

$$V_n = \Pi_B[V_{n-1} + \rho_n((A_{n\tau} - \lambda I)V_{n-1} + b_{n\tau})], \tag{6.40}$$

For τ large enough, of the order of the mixing time of the Markov chain, the new sample $(x_{n\tau}, x'_{n\tau})$ is almost independent from the past ones $(x_{k\tau}, x'_{k\tau})_{k<n}$. Of course, since we need to generate τ times more samples to make a step, we must look at the distance of $V_{n/\tau}$ to the optimum. Convergence rates for τ-Skip-TD are derived in the following result:

Corollary 2. *Assuming (A2) and that the samples are produced by a Markov chain with uniform geometric mixing (A3), the projected τ-Skip-TD iterations (6.40) are such that:*

(i) Using $\lambda = n^{-\frac{1}{2+\theta}}$, a constant step size $\rho = \frac{\log n}{2\lambda n}$, $\tau = \lceil \frac{\log(1/\rho)}{\log(1/\mu)} + 1 \rceil$, and a projection radius B which is provided by an oracle and such that $\|V_\lambda^*\|_{\mathcal{H}} \leq B$, then the following inequality holds:

$$\mathbb{E}\left[\|V_{n/\tau}^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left(\frac{(\log n) n^{-\frac{1+\theta}{2+\theta}}}{\log(1/\mu)}\right). \tag{6.41}$$

(ii) Using $\lambda = n^{-\frac{1}{4+\theta}}$, $\rho = \frac{\log n}{2\lambda n}$, $\tau = \lceil \frac{\log(1/\rho)}{\log(1/\mu)} + 1 \rceil$, and the projection radius B of Proposition 1, then we obtain:

$$\mathbb{E}\left[\|V_{n/\tau}^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left((\log n) n^{-\frac{1+\theta}{4+\theta}}\right), \tag{6.42}$$

assuming that n is a multiple of τ, with $V_n^{(e)} = \sum_{k=1}^n (1-2\rho\lambda)^{n-k} V_{k-1} / \sum_{j=1}^n (1-2\rho\lambda)^{n-j}$.

In setting *(i)*, they are similar to Theorem 8 up to a $\log(n)$ factor. This suggests that making updates at each sample of the Markov chain is not more efficient than τ-Skip-TD for large τ, at least in our theoretical analysis. In practice, using all samples seems slightly better, especially for a slowly mixing Markov chain (see Section 6.6). In setting *(ii)*, we obtain a rate for Skip-TD whose leading term does not depend on $\log(1/\mu)$ – which only appears in higher order terms – suggesting that the rate and parameters of Theorem 8, setting *(ii)* might be suboptimal.

Remark. The analysis of TD with linear function approximation by Srikant and Ying (2019) does not require a projection step. Hence the necessity of the projection step might only be an artifact of our proof technique inspired by Bhandari et al. (2018). In the above experiments, we simply omit the projection step.

6.6 Experiments on Artificial Data

6.6.1 Setting of the Problem

Building a value function. We build a toy model for which the main parameters can be computed in closed form. We consider the dynamics on the circle $\mathcal{X} = [0, 1]$ defined by:

- with probability ε, $x_{n+1} \sim \mathcal{U}([0, 1])$,
- with probability $1 - \varepsilon$, $x_{n+1} = x_n$.

Because the Markov kernel is symmetric, the invariant distribution is $p = \mathcal{U}([0, 1])$. In particular, the mixing parameter can be bounded explicitly with $m = 1$ and $\mu = 1 - \varepsilon$ (see Appendix 6.B.1).

Also, simple computations show that V^* is an affine transform of r:

$$V^*(x) = ar(x) + b, \tag{6.43}$$

with $a = (1 - \gamma(1 - \varepsilon))^{-1}$ and $b = -a \int_0^1 r(u)du$. Hence we can build a V^* with a given regularity by choosing an appropriate reward with the same regularity. We consider two rewards: $r_{\text{abs}}(x) := 2|x - 1/2|$ and $r_{\cos}(x) := (1 + \cos(2\pi x))/2$.

Kernels on the torus. We consider the RKHS of splines on the circle (Wahba, 1990) of regularity $s \in \mathbb{N}^*$, denoted by H^s_{per}. It is a Sobolev space equipped with the following norm:

$$\|f\|^2_{H^s_{\text{per}}} = \left(\int_0^1 f(x)\mathrm{d}x\right)^2 + \frac{1}{(2\pi)^{2s}} \int_0^1 |f^{(s)}(x)|^2 \mathrm{d}x. \tag{6.44}$$

Its corresponding reproducing kernel K_s is a translation-invariant kernel defined by:

$$K_s(x,y) = 1 + (-1)^{s-1} \frac{(2\pi)^{2s}}{(2s)!} B_{2s}(\{x-y\}), \tag{6.45}$$

where $\{x\} := x - \lfloor x \rfloor$ and B_j is the j-th Bernoulli polynomial (Olver et al., 2010). Let us recall that the Fourier series expansion on the torus of a 1-periodic function $f \in L^2(p)$ is:

$$f(x) = \sum_{\omega \in \mathbb{Z}} e^{2i\omega\pi x} \hat{f}_\omega \ , \quad \text{with} \quad \hat{f}_\omega := \int_0^1 f(x) e^{-2i\omega\pi x} \mathrm{d}x \ , \tag{6.46}$$

for $\omega \in \mathbb{Z}$. The kernel K_s has an embedding in the space of Fourier coefficients:

$$\Phi(x) = (\sqrt{c_\omega} e^{2i\omega\pi x})^\top_{m \in \mathbb{Z}} \ , \tag{6.47}$$

with $c_\omega := |\omega|^{-2s}$ if $\omega \neq 0$ and $c_0 := 1$. Using Parseval's theorem in Eqn. (6.44), one can compute the norm of f from its Fourier coefficients:

$$\|f\|^2_{H^s_{\text{per}}} = \sum_{\omega \in \mathbb{Z}} |\hat{f}_\omega|^2 / c_\omega \ . \tag{6.48}$$

The operators Σ and Σ_1 can be represented as countably infinite-dimensional matrices $\Sigma = \mathrm{diag}(c)$ and $\Sigma_1 = (1-\varepsilon)\Sigma + \varepsilon\sqrt{c}(\sqrt{c})^\top$. Hence the source condition states that

$$|\hat{f}_0|^2 + \sum_{\omega \neq 0} |\omega|^{2s(1+\theta)} |\hat{f}_\omega|^2 < \infty \ . \tag{6.49}$$

In particular, it holds if $f \in H^{s'}_{\text{per}}$, for any $s' \geq s(1+\theta)$. In our example, we consider two Sobolev spaces H^1_{per} and H^2_{per}, and our two example functions have Fourier coefficients $(\hat{r}_{\text{abs}})_\omega = \frac{1-(-1)^\omega}{\pi^2\omega^2}$ for $\omega \neq 0$, and $(\hat{r}_{\cos})_\omega = 0$ for $|\omega| > 1$. The largest $\theta \in [0,1]$ such that the source condition holds are indicated in the first row of Table 6.1.

6.6.2 Qualitative and Quantitative Results

Convergence rates. We run TD on functions r_{abs} and $r_{\cos}$, with kernels K_1 and K_2. We use parameters λ and ρ and exponential averaging as prescribed in Theorem 7 *(b)*. Each experiment is repeated 10 times and we record the mean $\pm$ one standard deviation. The implementation is based on a finite dimensional representation of the iterates $(V_k)_{k \leq n}$ in $\mathbb{R}^n$ (see further details in Appendix 6.B.2). This implies computing the kernel matrix in $O(n^2)$ operations. To accelerate this computation when the eigenvalues decrease fast, we approximate it with the incomplete Cholesky decomposition (Bach and Jordan, 2002).

In Table 6.1, we set $\varepsilon = 0.8$, $\gamma = 0.5$ and report the observed convergence rates *v.s.* the ones expected by Theorem 8, which are fairly consistent. In Figure 6.1, we plot the obtained value functions with the two different kernels: the algorithm learns a smoother function when the kernel is itself smoother ($s = 2$).

In Figure 6.2, we show the respective effects of varying ε and γ. Larger values of ε or γ make the problem more difficult and slow down convergence, presumably in the constants without affecting the rates, as predicted by Theorem 8.

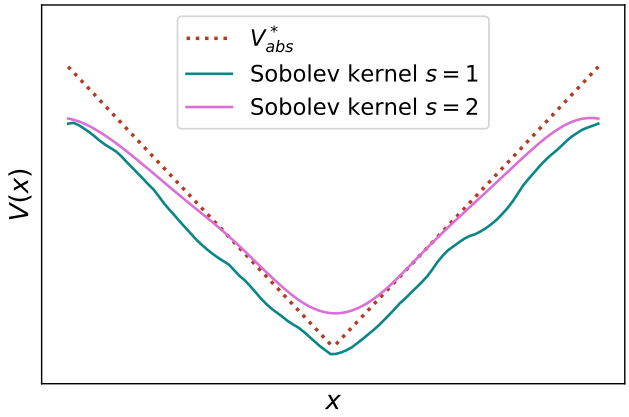


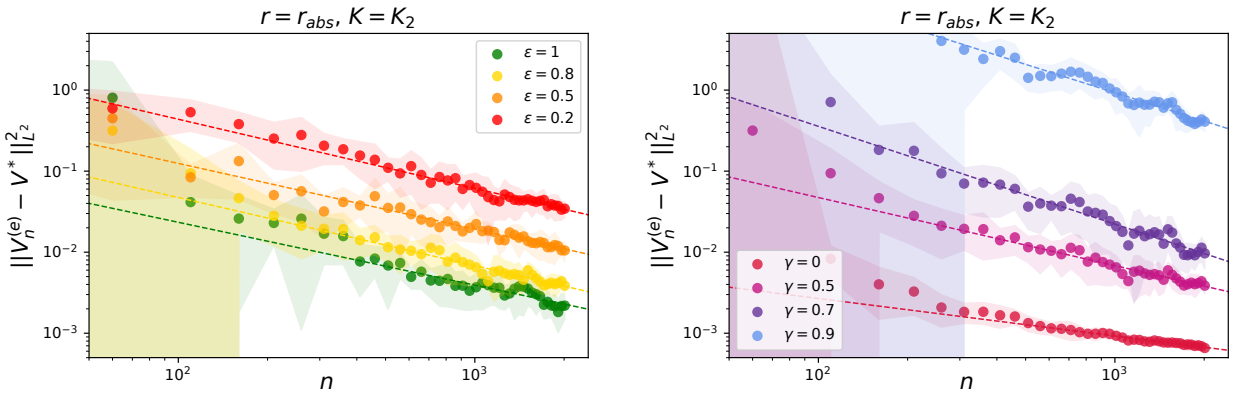
Figure 6.1: Approximate value functions obtained with $r = r_{\text{abs}}$, $n = 1000$, $\varepsilon = 0.8$, $\gamma = 0.5$, with the two different kernels, and using the values of θ from Table 6.1.

Table 6.1: Predicted and observed convergence rates with different reward functions and kernels.

	Sobolev kernel $s = 1$		Sobolev kernel $s = 2$	
	$r = r_{\text{abs}}$	$r = r_{\cos}$	$r = r_{\text{abs}}$	$r = r_{\cos}$
Maximal θ	$1/2$	1	$-1/4$	1
Predicted rate	-0.6	-0.67	-0.43	-0.67
Observed rate	-0.72	-0.64	-0.58	-0.64

Robustness to misspecification of θ. We test the robustness of TD to inexact estimations of θ, hence resulting in too large or too small λ. If θ is under-estimated, our theorems still guarantee convergence for $\theta > -1$, but not if it is over-estimated. In Figure 6.3, we plot the convergence of the averaged iterates for different values of θ, smaller or larger than the optimal $\theta = -1/4$ (standard deviations have been removed for readability). Figure 6.3 shows that the convergence is quite robust and gives similar results for $\theta = 0$ or $\theta = -1/2$. A strongly overestimated $\theta = 1$ shows a slow convergence (not covered by our theorems). However, as expected, with $\theta = -1$, the algorithm does not converge. Indeed, the corresponding step size is unbounded.

Comparison of TD and Skip-TD. Finally, we compare TD and τ-Skip-TD, with τ prescribed by Corollary 2. Computing this τ requires the access to an oracle on the mixing parameter μ ($\mu = 1 - \varepsilon$ in our example). We then use $\tau = \lceil \frac{\log(1/\rho)}{\log(1/\mu)} + 1 \rceil$. We compare the results of TD and τ-Skip-TD for two different values of ε. We expect similar convergence rates, but with different constants. The results are plotted in Figure 6.4. For the fast mixing chain ($\varepsilon = 0.8$), we get comparable results. For the slowly mixing chain ($\varepsilon = 0.2$), plain TD seems faster, although maybe the asymptotic regime has not been reached yet for $n = 2000$.



(a) Varying ε (for $\gamma = 0.5$ fixed).

(b) Varying γ (for $\varepsilon = 0.8$ fixed).

Figure 6.2: Respective effects of varying ε and γ.

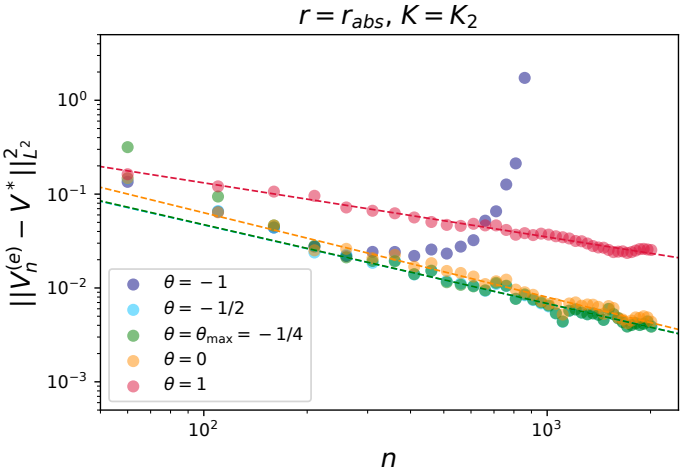


Figure 6.3: Convergence of the averaged TD iterates as in Theorem 7(b) with over and underestimated values of θ.

6.7 Conclusion

We have provided convergence rates for the regularized non-parametric TD algorithm in the *i.i.d.* and Markovian sampling settings. The rates depend on a source condition that quantifies the relative regularity of the optimal value function to the RKHS. They are compatible with our empirical observations on a one-dimensional MRP, but we have not proved optimality of such rates. Interesting directions include the extension to the TD(λ) algorithm, which we believe can be achieved with similar tools, as well as more challenging extensions to control counterparts of TD (Q-learning, SARSA,...) for which the policy is optimized.

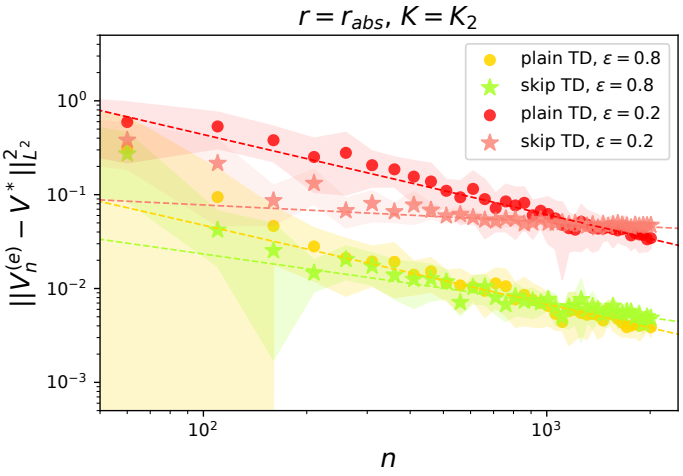


Figure 6.4: TD vs τ-Skip-TD with fast ($\varepsilon = 0.8$) and slowly ($\varepsilon = 0.2$) mixing Markov chains.

Appendix to Chapter 6

6.A Proofs and Intermediate Results

6.A.1 Problem Formulation and Generic Results

Proof of Lemma 2. Let $V \in L^2(p)$. Then:

$$\begin{aligned}
\|PV\|_{L^2(p)}^2 &= \int_{\mathcal{X}} (\mathbb{E}_{x' \sim \kappa(x,\cdot)} V(x'))^2 p(\mathrm{d}x) \\
&\leq \int_{\mathcal{X}} \mathbb{E}_{x' \sim \kappa(x,\cdot)} [V(x')^2] p(\mathrm{d}x) \\
&= \int_{\mathcal{X}} \left(\int_{\mathcal{X}} V(x')^2 \kappa(x, \mathrm{d}x') \right) p(\mathrm{d}x) \\
&= \int_{\mathcal{X}} V(x')^2 \left(\int_{\mathcal{X}} \kappa(x, \mathrm{d}x') p(\mathrm{d}x) \right) \\
&= \int_{\mathcal{X}} V(x')^2 p(\mathrm{d}x') \\
&= \|V\|_{L^2(p)}^2.
\end{aligned}$$

The second line is an application of Jensen's inequality, with equality if $\forall x, V(x')|x$ is constant almost surely (a.s.). The fourth line is an application of Fubini-Tonelli's theorem. The fifth line results from the stationarity of p with respect to κ, and $\kappa(\cdot, \mathrm{d}x')$ being $\mathcal{A}$-measurable. □

6.A.2 Analysis of a Continuous-Time Version of the Population TD Algorithm

Proposition 1 is a consequence of the following Lemma 4:

Lemma 4. *For $\lambda > 0$, the operator $\Sigma + \lambda I - \gamma\Sigma_1 : \mathcal{H} \to \mathcal{H}$ is bijective, and the operator norm of its inverse is bounded as follows:*

$$\|(\Sigma + \lambda I - \gamma\Sigma_1)^{-1}\|_{\text{op}} \leq \frac{1}{\lambda}.$$

Proof of Lemma 4. From Lemma 1, there exists $\tilde{\Sigma}_1$ with $\|\tilde{\Sigma}_1\|_{\text{op}} \leq 1$ such that $\Sigma_1 = \Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}$.

For $\lambda > 0$, $\Sigma + \lambda I \succ 0$, hence we have the following decomposition,

$$\Sigma + \lambda I - \gamma\Sigma_1 = (\Sigma + \lambda I)^{1/2}\left[I - \gamma(\Sigma + \lambda I)^{-1/2}\Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}\right](\Sigma + \lambda I)^{1/2}. \tag{6.50}$$

Since the operator norm is an induced norm, we deduce:

$$\begin{aligned}&\|(\Sigma + \lambda I)^{-1/2}\Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}\|_{\text{op}} \\ &\qquad \leq \|(\Sigma + \lambda I)^{-1/2}\Sigma^{1/2}\|_{\text{op}} \cdot \|\tilde{\Sigma}_1\|_{\text{op}} \cdot \|\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}\|_{\text{op}}.\end{aligned}$$

Furthermore, from $\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2} \preceq I$, we obtain:

$$\|\gamma(\Sigma + \lambda I)^{-1/2}\Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}\|_{\text{op}} \leq \gamma < 1.$$

We can then apply Theorem 5.11 from Weidmann (2012), showing that the term inside the brackets in Eqn. (6.50) is invertible, with inverse equal to:

$$\sum_{k=0}^{+\infty} \gamma^k [(\Sigma + \lambda I)^{-1/2}\Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}]^k. \tag{6.51}$$

Hence, $\Sigma + \lambda I - \gamma\Sigma_1$ is invertible, with inverse equal to:

$$(\Sigma + \lambda I)^{-1/2}\left[I - \gamma(\Sigma + \lambda I)^{-1/2}\Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}\right]^{-1}(\Sigma + \lambda I)^{-1/2}.$$

We will now upper-bound the operator norm of $(\gamma\Sigma_1 - \Sigma - \lambda I)^{-1}$. Let us take $f, g \in \mathcal{H}$ such that $g = (\lambda I + \Sigma - \gamma\Sigma_1)f$ and $\|g\|_{\mathcal{H}} = 1$, we get

$$\begin{aligned}1 &= \|(\lambda I + \Sigma - \gamma\Sigma_1)f\|_{\mathcal{H}}^2 \\ &= \lambda^2\|f\|_{\mathcal{H}}^2 + 2\lambda\langle f, \Sigma f\rangle_{\mathcal{H}} - \lambda\gamma\langle f, (\Sigma_1 + \Sigma_1^*)f\rangle_{\mathcal{H}} + \|(\Sigma - \gamma\Sigma_1)f\|_{\mathcal{H}}^2 \\ &\geq \lambda^2\|f\|_{\mathcal{H}}^2 + 2\lambda\langle f, \Sigma f\rangle_{\mathcal{H}} - \lambda\gamma\langle f, (\Sigma_1 + \Sigma_1^*)f\rangle_{\mathcal{H}}.\end{aligned}$$

Moreover, we have:

$$\begin{aligned}\langle f, \Sigma_1 f\rangle_{\mathcal{H}} &= \mathbb{E}_q[f(x)f(x')] \\ &\leq \mathbb{E}_q\left[\frac{f(x)^2}{2} + \frac{f(x')^2}{2}\right] \\ &= \mathbb{E}_{x\sim p}\left[\frac{f(x)^2}{2}\right] + \mathbb{E}_{x'\sim p}\left[\frac{f(x')^2}{2}\right] \\ &= \langle f, \Sigma f\rangle_{\mathcal{H}},\end{aligned} \tag{6.52}$$

because p is an invariant distribution. Similarly,

$$\langle f, \Sigma_1^* f\rangle_{\mathcal{H}} = \langle \Sigma_1 f, f\rangle_{\mathcal{H}} = \langle f, \Sigma_1 f\rangle_{\mathcal{H}} \leq \langle f, \Sigma f\rangle_{\mathcal{H}}.$$

Consequently, since $\gamma \leq 1$, we get $1 \geq \lambda^2 \|f\|^2 = \lambda^2 \|(\lambda I + \Sigma - \gamma\Sigma_1)^{-1} g\|^2_{\mathcal{H}}$. We conclude by using the definition of the operator norm, *i.e.*,

$$\|(\lambda I + \Sigma - \gamma\Sigma_1)^{-1}\|_{\text{op}} = \sup_{\|g\|_{\mathcal{H}}=1} \|(\lambda I + \Sigma - \gamma\Sigma_1)^{-1} g\|_{\mathcal{H}} \leq 1/\lambda.$$

□

Proof of Proposition 1. Consider the fixed point equation (6.18). Since $\lambda > 0$, it is equivalent to:

$$V = \frac{1}{\lambda}\left[\Sigma r + \gamma\Sigma_1 V - \Sigma V\right].$$

As a consequence, any solution of this equation is in $\mathcal{H}$. Using Lemma 4, it is unique and such that:

$$V = (\gamma\Sigma_1 - \Sigma - \lambda I)^{-1}\Sigma r.$$

□

Proof of Proposition 2. The fixed point equations verified by V_λ^* and V^* are respectively:

$$\Sigma r + (\gamma\Sigma_1 - \Sigma - \lambda I)V_\lambda^* = 0. \tag{6.53}$$
$$\Sigma r + (\gamma\Sigma_1 - \Sigma - \lambda I)V^* = -\lambda V^* \tag{6.54}$$

Let $\bar{V}^* := \Sigma^{1/2}V^*$, $\bar{V}_\lambda^* := \Sigma^{1/2}V_\lambda^*$, and $\bar{r} := \Sigma^{1/2}r$. Then $\bar{V}^*$, $\bar{V}_\lambda^*$ and $\bar{r}$ are all in $\mathcal{H}$. Using Lemma 1, there exists $\tilde{\Sigma}_1 : \mathcal{H} \to \mathcal{H}$ with $\|\tilde{\Sigma}_1\|_{\text{op}} \leq 1$ such that $\Sigma_1 = \Sigma^{1/2}\tilde{\Sigma}_1\Sigma^{1/2}$. Because of assumption **(A2)**, this equality holds on $\overline{\mathcal{H}} = L^2(p)$. In particular, $\Sigma^{1/2}\Sigma_1 V^* = \Sigma\tilde{\Sigma}_1\bar{V}^*$.

Left multiplying Eqns. (6.53) and (6.54) by $\Sigma^{1/2}$, we get:

$$\Sigma\bar{r} + (\gamma\Sigma\tilde{\Sigma}_1 - \Sigma - \lambda I)\bar{V}_\lambda^* = 0. \tag{6.55}$$
$$\Sigma\bar{r} + (\gamma\Sigma\tilde{\Sigma}_1 - \Sigma - \lambda I)\bar{V}^* = -\lambda\bar{V}^* \tag{6.56}$$

Subtracting Eqns. (6.55) and (6.56), we get:

$$(\Sigma + \lambda I - \gamma\Sigma\tilde{\Sigma}_1)(\bar{V}_\lambda^* - \bar{V}^*) = -\lambda\bar{V}^*. \tag{6.57}$$

Since $\Sigma + \lambda I \succ 0$, then:

$$(I - \gamma(\Sigma + \lambda I)^{-1}\Sigma\tilde{\Sigma}_1)(\bar{V}_\lambda^* - \bar{V}^*) = -\lambda(\Sigma + \lambda I)^{-1}\bar{V}^*.$$

Let $\tilde{\Sigma}_{1,\lambda} := (\Sigma + \lambda I)^{-1}\Sigma\tilde{\Sigma}_1$. Since $(\Sigma + \lambda I)^{-1}\Sigma \preceq I$, we know that $\|\gamma\tilde{\Sigma}_{1,\lambda}\|_{\text{op}} \leq \gamma < 1$. Hence $(I - \gamma\tilde{\Sigma}_{1,\lambda})$ is invertible and:

$$\bar{V}_\lambda^* - \bar{V}^* = -\lambda(I - \gamma\tilde{\Sigma}_{1,\lambda})^{-1}(\Sigma + \lambda I)^{-1}\bar{V}^*$$

$$= -\lambda \sum_{k=0}^{+\infty} \gamma^k \tilde{\Sigma}_{1,\lambda}^k (\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*.$$

Taking the $\mathcal{H}$-norm on both sides, and using the isometry property (6.13), valid on $\overline{\mathcal{H}} = L^2(p)$ since we are using a universal kernel:

$$\|\Sigma^{1/2}(V_\lambda^* - V^*)\|_{\mathcal{H}} \leq \lambda \sum_{k=0}^{+\infty} \gamma^k \|\tilde{\Sigma}_{1,\lambda}^k (\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*\|_{\mathcal{H}} \tag{6.58}$$

$$\|V_\lambda^* - V^*\|_{L^2(p)} \leq \lambda \sum_{k=0}^{+\infty} \gamma^k \|(\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*\|_{\mathcal{H}} \tag{6.59}$$

$$= \frac{\lambda}{1-\gamma} \|(\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*\|_{\mathcal{H}}. \tag{6.60}$$

Assuming that V^* verifies the source condition with constant θ, the norm on the right-hand side can be bounded as follows:

$$\begin{aligned}
\|(\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*\|_{\mathcal{H}} &= \|(\Sigma + \lambda I)^{-1} \Sigma^{(1+\theta)/2} \Sigma^{-\theta/2} V^*\|_{\mathcal{H}} \\
&= \|(\Sigma + \lambda I)^{(\theta-1)/2} (\Sigma + \lambda I)^{-(1+\theta)/2} \Sigma^{(1+\theta)/2} \Sigma^{-\theta/2} V^*\|_{\mathcal{H}} \\
&\leq \lambda^{(\theta-1)/2} \|(\Sigma + \lambda I)^{-(1+\theta)/2} \Sigma^{(1+\theta)/2} \Sigma^{-\theta/2} V^*\|_{\mathcal{H}},
\end{aligned}$$

because $0 \prec (\Sigma + \lambda I)^{(\theta-1)/2} \preceq \lambda^{(\theta-1)/2} I$, since $(\theta - 1)/2 \leq 0$. Also, since $(1 + \theta)/2 \geq 0$, we have: $(\Sigma + \lambda I)^{-(1+\theta)/2} \Sigma^{(1+\theta)/2} \preceq I$, hence:

$$\|(\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*\|_{\mathcal{H}} \leq \lambda^{(\theta-1)/2} \|\Sigma^{-\theta/2} V^*\|_{\mathcal{H}}. \tag{6.61}$$

Combining Eqns. (6.60) and (6.61), we can then conclude that

$$\|V_\lambda^* - V^*\|_{L^2(p)} \leq \frac{\lambda^{\frac{1+\theta}{2}}}{1-\gamma} \|\Sigma^{-\theta/2} V^*\|_{\mathcal{H}}.$$

□

Proof of Corollary 1. We can reproduce the beginning of the proof of Proposition 2, until Eqn. (6.60):

$$\|V_\lambda^* - V^*\|_{L^2(p)} \leq \frac{\lambda}{1-\gamma} \|(\Sigma + \lambda I)^{-1} \Sigma^{1/2} V^*\|_{\mathcal{H}}.$$

Using the isometry property (6.13) because K is a universal kernel:

$$\|V_\lambda^* - V^*\|_{L^2(p)} \leq \frac{\lambda}{1-\gamma} \|(\Sigma + \lambda I)^{-1} V^*\|_{L^2(p)}.$$

Because K is a Mercer kernel, there exists a sequence $(\psi_n)_{n \geq 1}$ in $L^2(p)$ which is an orthonormal eigenbasis of $\overline{\mathcal{H}} = L^2(p)$ (because K is universal) for the $L^2(p)$ inner product, with strictly positive eigenvalues $(\lambda_n)_{n \geq 1}$, ordered in decreasing order, such that (Dieuleveut and Bach, 2016):

$$\forall n \geq 1, \quad \Sigma \psi_n = \lambda_n \psi_n.$$

Then, since $V^* = \sum_{n\geq 1}\langle V^*, \psi_n\rangle_{L^2(p)}\psi_n$:

$$\begin{aligned}\|V_\lambda^* - V^*\|_{L^2(p)}^2 &\leq \frac{\lambda^2}{(1-\gamma)^2}\|(\Sigma+\lambda I)^{-1}V^*\|_{L^2(p)}^2 \\ &= \frac{1}{(1-\gamma)^2}\sum_{n\geq 1}\frac{\lambda^2}{(\lambda+\lambda_n)^2}\langle V^*, \psi_n\rangle_{L^2(p)}^2.\end{aligned}$$

For $\lambda > 0$, the series on the right-hand side is dominated by

$$\sum_{n\geq 1}\langle V^*, \psi_n\rangle_{L^2(p)}^2 = \|V\|_{L^2(p)}^2 < \infty,$$

and for each $n \geq 1$:

$$\frac{\lambda^2}{(\lambda+\lambda_n)^2}\langle V^*, \psi_n\rangle_{L^2(p)}^2 \xrightarrow[\lambda\to 0^+]{} 0,$$

because each λ_n is strictly positive. Then by Lebesgue's dominated convergence theorem (Rudin, 1987):

$$\|V_\lambda^* - V^*\|_{L^2(p)}^2 \xrightarrow[\lambda\to 0^+]{} 0.$$

□

Remark. This proof can be repeated to prove the same convergence rate as Proposition 2 in the case of Mercer kernels with assumption **(A2)**. Moreover, we can also obtain convergence rates in $\Sigma^{\tilde{\theta}/2}(\mathcal{H})$-norm (instead of the norm $L^2(p) = \Sigma^{-1/2}(\mathcal{H})$), for $\tilde{\theta} \in (-1, \theta)$.

Proof of Lemma 3. For $\beta = 0$, and $\lambda > 0$, $V_t - V_\lambda^* \in \Sigma^{0/2}(\mathcal{H}) = \mathcal{H}$ is always true as proved in Proposition 1, hence $W^0(t)$ is finite for all $t \geq 0$. Similarly, $W^{-1}(t)$ is finite for all $t \geq 0$ because V_t and $V_\lambda^* \in L^2(p)$.

$$\begin{aligned}\frac{\mathrm{d}W^0(t)}{\mathrm{d}t} &= 2\langle V_t - V_\lambda^*, \frac{\mathrm{d}V_t}{\mathrm{d}t}\rangle_{\mathcal{H}} \\ &= 2\langle V_t - V_\lambda^*, (A-\lambda I)V_t + b\rangle_{\mathcal{H}} \\ &= 2\langle V_t - V_\lambda^*, (\gamma\Sigma_1 - \Sigma - \lambda I)V_t) + \Sigma r\rangle_{\mathcal{H}}.\end{aligned}$$

Recalling that V_λ^* is a solution of Eqn. (6.18), we obtain:

$$\begin{aligned}\frac{\mathrm{d}W^0}{\mathrm{d}t} &= 2\langle V_t - V_\lambda^*, (\gamma\Sigma_1 - \Sigma - \lambda I)(V_t - V_\lambda^*)\rangle_{\mathcal{H}} \\ &= 2\gamma\langle V_t - V_\lambda^*, \Sigma_1(V_t - V_\lambda^*)\rangle_{\mathcal{H}} - 2\lambda\langle V_t - V_\lambda^*, V_t - V_\lambda^*\rangle_{\mathcal{H}} - 2\langle V_t - V_\lambda^*, \Sigma(V_t - V_\lambda^*)\rangle_{\mathcal{H}} \\ &= 2\gamma\langle V_t - V_\lambda^*, \Sigma P(V_t - V_\lambda^*)\rangle_{\mathcal{H}} - 2\lambda W^0(t) - 2W^{-1}(t) \\ &= 2\gamma\langle \Sigma^{1/2}(V_t - V_\lambda^*), \Sigma^{1/2}P(V_t - V^*)\rangle_{\mathcal{H}} - 2\lambda W^0(t) - 2W^{-1}(t),\end{aligned}$$

where the third line results from Eqn. (6.15). Using Cauchy-Schwarz inequality for $\langle\cdot,\cdot\rangle_{\mathcal{H}}$, the first term is bounded by:

$$2\gamma\langle \Sigma^{1/2}(V_t - V_\lambda^*), \Sigma^{1/2}P(V_t - V^*)\rangle_{\mathcal{H}} \leq 2\gamma\|\Sigma^{1/2}(V_t - V_\lambda^*)\|_{\mathcal{H}} \cdot \|\Sigma^{1/2}P(V_t - V_\lambda^*)\|_{\mathcal{H}}$$

$$
\begin{aligned}
&= 2\gamma\sqrt{W^{-1}(t)} \cdot \|P(V_t - V_\lambda^*)\|_{L^2(p)} \\
&\leq 2\gamma\sqrt{W^{-1}(t)} \cdot \|V_t - V_\lambda^*\|_{L^2(p)} \\
&= 2\gamma W^{-1}(t),
\end{aligned}
$$

where we have used successively Eqn. (6.13) (on an element of $\mathcal{H}$) and Lemma 2. Note that the same result could have been obtained directly from applying (6.52) to $\langle V_t - V_\lambda^*, \Sigma_1(V_t - V_\lambda^*)\rangle_{\mathcal{H}}$.

Finally, we get:

$$
\frac{\mathrm{d}W^0(t)}{\mathrm{d}t} \leq 2\gamma W^{-1}(t) - 2\lambda W^0(t) - 2W^{-1}(t),
$$

where all of the above quantities are finite.

□

Proof of Proposition 3. Let us split the proof in two parts, each one corresponding to a different assumption.

- Under assumption **(A1)**, we define the sequence of Polyak-Ruppert averaged iterates:

$$
\overline{V}_t := \frac{1}{t}\int_0^t V(s)\mathrm{d}s, \quad \text{for } t \geq 0.
$$

If $\Sigma \succ 0$ and $V^* \in \mathcal{H}$, Lemma 3 holds for $\lambda = 0$ and the proof is similar, *i.e.*,

$$
\frac{\mathrm{d}\|V_t - V^*\|_{\mathcal{H}}^2}{\mathrm{d}t} \leq -2(1-\gamma)\|V_t - V^*\|_{L^2(p)}^2.
$$

Let $T > 0$, we integrate between 0 and T and divide by T, noting that all quantities are finite because $\|V^*\|_{\mathcal{H}}$ is finite:

$$
\frac{W^0(T) - W^0(0)}{T} \leq -2(1-\gamma)\frac{1}{T}\int_0^T \|V_t - V^*\|_{L^2(p)}^2\mathrm{d}t.
$$

$$
\frac{1}{T}\int_0^T \|V_t - V^*\|_{L^2(p)}^2\mathrm{d}t \leq \frac{1/2}{1-\gamma}\frac{W^0(0) - W^0(T)}{T} \leq \frac{1/2}{1-\gamma}\frac{W^0(0)}{T}.
$$

This and Jensen's inequality imply:

$$
\|\overline{V}_T - V^*\|_{L^2(p)}^2 \leq \frac{1}{T}\int_0^T \|V_t - V^*\|_{L^2(p)}^2\mathrm{d}t,
$$

and then:

$$
\|\overline{V}_T - V^*\|_{L^2(p)}^2 \leq \frac{1}{2(1-\gamma)}\frac{\|V^*\|_{\mathcal{H}}^2}{T}.
$$

- Under assumption **(A2)**, Lemma 3 gives:

$$
\begin{aligned}
\frac{\mathrm{d}\|V_t - V_\lambda^*\|_{\mathcal{H}}^2}{\mathrm{d}t} &\leq -2(1-\gamma)\|V_t - V_\lambda^*\|_{L^2(p)}^2 - 2\lambda\|V_t - V_\lambda^*\|_{\mathcal{H}}^2 \\
&\leq -2\lambda\|V_t - V_\lambda^*\|_{\mathcal{H}}^2.
\end{aligned}
$$

Using Grönwall's lemma, we directly get the linear convergence of V_t to V_λ^* in $\mathcal{H}$ norm:

$$
\|V_t - V_\lambda^*\|_{\mathcal{H}}^2 \leq \|V_\lambda^*\|_{\mathcal{H}}^2 e^{-2t\lambda}.
$$

□

6.A.3 Stochastic TD with *i.i.d.* Sampling

First, we need to state a technical lemma which will be used several times:

Lemma 5. *For any fixed $V \in L^2(p)$, and $n \geq 1$, the following inequality holds:*

$$\mathbb{E}_q \|A_n V\|_{\mathcal{H}}^2 \leq 2M_{\mathcal{H}}(1+\gamma^2)\|\Sigma^{1/2}V\|_{\mathcal{H}}^2.$$

Proof of Lemma 5. Let us recall that since $(x_n, x_n') \sim q$, the marginals of x_n and x_n' are the same, *i.e.*, $x_n \sim p$ and $x_n' \sim p$. In addition, for a fixed realization of (x_n, x_n'), the operator A_n must be interpreted as the extended operator acting on $L^2(p)$, in a similar way as in Eqn. (6.12). In particular, for $V \in L^2(p)$, $A_n V = \gamma V(x_n')\Phi(x_n) - V(x_n)\Phi(x_n) \in \mathcal{H}$. Therefore we get:

$$\begin{aligned}
\mathbb{E}_q \|A_n V\|_{\mathcal{H}}^2 &= \mathbb{E}_q \|(\gamma V(x_n') - V(x_n))\Phi(x_n)\|_{\mathcal{H}}^2 \\
&= \mathbb{E}_q \left[|\gamma V(x_n') - V(x_n)|^2 \|\Phi(x_n)\|_{\mathcal{H}}^2\right] \\
&\leq 2M_{\mathcal{H}}(\gamma^2 \mathbb{E}_p[V(x_n')^2] + \mathbb{E}_p[V(x_n)^2]) \\
&= 2M_{\mathcal{H}}(1+\gamma^2)\|V\|_{L^2(p)}^2,
\end{aligned}$$

which concludes the proof. □

We now derive the stochastic equivalent of the Descent Lemma 3.

Lemma 6. *Let $\sigma^2 := 10M_{\mathcal{H}}\|r\|_{L^2(p)}^2 + \left(\frac{8(1+\gamma^2)}{(1-\gamma)^2} + 16(1+\gamma^2)\right) M_{\mathcal{H}}\|V^*\|_{L^2(p)}^2$. The following inequality holds for $n \geq 1$:*

$$\mathbb{E}W_n^0 \leq (1 - 2\rho_n\lambda + 2\rho_n^2\lambda^2)\mathbb{E}W_{n-1}^0 - \left(2\rho_n(1-\gamma) - 8\rho_n^2(1+\gamma^2)M_{\mathcal{H}}\right)\mathbb{E}W_{n-1}^{-1} + 4\rho_n^2\sigma^2.$$

In particular, for $\rho_n \leq \min\left\{\frac{1}{2\lambda}, \frac{1-\gamma}{8M_{\mathcal{H}}(1+\gamma^2)} =: \bar{\rho}\right\}$, we obtain:

$$\mathbb{E}W_n^0 \leq (1 - \rho_n\lambda)\mathbb{E}W_{n-1}^0 - \rho_n(1-\gamma)\mathbb{E}W_{n-1}^{-1} + 4\rho_n^2\sigma^2.$$

Proof of Lemma 6. Almost surely, the following decomposition holds:

$$\begin{aligned}
W_n^0 &= \langle V_n - V_\lambda^*, V_n - V_\lambda^*\rangle_{\mathcal{H}} \\
&= \langle V_{n-1} + \rho_n((A_n - \lambda I)V_{n-1} + b_n) - V_\lambda^*, V_{n-1} + \rho_n((A_n - \lambda I)V_{n-1} + b_n) - V_\lambda^*\rangle_{\mathcal{H}} \\
&= \langle V_{n-1} - V_\lambda^*, V_{n-1} - V_\lambda^*\rangle_{\mathcal{H}} + 2\rho_n\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}} \\
&\qquad + \rho_n^2\|(A_n - \lambda I)V_{n-1} + b_n\|_{\mathcal{H}}^2.
\end{aligned}$$

Let $z_i := (x_i, x_i')$, for $i \geq 1$. The z_i are *i.i.d.* with probability distribution q. Taking the expectation in the latter equality with respect to the filtration $\mathcal{F}_n := \sigma(z_1, ..., z_n)$, the resulting quantity may be decomposed into three parts as follows:

$$\begin{aligned}
\mathbb{E}W_n^0 = \mathbb{E}W_{n-1}^0 &+ 2\rho_n\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}}\right] \\
&+ \rho_n^2\mathbb{E}\left[\|(A_n - \lambda I)V_{n-1} + b_n\|_{\mathcal{H}}^2\right].
\end{aligned}$$

• The second term – the inner product – may be treated as follows:

$$\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}}\right] = \mathbb{E}\left[\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}}|\mathcal{F}_{n-1}\right]\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A - \lambda I)V_{n-1} + b\rangle_{\mathcal{H}}\right] \\
&\leq -(1-\gamma)\mathbb{E}W_{n-1}^{-1} - \lambda \mathbb{E}W_{n-1}^{0},
\end{aligned}
$$

where the last line is obtained using a similar argument as in the proof of Lemma 3, *i.e.*,

$$
\langle V - V_\lambda^*, (A - \lambda I)V + b\rangle_{\mathcal{H}} \leq -(1-\gamma)\|V - V_\lambda^*\|_{L^2(p)}^2 - \lambda \|V - V_\lambda^*\|_{\mathcal{H}}^2.
$$

• The third term – the variance term – can be upper-bounded as follows:

$$
\begin{aligned}
\mathbb{E}\left[\|(A_n - \lambda I)V_{n-1} + b_n\|_{\mathcal{H}}^2\right] &\leq 2\mathbb{E}\left[\|\lambda(V_{n-1} - V_\lambda^*)\|_{\mathcal{H}}^2\right] + 2\mathbb{E}\left[\|A_n V_{n-1} + b_n - \lambda V_\lambda^*\|_{\mathcal{H}}^2\right] \\
&\leq 2\lambda^2 \mathbb{E}W_{n-1}^0 + 4\mathbb{E}\left[\|A_n(V_{n-1} - V_\lambda^*)\|_{\mathcal{H}}^2\right] \\
&\qquad + 4\mathbb{E}\left[\|(A_n - \lambda I)V_\lambda^* + b_n\|_{\mathcal{H}}^2\right] \\
&\leq 2\lambda^2 \mathbb{E}W_{n-1}^0 + 4\mathbb{E}\left[\mathbb{E}\left[\|A_n(V_{n-1} - V_\lambda^*)\|_{\mathcal{H}}^2 | \mathcal{F}_{n-1}\right]\right] \\
&\qquad + 4\mathbb{E}\left[\|(A_n - \lambda I)V_\lambda^* + b_n\|_{\mathcal{H}}^2\right] \\
&\leq 2\lambda^2 \mathbb{E}W_{n-1}^0 + 8M_{\mathcal{H}}(1+\gamma^2)\mathbb{E}W_{n-1}^{-1} \\
&\qquad + 4\mathbb{E}\left[\|(A_n - \lambda I)V_\lambda^* + b_n\|_{\mathcal{H}}^2\right],
\end{aligned}
$$

where the last inequality comes from applying Lemma 5 to $V_{n-1} - V_\lambda^*$, since V_{n-1} is $\mathcal{F}_{n-1}$-measurable.

Let us now consider the remaining term $\mathbb{E}\left[\|(A_n - \lambda I)V_\lambda^* + b_n\|_{\mathcal{H}}^2\right]$, and prove that it is bounded by σ^2. This is the variance of the updates at the optimum. Using Proposition 1, we get:

$$
\begin{aligned}
\mathbb{E}\left[\|(A_n - \lambda I)V_\lambda^* + b_n\|_{\mathcal{H}}^2\right] &\leq 2\lambda^2 \|V_\lambda^*\|_{\mathcal{H}}^2 + 2\mathbb{E}\left[\|A_n V_\lambda^* + b_n\|_{\mathcal{H}}^2\right] \\
&\leq 2M_{\mathcal{H}}\|r\|_{L^2(p)}^2 + 2\mathbb{E}\left[\|A_n V_\lambda^* + b_n\|_{\mathcal{H}}^2\right],
\end{aligned}
$$

where:

$$
\begin{aligned}
2\mathbb{E}\left[\|A_n V_\lambda^* + b_n\|_{\mathcal{H}}^2\right] &\leq 4\mathbb{E}\left[\|A_n(V_\lambda^* - V^*)\|_{\mathcal{H}}^2\right] + 4\mathbb{E}\left[\|A_n V^* + b_n\|_{\mathcal{H}}^2\right] \\
&\leq 8M_{\mathcal{H}}(1+\gamma^2)\|\Sigma^{1/2}(V_\lambda^* - V^*)\|_{\mathcal{H}}^2 + 4\mathbb{E}\left[\|A_n V^* + b_n\|_{\mathcal{H}}^2\right],
\end{aligned}
$$

applying Lemma 5 to $V_\lambda^* - V^*$. Then, using Proposition 2 with $\theta = -1$ (which always holds):

$$
\begin{aligned}
2\mathbb{E}\left[\|A_n V_\lambda^* + b_n\|_{\mathcal{H}}^2\right] &\leq \frac{8M_{\mathcal{H}}(1+\gamma^2)\|V^*\|_{L^2(p)}^2}{(1-\gamma)^2} + 4\mathbb{E}\left[\|A_n V^* + b_n\|_{\mathcal{H}}^2\right] \\
&\leq \frac{8M_{\mathcal{H}}(1+\gamma^2)\|V^*\|_{L^2(p)}^2}{(1-\gamma)^2} + 8\mathbb{E}\left[\|A_n V^*\|_{\mathcal{H}}^2\right] + 8\mathbb{E}\left[\|b_n\|_{\mathcal{H}}^2\right] \\
&\leq \frac{8M_{\mathcal{H}}(1+\gamma^2)\|V^*\|_{L^2(p)}^2}{(1-\gamma)^2} + 16M_{\mathcal{H}}(1+\gamma^2)\|V^*\|_{L^2(p)}^2 \\
&\qquad + 8M_{\mathcal{H}}\|r\|_{L^2(p)}^2,
\end{aligned}
$$

where we have used again Lemma 5 applied to V^*, and the fact that:

$$
\mathbb{E}[\|b_n\|_{\mathcal{H}}^2] = \mathbb{E}[r(x_n)^2\|\Phi(x_n)\|_{\mathcal{H}}^2] \leq M_{\mathcal{H}}\mathbb{E}_p[r(x_n)^2] = M_{\mathcal{H}}\|r\|_{L^2(p)}^2.
$$

Hence the variance $\mathbb{E}\left[\|(A_n - \lambda I)V_\lambda^* + b_n\|_{\mathcal{H}}^2\right]$ is finally bounded by:

$$\sigma^2 := 10M_{\mathcal{H}}\|r\|_{L^2(p)}^2 + \left(\frac{8(1+\gamma^2)}{(1-\gamma)^2} + 16(1+\gamma^2)\right) M_{\mathcal{H}}\|V^*\|_{L^2(p)}^2.$$

Back to the main term, we get:

$$\mathbb{E}\left[\|(A_n - \lambda I)V_{n-1} + b_n\|_{\mathcal{H}}^2\right] \leq 2\lambda^2 \mathbb{E}W_{n-1}^0 + 8M_{\mathcal{H}}(1+\gamma^2)\mathbb{E}W_{n-1}^{-1} + 4\sigma^2.$$

Then, we get the result:

$$\begin{aligned}\mathbb{E}W_n^0 \leq \mathbb{E}W_{n-1}^0 &- 2\rho_n(1-\gamma)\mathbb{E}W_{n-1}^{-1} - 2\rho_n\lambda\mathbb{E}W_{n-1}^0 \\ &+ 2\rho_n^2\lambda^2\mathbb{E}W_{n-1}^0 + 8\rho_n^2 M_{\mathcal{H}}(1+\gamma^2)\mathbb{E}W_{n-1}^{-1} + 4\rho_{n-1}^2\sigma^2.\end{aligned}$$

□

Proof of Proposition 4. From Lemma 6 with $\lambda = 0$ and a constant step size $\rho \leq \bar{\rho}$, we obtain:

$$\mathbb{E}W_{k-1}^{-1} \leq \frac{\mathbb{E}W_{k-1}^0 - \mathbb{E}W_k^0}{\rho(1-\gamma)} + \frac{4\rho\sigma^2}{1-\gamma}.$$

Summing the latter inequality over k between 1 and n, and dividing by n, we get a telescoping sum:

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}W_{k-1}^{-1} \leq \frac{\mathbb{E}W_0^0 - \mathbb{E}W_n^0}{n\rho(1-\gamma)} + \frac{4\rho\sigma^2}{1-\gamma} \leq \frac{\mathbb{E}W_0^0}{n\rho(1-\gamma)} + \frac{4\rho\sigma^2}{1-\gamma}.$$

Then we use Jensen's inequality:

$$\mathbb{E}\|\overline{V}_n - V^*\|_{L^2(p)}^2 \leq \frac{\|V^*\|_{\mathcal{H}}^2}{(1-\gamma)\rho n} + \frac{4\rho\sigma^2}{1-\gamma}.$$

Finally, we choose a constant step size $\rho = 1/\sqrt{n}$ and $n \geq n_0 := 1/\bar{\rho}^2$. For $n \geq n_0$, this leads to the desired bound:

$$\mathbb{E}\|\overline{V}_n - V^*\|_{L^2(p)}^2 \leq O(1/\sqrt{n}).$$

□

Proof of Theorem 7. The three non-asymptotic upper-bounds will be proved one after another. In each case, we consider a couple (λ, ρ_n) that might be explicitly defined later, such that the assumptions of Lemma 6 hold. Then we pick particular choices of λ and ρ_n which balance the terms of the upper-bound, and check that the assumptions are indeed satisfied.

(a) Let $\lambda > 0$ and ρ a constant step size such that $\rho \leq \bar{\rho}$ and $\rho \leq 1/(2\lambda)$. In this case, Lemma 6 reads, for each $k \in \{1, ..., n\}$:

$$\mathbb{E}W_k^0 \leq (1-\rho\lambda)\mathbb{E}W_{k-1}^0 - \rho(1-\gamma)\mathbb{E}W_{k-1}^{-1} + 4\rho^2\sigma^2,$$

in particular:

$$\mathbb{E}W_k^0 \leq (1-\rho\lambda)\mathbb{E}W_{k-1}^0 + 4\rho^2\sigma^2. \tag{6.62}$$

Subtracting the quantity $\ell = \frac{4\rho\sigma^2}{\lambda}$, which is such that $\ell = (1-\rho\lambda)\ell + 4\rho^2\sigma^2$, from both sides of the inequality (6.62), we get:

$$\mathbb{E}W_k^0 - \frac{4\rho\sigma^2}{\lambda} \leq (1-\rho\lambda)\left(\mathbb{E}W_{k-1}^0 - \frac{4\rho\sigma^2}{\lambda}\right). \tag{6.63}$$

Since $\rho\lambda \leq 1/2$, the left-hand side is a geometrically contracting sequence and, applying (6.63) recursively, we get:

$$\begin{aligned}\mathbb{E}W_n^0 - \frac{4\rho\sigma^2}{\lambda} &\leq (1-\rho\lambda)^n\left(\mathbb{E}W_0^0 - \frac{4\rho\sigma^2}{\lambda}\right)\\ &\leq (1-\rho\lambda)^n\mathbb{E}W_0^0.\end{aligned}$$

Finally, the latter inequality and Proposition 1 imply:

$$\mathbb{E}W_n^0 \leq \frac{4\rho\sigma^2}{\lambda} + (1-\rho\lambda)^n\frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda^2}. \tag{6.64}$$

Let us take (λ,ρ) defined by $\lambda = \lambda_0 n^{-\frac{1}{3+\theta}}$ and $\rho = \frac{\log n}{\lambda n}$, for some λ_0. The conditions of Lemma 6 read:

- $\rho \leq 1/(2\lambda)$ if and only if $\frac{\log n}{n} \leq 1/2$, which is true for all $n \geq 1$.
- $\rho \leq \bar{\rho}$ if and only if $(\log n)n^{\frac{1}{3+\theta}-1}/\lambda_0 \leq \bar{\rho}$. Since $\theta > -1$, $\frac{1}{3+\theta} - 1 < -1/2$, hence

  $$(\log n)n^{\frac{1}{3+\theta}-1}/\bar{\rho} \to 0.$$

  In particular it is bounded for all $n \geq 1$. Hence if we define:

  $$\underline{\lambda}_\theta^{(0)} := \max\{(\log n)n^{\frac{1}{3+\theta}-1}/\bar{\rho} \mid n \geq 1\},$$

  then for $\lambda_0 \geq \underline{\lambda}_\theta^{(0)}$, $\rho \leq \bar{\rho}$ is satisfied. Note that $\underline{\lambda}_\theta^{(0)}$ is independent of n.

For this choice of λ and ρ, we get:

$$\mathbb{E}W_n^0 \leq \frac{4\sigma^2\log n}{\lambda_0^2 n^{1-\frac{2}{3+\theta}}} + \left(1-\frac{\log n}{n}\right)^n\frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda_0^2 n^{-\frac{2}{3+\theta}}}.$$

For $n \geq 1$, we recall that $\log\left(1-\frac{\log n}{n}\right) \leq -\frac{\log n}{n}$, hence $\left(1-\frac{\log n}{n}\right)^n \leq 1/n$ and:

$$\mathbb{E}W_n^0 \leq \frac{4\sigma^2(\log n)n^{-\frac{1+\theta}{3+\theta}}}{\lambda_0^2} + \frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda_0^2}n^{-\frac{1+\theta}{3+\theta}}.$$

From Proposition 2, we finally obtain the following inequalities:

$$\mathbb{E}\|V_n - V^*\|_{L^2(p)}^2 \leq 2M_{\mathcal{H}}\mathbb{E}\|V_n - V_\lambda^*\|_{\mathcal{H}}^2 + 2\|V_\lambda^* - V^*\|_{L^2(p)}^2$$

$$\leq \frac{8M_{\mathcal{H}}\sigma^2}{\lambda_0^2}(\log n)n^{-\frac{1+\theta}{3+\theta}} + \frac{2M_{\mathcal{H}}^2\|r\|_{L^2(p)}^2}{\lambda_0^2}n^{-\frac{1+\theta}{3+\theta}} + \frac{2\|\Sigma^{-\theta/2}V^*\|_{\mathcal{H}}^2\lambda_0^{1+\theta}}{(1-\gamma)^2}n^{-\frac{1+\theta}{3+\theta}}.$$

(b) Let $\lambda > 0$ and ρ a constant step size such that $\rho \leq \bar{\rho}$ and $\rho \leq 1/(2\lambda)$. In this case, Lemma 6 reads, for each $k \in \{1, ..., n\}$:

$$\mathbb{E}W_k^0 \leq (1-\rho\lambda)\mathbb{E}W_{k-1}^0 - \rho(1-\gamma)\mathbb{E}W_{k-1}^{-1} + 4\rho^2\sigma^2. \tag{6.65}$$

Using (6.65) recursively, we obtain:

$$\mathbb{E}W_n^0 \leq (1-\rho\lambda)^n\mathbb{E}W_0^0 - (1-\gamma)\rho\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}\mathbb{E}W_{k-1}^{-1} + 4\sigma^2\rho^2\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}.$$

Re-arranging the terms, we get:

$$\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}\mathbb{E}W_{k-1}^{-1} \leq \frac{(1-\rho\lambda)^n}{\rho(1-\gamma)}\mathbb{E}W_0^0 - \frac{1}{\rho(1-\gamma)}\mathbb{E}W_n^0 + \frac{4\sigma^2\rho}{1-\gamma}\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}$$
$$\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}\mathbb{E}W_{k-1}^{-1} \leq \frac{(1-\rho\lambda)^n}{\rho(1-\gamma)}\frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda^2} + \frac{4\sigma^2\rho}{1-\gamma}\sum_{k=1}^{n}(1-\rho\lambda)^{n-k},$$

using Proposition 1 on the last line.

Since $\sum_{k=1}^{n}(1-\rho\lambda)^{n-k} = \frac{1-(1-\rho\lambda)^n}{\rho\lambda}$, we get:

$$\frac{\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}\mathbb{E}W_{k-1}^{-1}}{\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}} \leq \frac{(1-\rho\lambda)^n}{1-(1-\rho\lambda)^n}\frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda(1-\gamma)} + \frac{4\sigma^2\rho}{1-\gamma}$$

Using Jensen's inequality, we get:

$$\mathbb{E}\|V_n^{(e)} - V_\lambda^*\|_{L^2(p)}^2 \leq \frac{(1-\rho\lambda)^n}{1-(1-\rho\lambda)^n}\frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda(1-\gamma)} + \frac{4\sigma^2\rho}{1-\gamma}, \tag{6.66}$$

with $V_n^{(e)} := \frac{\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}V_{k-1}}{\sum_{k=1}^{n}(1-\rho\lambda)^{n-k}}$ the exponentially weighted average iterate.

Let $\lambda = \lambda_0 n^{-\frac{1}{2+\theta}}$, for some $\lambda_0 > 0$, and $\rho = \frac{\log n}{\lambda n}$. The conditions of Lemma 6 are:

- $\rho \leq 1/(2\lambda)$ if and only if $\frac{\log n}{n} \leq 1/2$, which is true for all $n \geq 1$.
- $\rho \leq \bar{\rho}$ if and only if $(\log n)n^{\frac{1}{2+\theta}-1}/\lambda_0 \leq \bar{\rho}$. Since $\theta > -1$, $\frac{1}{2+\theta} - 1 < 0$, hence

  $$(\log n)n^{\frac{1}{2+\theta}-1}/\bar{\rho} \to 0.$$

  In particular it is bounded for all $n \geq 1$. Hence defining:

  $$\underline{\lambda}_\theta^{(e)} := \max\{(\log n)n^{\frac{1}{2+\theta}-1}/\bar{\rho} \mid n \geq 1\},$$

  then for $\lambda_0 \geq \underline{\lambda}_\theta^{(e)}$, $\rho \leq \bar{\rho}$ is satisfied. Again, $\underline{\lambda}_\theta^{(e)}$ is independent of n.

For this choice of parameters, for $n \geq 2$, we recall that:

$$(1-\rho\lambda)^n = \left(1-\frac{\log n}{n}\right)^n = \exp\left(n\log\left(1-\frac{\log n}{n}\right)\right) \leq \exp\left(n\left(-\frac{\log n}{n}\right)\right) \leq \frac{1}{n} \leq \frac{1}{2},$$

which implies:

$$\begin{aligned}
\mathbb{E}\|V_n^{(e)} - V_\lambda^*\|_{L^2(p)}^2 &\leq 2(1-\rho\lambda)^n \frac{n^{\frac{1}{2+\theta}} M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda_0(1-\gamma)} + \frac{4\sigma^2(\log n) n^{-\frac{1+\theta}{2+\theta}}}{\lambda_0(1-\gamma)} \\
&\leq \frac{2}{n} \cdot \frac{n^{\frac{1}{2+\theta}} M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda_0(1-\gamma)} + \frac{4\sigma^2(\log n) n^{-\frac{1+\theta}{2+\theta}}}{\lambda_0(1-\gamma)} \\
&\leq \frac{2n^{-\frac{1+\theta}{2+\theta}} M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda_0(1-\gamma)} + \frac{4\sigma^2(\log n) n^{-\frac{1+\theta}{2+\theta}}}{\lambda_0(1-\gamma)}.
\end{aligned}$$

From Proposition 2, we finally obtain the following inequalities:

$$\begin{aligned}
\mathbb{E}\|V_n^{(e)} - V^*\|_{L^2(p)}^2 &\leq 2\mathbb{E}\|V_n^{(e)} - V_\lambda^*\|_{L^2(p)}^2 + 2\|V_\lambda^* - V^*\|_{L^2(p)}^2 \\
\mathbb{E}\|V_n^{(e)} - V^*\|_{L^2(p)}^2 &\leq \frac{4M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda_0(1-\gamma)} n^{-\frac{1+\theta}{2+\theta}} + \frac{8\sigma^2}{\lambda_0(1-\gamma)}(\log n) n^{-\frac{1+\theta}{2+\theta}} \\
&\quad + \frac{2\|\Sigma^{-\theta/2} V^*\|_{\mathcal{H}}^2 \lambda_0^{1+\theta}}{(1-\gamma)^2} n^{-\frac{1+\theta}{2+\theta}}.
\end{aligned}$$

(c) Let $n \geq 1$ and $\lambda > 0$. We will consider a different step size schedule: first constant, then decreasing. For $k \in \{1, ..., \lfloor n/2 \rfloor - 1\}$, set $\rho_k = \frac{2\log n}{\lambda n} =: \rho$. Then for $k \in \{\lfloor n/2 \rfloor, ..., n\}$, set $\rho_k = \frac{1}{\lambda k}$.

- We first look at the first $\lfloor n/2 \rfloor - 1$ iterates.

Assume that λ is chosen such that $\rho \leq \min\{1/(2\lambda), \bar{\rho}\}$. Under this condition, from (6.64) which holds here, we obtain:

$$\mathbb{E}W_{\lfloor n/2 \rfloor - 1}^0 \leq \frac{4\rho\sigma^2}{\lambda} + (1-\rho\lambda)^{\lfloor n/2 \rfloor - 1} \frac{M_{\mathcal{H}}\|r\|_{L^2(p)}^2}{\lambda^2}. \tag{6.67}$$

- For the other iterates, we consider $\rho_k = \frac{1}{\lambda k}$. We also assume that λ is chosen such that

$$\rho_k \leq \min\{1/(2\lambda), \bar{\rho}\}, \quad \forall k \in \{\lfloor n/2 \rfloor, ..., n\}.$$

Under this condition, for $k \in \{\lfloor n/2 \rfloor, ..., n\}$, Lemma 6 reads:

$$\mathbb{E}W_k^0 \leq (1-\rho_k\lambda)\mathbb{E}W_{k-1}^0 - \rho_k(1-\gamma)\mathbb{E}W_{k-1}^{-1} + 4\rho_k^2\sigma^2.$$

Re-arranging the terms, we get:

$$\mathbb{E}W_{k-1}^{-1} \leq \frac{1}{1-\gamma}\left(\frac{1}{\rho_k} - \lambda\right)\mathbb{E}W_{k-1}^0 - \frac{1}{1-\gamma}\frac{1}{\rho_k}\mathbb{E}W_k^0 + \frac{4\sigma^2}{1-\gamma}\rho_k. \tag{6.68}$$

The step size is such that:

$$1/\rho_k - \lambda = \lambda k - \lambda = \lambda(k-1) = 1/\rho_{k-1},$$

where the very last equality only holds for $k \leq \lfloor n/2 \rfloor + 1$ (because of overlapping notations). Summing the above inequalities (6.68) for $k \in \{\lfloor n/2 \rfloor, ..., n\}$, we obtain a telescoping sum:

$$\begin{aligned}\sum_{k=\lfloor n/2 \rfloor}^{n} \mathbb{E}W_{k-1}^{-1} &\leq \frac{1}{1-\gamma} \sum_{k=\lfloor n/2 \rfloor}^{n} \left(\frac{\mathbb{E}W_{k-1}^{0}}{\rho_{k-1}} - \frac{\mathbb{E}W_{k}^{0}}{\rho_k} \right) + \frac{4\sigma^2}{1-\gamma} \sum_{k=\lfloor n/2 \rfloor}^{n} \rho_k \\ &\leq \frac{1}{1-\gamma} \lambda (\lfloor n/2 \rfloor - 1) \mathbb{E}W_{\lfloor n/2 \rfloor - 1}^{0} + \frac{4\sigma^2}{1-\gamma} \sum_{k=\lfloor n/2 \rfloor}^{n} \frac{1}{\lambda k} \\ &\leq \frac{\lambda n}{2(1-\gamma)} \mathbb{E}W_{\lfloor n/2 \rfloor - 1}^{0} + \frac{4\sigma^2}{1-\gamma} \frac{1+\log n}{\lambda}.\end{aligned}$$

For $n \geq 3$, so that $1 + \log(n) \leq 2 \log n$, from (6.67) and the latter inequality, we obtain:

$$\begin{aligned}\sum_{k=\lfloor n/2 \rfloor}^{n} \mathbb{E}W_{k-1}^{-1} &\leq \frac{\lambda n}{2(1-\gamma)} \left[\frac{4\rho\sigma^2}{\lambda} + (1-\rho\lambda)^{\lfloor n/2 \rfloor - 1} \frac{M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda^2} \right] + \frac{8\sigma^2}{1-\gamma} \frac{\log n}{\lambda} \\ &\leq \frac{\lambda n}{2(1-\gamma)} \left[\frac{8(\log n)\sigma^2}{\lambda^2 n} + \left(1 - \frac{2\log n}{n}\right)^{\lfloor n/2 \rfloor - 1} \frac{M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda^2} \right] \\ &\qquad + \frac{8\sigma^2}{1-\gamma} \frac{\log n}{\lambda}.\end{aligned}$$

Let us look at the central term:

$$\left(1 - \frac{2\log n}{n}\right)^{\lfloor n/2 \rfloor - 1} = \left(1 - \frac{2\log n}{n}\right)^{n/2} \left(1 - \frac{2\log n}{n}\right)^{\lfloor n/2 \rfloor - 1 - n/2}.$$

Since $2\log n/n \in [0,1]$ for any $n \geq 1$, and $\lfloor n/2 \rfloor - n/2 - 1 \geq -2$, we have:

$$\left(1 - \frac{2\log n}{n}\right)^{\lfloor n/2 \rfloor - 1 - n/2} \leq \left(1 - \frac{2\log n}{n}\right)^{-2} \leq \max_{u \geq 1} \left[\left(1 - \frac{2\log u}{u}\right)^{-2} \right] \leq 16.$$

This implies:

$$\begin{aligned}\left(1 - \frac{2\log n}{n}\right)^{\lfloor n/2 \rfloor - 1} &\leq 16 \left(1 - \frac{2\log n}{n}\right)^{n/2} \\ &\leq 16 \exp\left(n/2 \log\left(1 - \frac{2\log n}{n}\right)\right) \\ &\leq 16 \exp\left(-n/2 \times \frac{2\log n}{n}\right) \leq 16/n.\end{aligned}$$

Coming back to the telescoping sum, we get:

$$\sum_{k=\lfloor n/2 \rfloor}^{n} \mathbb{E}W_{k-1}^{-1} \leq \frac{\lambda n}{2(1-\gamma)} \left[\frac{8(\log n)\sigma^2}{\lambda^2 n} + \frac{16}{n} \frac{M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda^2} \right] + \frac{8\sigma^2}{1-\gamma} \frac{\log n}{\lambda}.$$

Then we divide by $n - \lfloor n/2 \rfloor + 1 \geq n/2$:

$$\frac{1}{n - \lfloor n/2 \rfloor + 1} \sum_{k=\lfloor n/2 \rfloor}^{n} \mathbb{E} W_{k-1}^{-1} \leq \frac{1}{(1-\gamma)} \left[\frac{8(\log n)\sigma^2}{\lambda n} + \frac{16}{n} \frac{M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda} \right] + \frac{16\sigma^2}{1-\gamma} \frac{\log n}{\lambda n},$$

where all the terms are of order $\tilde{O}(\frac{\log n}{\lambda n})$.

Let us consider the n-th tail averaged iterate defined by:

$$V_n^{(t)} := \frac{1}{n - \lfloor n/2 \rfloor + 1} \sum_{k=\lfloor n/2 \rfloor}^{n} V_{k-1}.$$

Using Jensen's inequality, we have a bound on its distance to V_λ^*:

$$\mathbb{E} \|V_n^{(t)} - V_\lambda^*\|_{L^2(p)}^2 \leq \frac{16}{n} \frac{M_{\mathcal{H}} \|r\|_{L^2(p)}^2}{\lambda(1-\gamma)} + \frac{24\sigma^2}{1-\gamma} \frac{\log n}{\lambda n}.$$

Now we need to choose λ such that $\rho_k \leq \min\{1/(2\lambda), \bar{\rho}\}$, for all k. Let $\lambda = \lambda_0 n^{-\frac{1}{2+\theta}}$.

- For the first half of the iterates, we take $\rho = \frac{2 \log n}{\lambda n}$, and $\rho \leq 1/(2\lambda)$ if and only if $\log n / n \leq 4$, which is true for $n \geq 9$.

Now $\rho \leq \bar{\rho}$ is equivalent to $\frac{2 \log n}{\lambda n} = (\log n) n^{\frac{1}{2+\theta} - 1} / \lambda_0 \leq \bar{\rho}$. Since $\theta > -1$, $\frac{1}{2+\theta} - 1 < 0$ and $(\log n) n^{\frac{1}{2+\theta} - 1} / \bar{\rho} \to 0$. In particular it is bounded for all $n \geq 1$. Hence using again:

$$\underline{\lambda}_\theta^{(e)} = \max\{(\log n) n^{\frac{1}{2+\theta} - 1} / \bar{\rho} \mid n \geq 1\},$$

then for $\lambda_0 \geq \underline{\lambda}_\theta^{(e)}$, $\rho \leq \bar{\rho}$ is satisfied.

- For the second half of the iterates, ρ_k is decreasing with k, hence a sufficient condition is that:

$$\frac{1}{\lambda \lfloor n/2 \rfloor} = \rho_{\lfloor n/2 \rfloor} \leq \min\{1/(2\lambda), \bar{\rho}\}.$$

For $n \geq 4$, $\lfloor n/2 \rfloor \geq 2$ and $\rho_{\lfloor n/2 \rfloor} \leq 1/(2\lambda)$ and this condition holds. On the other hand, the second condition reads:

$$\frac{1}{\lambda \lfloor n/2 \rfloor} = \frac{n^{\frac{1}{2+\theta}}}{\lambda_0 \lfloor n/2 \rfloor} \leq \frac{4 n^{\frac{1}{2+\theta} - 1}}{\lambda_0} \leq \bar{\rho},$$

for $n \geq 2$. Since $\theta > -1$, $\frac{1}{2+\theta} - 1 < 0$ and we get $4 n^{\frac{1}{2+\theta} - 1} / \bar{\rho} \to 0$. In particular, the latter term is bounded for all $n \geq 1$. Hence using:

$$\underline{\lambda}_\theta^{(t)} := \max\{\max\{4 n^{\frac{1}{2+\theta} - 1} / \bar{\rho} \mid n \geq 1\}, \underline{\lambda}_\theta^{(e)}\},$$

then for $\lambda_0 \geq \underline{\lambda}_\theta^{(t)}$, $\rho_k \leq \bar{\rho}$ is satisfied for all k.

For this specific choice of λ, we have the final bound:

$$\mathbb{E} \|V_n^{(t)} - V^*\|_{L^2(p)}^2 \leq 2 \mathbb{E} \|V_n^{(t)} - V_\lambda^*\|_{\mathcal{H}}^2 + 2 \|V_\lambda^* - V^*\|_{L^2(p)}^2$$

$$\leq \frac{32}{n}\frac{M_{\mathcal{H}}\|r\|^2_{L^2(p)}}{\lambda(1-\gamma)} + \frac{48\sigma^2}{1-\gamma}\frac{\log n}{\lambda n} + \frac{2\|\Sigma^{-\theta/2}V^*\|^2_{\mathcal{H}}\lambda_0^{1+\theta}}{(1-\gamma)^2}n^{-\frac{1+\theta}{2+\theta}}$$
$$\leq \frac{32M_{\mathcal{H}}\|r\|^2_{L^2(p)}}{\lambda_0(1-\gamma)}n^{-\frac{1+\theta}{2+\theta}} + \frac{48\sigma^2}{\lambda_0(1-\gamma)}(\log n)n^{-\frac{1+\theta}{2+\theta}}$$
$$+ \frac{2\|\Sigma^{-\theta/2}V^*\|^2_{\mathcal{H}}\lambda_0^{1+\theta}}{(1-\gamma)^2}n^{-\frac{1+\theta}{2+\theta}}.$$

Finally, we define $\underline{\lambda}_\theta := \max\{\underline{\lambda}_\theta^{(0)}, \underline{\lambda}_\theta^{(e)}, \underline{\lambda}_\theta^{(t)}\}$ which is used in the theorem as lower bound on λ_0. □

6.A.4 Stochastic TD with Markovian Sampling

We begin by reproducing Lemma 9 from Bhandari et al. (2018):

Lemma 7 (Control of couplings)**.** *Consider two random variables X and Y such that:*

$$X \to x_n \to x_{n+\tau} \to Y$$

forms a Markov chain, for some fixed $n \geq 1$ with $\tau > 0$. Assume the Markov chain mixes at uniform geometric rate, as defined in (6.36). Let X' and Y' denote independent copies drawn from the marginal distributions of X and Y, so that

$$\mathbb{P}(X' = \cdot, Y' = \cdot) = \mathbb{P}(X = \cdot) \otimes \mathbb{P}(Y = \cdot).$$

Then for any bounded function h:

$$|\mathbb{E}[h(X,Y)] - \mathbb{E}[h(X',Y')]| \leq 2\|h\|_\infty m\mu^\tau.$$

Note that, here, $\otimes$ does not refer to the outer product in the RKHS $\mathcal{H}$ but to the independent product of probability distributions.

Then we can state a descent lemma, similar to Lemma 6:

Lemma 8. *Assume that $\|V_\lambda^*\|_{\mathcal{H}} \leq B$ and that the Markov chain mixes geometrically (6.36). Let:*

$$\begin{cases} G^2 := 4M_{\mathcal{H}}^2B^2 + \lambda^2B^2 + M_{\mathcal{H}}R^2/2 \\ L := 12M_{\mathcal{H}}B + 2\sqrt{M_{\mathcal{H}}}R \\ C := 2M_{\mathcal{H}}B + \lambda B + \sqrt{M_{\mathcal{H}}}R \\ C' := 8M_{\mathcal{H}}B^2 + 4\sqrt{M_{\mathcal{H}}}BR. \end{cases}$$

Then for $n \geq 1$ and $\tau > 1$, the following inequality holds:

$$\mathbb{E}W_n^0 \leq (1-2\rho_n\lambda)\mathbb{E}W_{n-1}^0 - 2\rho_n(1-\gamma)\mathbb{E}W_{n-1}^{-1} + 2\rho_n\left(2C'm\mu^\tau + LC\sum_{k=n-\tau}^{n-1}\rho_k\right) + 4G^2\rho_n^2. \quad (6.69)$$

Proof of Lemma 8. Because of correlations between samples, the proof of Lemma 6 breaks here:

$$\mathbb{E}[\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}}] \neq \mathbb{E}[\langle V_{n-1} - V_\lambda^*, (A - \lambda I)V_{n-1} + b\rangle_{\mathcal{H}}].$$

A similar thing occurs in the variance term, where we cannot apply Lemma 5. An easy fix is to assume that what is inside the variance remains bounded a.s. This is allowed by our projection step. We can now assume that a.s., $\forall n, \|V_n\|_{\mathcal{H}} \leq B$, implying that a.s.:

$$\|A_n V_{n-1}\|_{\mathcal{H}} \leq \|A_n\|_{\mathrm{op}} B \leq 2M_{\mathcal{H}} B,$$

where $\|A_n\|_{\mathrm{op}} \leq 2M_{\mathcal{H}}$ is induced by the following computation, for $f \in \mathcal{H}$:

$$\begin{aligned}\|A_n f\|_{\mathcal{H}} &\leq \|\Phi(x_n) \otimes \Phi(x_n')f\|_{\mathcal{H}} + \|\Phi(x_n) \otimes \Phi(x_n)f\|_{\mathcal{H}} \\ &\leq \left(|\langle f, \Phi(x_n')\rangle_{\mathcal{H}}| + |\langle f, \Phi(x_n)\rangle_{\mathcal{H}}|\right)\|\Phi(x_n)\|_{\mathcal{H}} \\ &\leq 2\|f\|_{\mathcal{H}}\sqrt{M_{\mathcal{H}}}\sqrt{M_{\mathcal{H}}} = 2M_{\mathcal{H}}\|f\|_{\mathcal{H}}.\end{aligned}$$

Also, since the reward function is uniformly bounded by R, we get:

$$\|b_n\|_{\mathcal{H}}^2 = \|r(x_n)\Phi(x_n)\|_{\mathcal{H}}^2 \leq R^2 M_{\mathcal{H}}.$$

Finally, the projection step does not impact the proof since $\Pi_B : \mathcal{H} \to \mathcal{H}$ is 1-Lipschitz continuous (in $\mathcal{H}$-norm).

Decomposition of errors. Let us now reproduce the beginning of the proof of Lemma 6. We have this decomposition a.s.:

$$\begin{aligned}W_n^0 &= \|V_n - V_\lambda^*\|_{\mathcal{H}}^2 \\ &= \|\Pi_B[V_{n-1} + \rho_n((A_n - \lambda I)V_{n-1} + b_n)] - \Pi_B V_\lambda^*\|_{\mathcal{H}}^2 \\ &\leq \|V_{n-1} + \rho_n((A_n - \lambda I)V_{n-1} + b_n) - V_\lambda^*\|_{\mathcal{H}}^2 \\ &= \|V_{n-1} - V_\lambda^*\|_{\mathcal{H}}^2 + 2\rho_n\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}} \\ &\qquad + \rho_n^2\|(A_n - \lambda I)V_{n-1} + b_n\|_{\mathcal{H}}^2 \\ &\leq W_{n-1}^0 + 2\rho_n\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}} \\ &\qquad + 2\rho_n^2\|(A_n - \lambda I)V_{n-1}\|^2 + 2\rho_n^2\|b_n\|^2 \\ &\leq W_{n-1}^0 + 2\rho_n\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}} \\ &\qquad + 4\rho_n^2(4M_{\mathcal{H}}^2 B^2 + \lambda^2 B^2) + 2\rho_n^2 R^2 M_{\mathcal{H}}.\end{aligned}$$

Taking the expectation with respect to $\mathcal{F}_n = \sigma(z_1, ..., z_n)$ (where $z_i = (x_i, x_i')$), we get three terms:

$$\begin{aligned}\mathbb{E}W_n^0 \leq \mathbb{E}W_{n-1}^0 &+ 2\rho_n\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}}\right] \\ &+ \rho_n^2 \underbrace{\left(16M_{\mathcal{H}}^2 B^2 + 4\lambda^2 B^2 + 2M_{\mathcal{H}} R^2\right)}_{:=4G^2}.\end{aligned}$$

We then deal with the central expectation.

$$\begin{aligned}\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_n - \lambda I)V_{n-1} + b_n\rangle_{\mathcal{H}}\right] &= \mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A - \lambda I)V_{n-1} + b\rangle_{\mathcal{H}}\right] \\ &\quad + \mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_n - A)V_{n-1} + (b_n - b)\rangle_{\mathcal{H}}\right].\end{aligned}$$

The first term has already been treated in Lemma 3:

$$\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A - \lambda I)V_{n-1} + b\rangle_{\mathcal{H}}\right] \leq -(1-\gamma)\mathbb{E}W_{n-1}^{-1} - \lambda\mathbb{E}W_{n-1}^{0}.$$

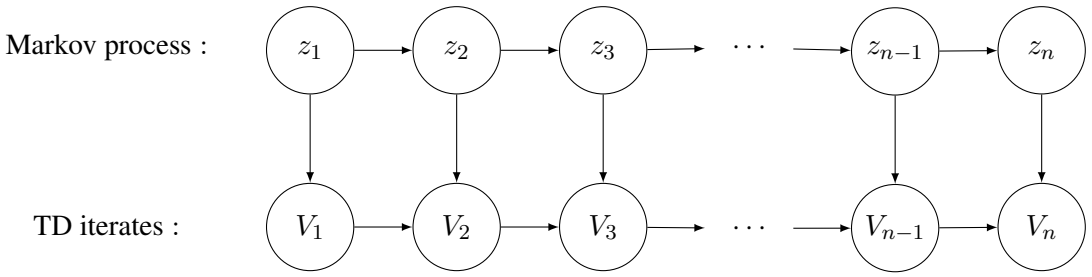
To control the remaining expectation (the bias), we must use a coupling argument. We use the notation:

$$\zeta(V_{n-1}, z_n) := \langle V_{n-1} - V_\lambda^*, (A_n - A)V_{n-1} + (b_n - b)\rangle_{\mathcal{H}}.$$

Note that in general:

$$\mathbb{E}\zeta(V_{n-1}, z_n) = \mathbb{E}[\mathbb{E}[\zeta(V_{n-1}, z_n)|\mathcal{F}_{n-1}]] \neq 0,$$

where $\mathcal{F}_k = \sigma(z_1, ..., z_k) = \sigma(z_1, V_1, ..., z_k, V_k)$. The dependence between the random variables is summarized in the following diagram.



Using the mixing assumption, we can control the deviation between the expectations of a bounded function of two iterates separated by τ steps, in the coupled *v.s.* the decoupled case. In other words, if τ is large, we can almost consider the iterates as being independent. This is achieved using Lemma 7.

Bounding the bias. Our goal here is to find an upper-bound of $\mathbb{E}[\zeta(V_{n-1}, z_n)]$. Let $\tau \in \mathbb{N}$, $\tau > 1$. This can be done in two steps:

(1) Relate $\mathbb{E}[\zeta(V_{n-1}, z_n)]$ to $\mathbb{E}[\zeta(V_{n-1-\tau}, z_n)]$, because ζ is Lipschitz in the first variable, as a quadratic function over a bounded domain. This is true almost surely, hence in expectation.

(2) Relate $\mathbb{E}[\zeta(V_{n-1-\tau}, z_n)]$ to $\mathbb{E}[\zeta(V'_{n-1-\tau}, z'_n)] = 0$, where $V'_{n-1-\tau}$ and z'_n are independent copies of $V_{n-1-\tau}$ and z_n that are decoupled.

(1) First we prove that ζ is L-Lipschitz in the first variable on the $\mathcal{H}$ ball of radius B: for fixed $V, V' \in \mathcal{H}$ with norm bounded by B, and z_n:

$$\begin{aligned}
|\zeta(V, z_n) - \zeta(V', z_n)| &= \Big|\langle (A_n - A)V + b_n - b, V - V_\lambda^*\rangle_{\mathcal{H}} \\
&\qquad - \langle (A_n - A)V' + b_n - b, V' - V_\lambda^*\rangle_{\mathcal{H}}\Big| \\
&= \Big|\langle (A_n - A)V + b_n - b, V - V'\rangle_{\mathcal{H}} \\
&\qquad + \langle (A_n - A)(V - V'), V' - V_\lambda^*\rangle_{\mathcal{H}}\Big|,
\end{aligned}$$

where we have used the equality:

$$\langle a, b\rangle - \langle c, d\rangle = \langle a, b - d\rangle + \langle a - c, d\rangle.$$

This implies that

$$\begin{aligned}|\zeta(V, z_n) - \zeta(V', z_n)| &\leq \|(A_n - A)V + b_n - b\|_{\mathcal{H}} \cdot \|V - V'\|_{\mathcal{H}} \\ &\quad + \|(A_n - A)(V - V')\|_{\mathcal{H}} \cdot \|V' - V_\lambda^*\|_{\mathcal{H}} \\ &\leq (4M_{\mathcal{H}}B + 2\sqrt{M_{\mathcal{H}}}R)\|V - V'\|_{\mathcal{H}} + 8M_{\mathcal{H}}B\|V - V'\|_{\mathcal{H}} \\ &= L\|V - V'\|_{\mathcal{H}},\end{aligned}$$

for $L := 4M_{\mathcal{H}}B + 2\sqrt{M_{\mathcal{H}}}R + 8M_{\mathcal{H}}B$.

Then almost surely, since all the V_k are such that $\|V_k\|_{\mathcal{H}} \leq B$:

$$\begin{aligned}\zeta(V_{n-1}, z_n) &\leq \zeta(V_{n-1-\tau}, z_n) + |\zeta(V_{n-1}, z_n) - \zeta(V_{n-1-\tau}, z_n)| \\ &\leq \zeta(V_{n-1-\tau}, z_n) + L\|V_{n-1} - V_{n-1-\tau}\|_{\mathcal{H}} \\ &\leq \zeta(V_{n-1-\tau}, z_n) + L\sum_{k=n-\tau}^{n-1} \|V_k - V_{k-1}\|_{\mathcal{H}} \\ &= \zeta(V_{n-1-\tau}, z_n) + L\sum_{k=n-\tau}^{n-1} \rho_k\|A_kV_{k-1} - \lambda V_{k-1} + b_k\|_{\mathcal{H}} \\ &\leq \zeta(V_{n-1-\tau}, z_n) + L\sum_{k=n-\tau}^{n-1} \rho_k \underbrace{(2M_{\mathcal{H}}B + \lambda B + \sqrt{M_{\mathcal{H}}}R)}_{=:C}.\end{aligned}$$

Taking the expectation w.r.t. $\mathbb{P}(z_1, ..., z_n)$, we get:

$$\mathbb{E}\zeta(V_{n-1}, z_n) \leq \mathbb{E}\zeta(V_{n-1-\tau}, z_n) + LC\sum_{k=n-\tau}^{n-1} \rho_k.$$

(2) Then we use a coupling argument with Lemma 7. First, we need to bound $\|\zeta\|_\infty$.

For fixed V, z_n, with $\|V\|_{\mathcal{H}} \leq B$, almost surely, one may notice that:

$$\begin{aligned}|\zeta(V, z_n)| &= |\langle (A_n - A)V + b_n - b, V - V_\lambda^*\rangle_{\mathcal{H}}| \\ &\leq \|V - V_\lambda^*\|_{\mathcal{H}}\Big(\|(A_n - A)V\|_{\mathcal{H}} + \|b_n - b\|_{\mathcal{H}}\Big) \\ &\leq 2B(4M_{\mathcal{H}}B + 2\sqrt{M_{\mathcal{H}}}R) =: C'.\end{aligned}$$

In Lemma 7, set $X = (z_1, ..., z_{n-1-\tau})$ and $Y = z_n$. Since:

$$X \to x_{n-\tau} \to x_n \to Y$$

forms a Markov chain, then let X' and Y' denote independent copies drawn from the marginal distributions of X and Y, so that $\mathbb{P}(X' = \cdot, Y' = \cdot) = \mathbb{P}(X = \cdot) \otimes \mathbb{P}(Y = \cdot)$. Then applying Lemma 7 to the function $h : (X, Y) \to \zeta(V_{n-1-\tau}, z_n)$ (recalling that $V_{n-1-\tau}$ is fully determined by the values of X):

$$|\mathbb{E}[h(X, Y)] - \mathbb{E}[h(X', Y')]| \leq 2\|h\|_\infty m\mu^\tau,$$

which, here, reads:

$$|\mathbb{E}\zeta(V_{n-1-\tau}, z_n) - \mathbb{E}\zeta(V'_{n-1-\tau}, z'_n)| \leq 2C'm\mu^\tau.$$

By definition of the random variables X', Y':

$$\mathbb{E}\zeta(V'_{n-1-\tau}, z'_n) = \mathbb{E}[\mathbb{E}[\zeta(V'_{n-1-\tau}, z'_n)|V'_{n-1-\tau}]] = 0.$$

Putting everything together, we get:

$$\begin{aligned}\mathbb{E}\zeta(V_{n-1}, z_n) &\leq \mathbb{E}\zeta(V_{n-1-\tau}, z_n) + LC\sum_{k=n-\tau}^{n-1}\rho_k \\ &\leq 2C'm\mu^\tau + LC\sum_{k=n-\tau}^{n-1}\rho_k.\end{aligned}$$

Using this upper-bound is interesting if $m\mu^\tau$ is of the order of $\sum_{k=n-\tau}^{n-1}\rho_k$. Else (for small n), one can always choose $\tau = n-1$, so that, because V_0 is deterministic,

$$\mathbb{E}\zeta(V_{n-1}, z_n) \leq \underbrace{\mathbb{E}\zeta(V_0, z_n)}_{=0} + LC\sum_{k=1}^{n-1}\rho_k.$$

□

Proof of Theorem 8. We use a constant step size ρ. From Lemma 8:

$$\mathbb{E}W_n^0 \leq (1-2\rho\lambda)\mathbb{E}W_{n-1}^0 - 2\rho(1-\gamma)\mathbb{E}W_{n-1}^{-1} + 2\rho\left(2C'm\mu^\tau + LC\tau\rho\right) + 4G^2\rho^2.$$

In particular, we choose τ such that $\mu^\tau = \rho$, that is $\tau = \frac{\log\rho}{\log\mu} = \frac{\log(1/\rho)}{\log(1/\mu)}$, and get:

$$\begin{aligned}\mathbb{E}W_n^0 &\leq (1-2\rho\lambda)\mathbb{E}W_{n-1}^0 - 2\rho(1-\gamma)\mathbb{E}W_{n-1}^{-1} + 2\rho\left(2C'm\rho + LC\rho\frac{\log(1/\rho)}{\log(1/\mu)}\right) + 4G^2\rho^2 \\ &\leq (1-2\rho\lambda)\mathbb{E}W_{n-1}^0 - 2\rho(1-\gamma)\mathbb{E}W_{n-1}^{-1} + \rho^2\left(\underbrace{4C'm + 2LC\frac{\log(1/\rho)}{\log(1/\mu)} + 4G^2}_{=:4\tilde{\sigma}^2_{\lambda,\rho}}\right).\end{aligned}$$

This expression is similar to (6.65). Adapting the proof of Theorem 7 (b), we obtain:

$$\mathbb{E}\|V_n^{(e)} - V_\lambda^*\|^2_{L^2(p)} \leq \frac{(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{M_{\mathcal{H}}\|r\|^2_{L^2(p)}}{\lambda(1-\gamma)} + \frac{2\rho\sigma^2_{\lambda,\rho}}{1-\gamma},$$

with $V_n^{(e)} = \frac{\sum_{k=1}^n (1-2\rho\lambda)^{n-k}V_{k-1}}{\sum_{k=1}^n (1-2\rho\lambda)^{n-k}}$ the exponentially weighted average iterate.

Finally, we get:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq \frac{2(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{M_{\mathcal{H}}\|r\|^2_{L^2(p)}}{\lambda(1-\gamma)} + \frac{4\rho\sigma^2_{\lambda,\rho}}{1-\gamma} + \frac{2\|\Sigma^{-\theta/2}V^*\|^2_{\mathcal{H}}}{(1-\gamma)^2}\lambda^{1+\theta}.$$

Note that $\sigma^2_{\lambda,\rho}$ depends on λ, ρ, and B. We look at two cases:

(i) we are given an oracle on B that does not depend on λ.

(ii) we use the bound of order $O(1/\lambda)$ given by Proposition 1:

$$B = \frac{\sqrt{M_{\mathcal{H}}}\|r\|_{L^2(p)}}{\lambda}.$$

Case *(i)*: with oracle. For a fixed λ (later chosen to be the optimal one), assume that we know a bound B on $\|V_\lambda^*\|_{\mathcal{H}}$. Then $B = O(1)$, and assuming $\lambda = O(1)$, we only keep track of the dependence in μ and put all the other constants in $O(1)$:

$$\sigma_{\lambda,\rho}^2 = O\left(\frac{\log(1/\rho)}{\log(1/\mu)}\right) + O(1).$$

Let us look for λ of the form $\lambda = n^{-\alpha}$ with $\alpha \in (0,1)$, then:

$$\mathbb{E}\|V_n^{(e)} - V^*\|_{L^2(p)}^2 \leq O\left(\frac{(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{1}{\lambda}\right) + O\left(\rho\frac{\log(1/\rho)}{\log(1/\mu)}\right) + O(\rho) + O\left(\lambda^{1+\theta}\right).$$

Taking $\rho = \frac{\log n}{2\lambda n}$, the latter inequality leads to:

$$\mathbb{E}\|V_n^{(e)} - V^*\|_{L^2(p)}^2 \leq O\left(\frac{1}{n\lambda}\right) + O\left(\frac{\log n}{\lambda n}\frac{\log(1/\rho)}{\log(1/\mu)}\right) + O(\rho) + O\left(\lambda^{1+\theta}\right).$$

Recalling that $\lambda = n^{-\alpha}$, we get:

$$\mathbb{E}\|V_n^{(e)} - V^*\|_{L^2(p)}^2 \leq O\left(n^{\alpha-1}\right) + O\left(\frac{(\log n)^2 n^{\alpha-1}}{\log(1/\mu)}\right) + O\left((\log n)n^{\alpha-1}\right) + O\left(n^{-\alpha(1+\theta)}\right).$$

The first and third terms are smaller than the second one. We can choose α such that: $\alpha - 1 = -\alpha(1+\theta)$, *i.e.*, $\alpha = \frac{1}{2+\theta}$, then we get the following inequality:

$$\mathbb{E}\left[\|V_n^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left(\frac{(\log n)^2 n^{-\frac{1+\theta}{2+\theta}}}{\log(1/\mu)}\right).$$

Case *(ii)*: without oracle. Now $B = O(1/\lambda)$. Let us unroll all the constants to see the full dependencies:

$$\begin{aligned}
\sigma_{\lambda,\rho}^2 &= C'm + \frac{1}{2}LC\frac{\log(1/\rho)}{\log(1/\mu)} + G^2 \\
&= 8mM_{\mathcal{H}}B^2 + 4m\sqrt{M_{\mathcal{H}}}RB \\
&\qquad + \left(12M_{\mathcal{H}}B + 2\sqrt{M_{\mathcal{H}}}R\right)\left(2M_{\mathcal{H}}B + \lambda B + \sqrt{M_{\mathcal{H}}}R\right)\frac{\log(1/\rho)}{2\log(1/\mu)} \\
&\qquad + 4M_{\mathcal{H}}^2B^2 + \lambda^2B^2 + M_{\mathcal{H}}R^2/2 \\
&= B^2\left(8mM_{\mathcal{H}} + 4M_{\mathcal{H}}^2 + \lambda^2 + 12M_{\mathcal{H}}^2\frac{\log(1/\rho)}{\log(1/\mu)} + 6\lambda M_{\mathcal{H}}\frac{\log(1/\rho)}{\log(1/\mu)}\right) \\
&\qquad + B\left(4m\sqrt{M_{\mathcal{H}}}R + 8M_{\mathcal{H}}^{3/2}R\frac{\log(1/\rho)}{\log(1/\mu)} + \lambda\sqrt{M_{\mathcal{H}}}R\frac{\log(1/\rho)}{\log(1/\mu)}\right)
\end{aligned}$$

$$+\left(M_{\mathcal{H}}R^2/2 + M_{\mathcal{H}}R^2\frac{\log(1/\rho)}{\log(1/\mu)}\right).$$

We focus on the case $\lambda = O(1)$, so this simplifies a bit to:

$$\sigma^2_{\lambda,\rho} = O(B^2) + O\left(\frac{\log(1/\rho)}{\log(1/\mu)}B^2\right) + O(B) + O\left(\frac{\log(1/\rho)}{\log(1/\mu)}B\right) + O(1) + O\left(\frac{\log(1/\rho)}{\log(1/\mu)}\right).$$

On the other hand, we recall that $B = O(1/\lambda)$, hence:

$$\sigma^2_{\lambda,\rho} = O(1/\lambda^2) + O\left(\frac{\log(1/\rho)}{\lambda^2\log(1/\mu)}\right) + O\left(\frac{1}{\lambda}\right) + O\left(\frac{\log(1/\rho)}{\lambda\log(1/\mu)}\right) + O(1) + O\left(\frac{\log(1/\rho)}{\log(1/\mu)}\right).$$

Let us look for λ of the form $\lambda = n^{-\alpha}$ with $\alpha \in (0,1)$.

In this case, we get $\sigma^2_{\lambda,\rho} = O(1/\lambda^2) + O\left(\frac{\log(1/\rho)}{\log(1/\mu)}1/\lambda^2\right)$ and:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(\frac{(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{1}{\lambda}\right) + O\left(\frac{\rho}{\lambda^2}\right) + O\left(\frac{\rho}{\lambda^2}\frac{\log(1/\rho)}{\log(1/\mu)}\right) + O\left(\lambda^{1+\theta}\right).$$

Let us now set $\rho = \frac{\log n}{2\lambda n}$, we obtain:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(\frac{1}{n\lambda}\right) + O\left(\frac{\log n}{\lambda^3 n}\right) + O\left(\frac{\log n}{\lambda^3 n}\frac{\log(1/\rho)}{\log(1/\mu)}\right) + O\left(\lambda^{1+\theta}\right).$$

Recalling that $\lambda = n^{-\alpha}$, we obtain:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(n^{\alpha-1}\right) + O\left((\log n)n^{3\alpha-1}\right) + O\left(\frac{(\log n)^2 n^{3\alpha-1}}{\log(1/\mu)}\right) + O\left(n^{-\alpha(1+\theta)}\right).$$

The first and second term are smaller than the third one. We can choose α such that: $3\alpha - 1 = -\alpha(1+\theta)$, *i.e.*, $\alpha = \frac{1}{4+\theta}$, hence we get the convergence rate:

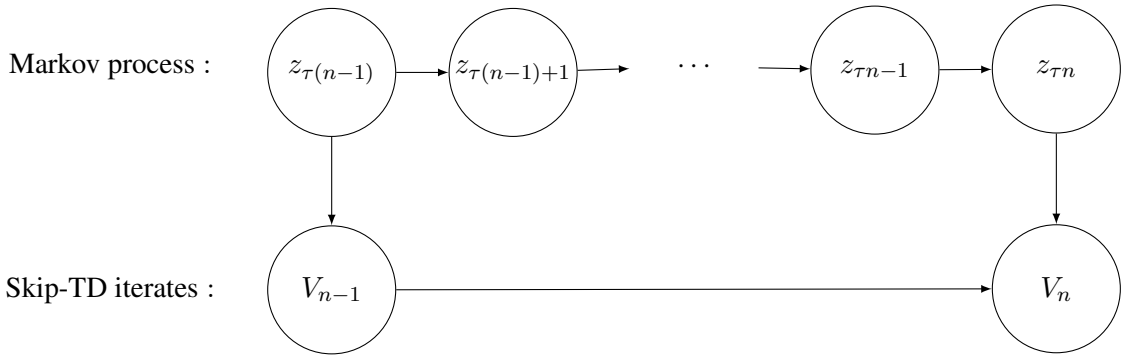
$$\mathbb{E}\left[\|V_n^{(e)} - V^*\|^2_{L^2(p)}\right] \leq O\left(\frac{(\log n)^2 n^{-\frac{1+\theta}{4+\theta}}}{\log(1/\mu)}\right).$$

□

Proof of Corollary 2. We consider the iterates (6.40), for some positive integer τ to be chosen later. The beginning of the proof of Lemma 8 can be reproduced:

$$\begin{aligned}\mathbb{E}W_n^0 &\leq \mathbb{E}W_{n-1}^0 + 2\rho_n\mathbb{E}\left[\langle V_{n-1} - V_\lambda^*, (A_{n\tau} - \lambda I)V_{n-1} + b_{n\tau}\rangle_{\mathcal{H}}\right] + 4\rho_n^2 G^2 \\ &\leq (1-2\rho_n\lambda)\mathbb{E}W_{n-1}^0 - 2\rho_n(1-\gamma)\mathbb{E}W_{n-1}^{-1} + 4G^2\rho_n^2 + 2\rho_n\mathbb{E}\zeta(V_{n-1}, z_{n\tau}).\end{aligned}$$

The only difference is that we now consider $\mathbb{E}\zeta(V_{n-1}, z_{n\tau})$ instead of $\mathbb{E}\zeta(V_{n-1}, z_n)$. To bound it, we do not need the step (1) (which exploits the fact that ζ is Lipschitz), and directly go to step (2). The dependencies between the random variables are now:



Applying again Lemma 7, we get the upper-bound:

$$|\mathbb{E}\zeta(V_{n-1}, z_{n\tau}) - \mathbb{E}\zeta(V'_{n-1}, z'_{n\tau})| \leq 2C'm\mu^{\tau-1},$$

where V'_{n-1}, and $z'_{n\tau}$ are independent copies such that $\mathbb{E}\zeta(V'_{n-1}, z'_{n\tau}) = 0$.

Now, using a constant step size ρ, we set $\tau := \lceil \frac{\log(1/\rho)}{\log(1/\mu)} + 1 \rceil$, such that $\mu^{\tau-1} \leq \rho$, which implies:

$$\mathbb{E}W_n^0 \leq (1 - 2\rho\lambda)\mathbb{E}W_{n-1}^0 - 2\rho(1-\gamma)\mathbb{E}W_{n-1}^{-1} + 4G^2\rho^2 + 4\rho^2 C'm.$$

Now we can do the same proof as for Theorem 8 with $\sigma^2_{\lambda,\rho} = C'm + G^2$, now independent of ρ:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq \frac{2(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{M_{\mathcal{H}}\|r\|^2_{L^2(p)}}{\lambda(1-\gamma)} + \frac{4\rho\sigma^2_{\lambda,\rho}}{1-\gamma} + \frac{2\|\Sigma^{-\theta/2}V^*\|^2_{\mathcal{H}}}{(1-\gamma)^2}\lambda^{1+\theta}.$$

Case *(i)*: with oracle. Now $\sigma^2_{\lambda,\rho} = O(1)$. We look for λ of the form $\lambda = n^{-\alpha}$, $\alpha \in (0,1)$:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(\frac{(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{1}{\lambda}\right) + O(\rho) + O\left(\lambda^{1+\theta}\right).$$

Let us now set $\rho = \frac{\log n}{2\lambda n}$:

$$\mathbb{E}\|V_n^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(\frac{1}{n\lambda}\right) + O(\rho) + O\left(\lambda^{1+\theta}\right).$$

Of course, to compute the n-th iteration, one needs to generate τn samples from the Markov chain. So for a fair comparison, we must look at the convergence of $V_{n/\tau}$ (assuming n is a multiple of τ for simplicity):

$$\mathbb{E}\|V_{n/\tau}^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(\frac{\tau}{n\lambda}\right) + O(\rho) + O\left(\lambda^{1+\theta}\right).$$

τ is such that:

$$\tau = O\left(\frac{\log(1/\rho)}{\log(1/\mu)}\right) = O\left(\frac{\log n}{\log(1/\mu)}\right).$$

Expressing everything with n only, we obtain:

$$\mathbb{E}\|V_{n/\tau}^{(e)} - V^*\|^2_{L^2(p)} \leq O\left(\frac{\log n}{\log(1/\mu)}n^{\alpha-1}\right) + O\left((\log n)n^{\alpha-1}\right) + O\left(n^{-\alpha(1+\theta)}\right).$$

Choosing α such that: $\alpha - 1 = -\alpha(1+\theta)$, *i.e.*, $\alpha = \frac{1}{2+\theta}$, we get:

$$\mathbb{E}\left[\|V_{n/\tau}^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left(\frac{(\log n) n^{-\frac{1+\theta}{2+\theta}}}{\log(1/\mu)}\right).$$

Case *(ii)*: without oracle. Using $B = O(1/\lambda)$, one may notice that:

$$\sigma_{\lambda,\rho}^2 = O(1/\lambda^2) + O\left(\frac{1}{\lambda}\right) + O(1).$$

Let us look for λ of the form $\lambda = n^{-\alpha}$ with $\alpha \in (0,1)$. We also set $\rho = \frac{\log n}{2\lambda n}$. In this case $\sigma_{\lambda,\rho}^2 = O(1/\lambda^2)$ and:

$$\begin{aligned}\mathbb{E}\|V_n^{(e)} - V^*\|_{L^2(p)}^2 &\leq O\left(\frac{(1-2\rho\lambda)^n}{1-(1-2\rho\lambda)^n}\frac{1}{\lambda}\right) + O\left(\frac{\rho}{\lambda^2}\right) + O\left(\lambda^{1+\theta}\right) \\ &\leq O\left(\frac{1}{n\lambda}\right) + O\left(\frac{\rho}{\lambda^2}\right) + O\left(\lambda^{1+\theta}\right).\end{aligned}$$

If n is a multiple of τ:

$$\mathbb{E}\|V_{n/\tau}^{(e)} - V^*\|_{L^2(p)}^2 \leq O\left(\frac{\tau}{n\lambda}\right) + O\left(\frac{\rho}{\lambda^2}\right) + O\left(\lambda^{1+\theta}\right).$$

τ is such that:

$$\tau = O\left(\frac{\log(1/\rho)}{\log(1/\mu)}\right) = O\left(\frac{\log n}{\log(1/\mu)}\right).$$

Expressing everything with n only, we get:

$$\mathbb{E}\|V_{n/\tau}^{(e)} - V^*\|_{L^2(p)}^2 \leq O\left(\frac{\log n}{\log(1/\mu)} n^{\alpha-1}\right) + O\left((\log n) n^{3\alpha-1}\right) + O\left(n^{-\alpha(1+\theta)}\right).$$

Choosing α such that: $3\alpha - 1 = -\alpha(1+\theta)$ *i.e.*, $\alpha = \frac{1}{4+\theta}$, we get:

$$\mathbb{E}\left[\|V_{n/\tau}^{(e)} - V^*\|_{L^2(p)}^2\right] \leq O\left((\log n) n^{-\frac{1+\theta}{4+\theta}}\right).$$

□

6.B Experimental Design

6.B.1 Geometric Mixing of the Markov Chain

Lemma 9. *Consider the Markov chain defined on the torus* $[0,1]$ *by:*

- *with probability* ε, $x_{n+1} \sim \mathcal{U}([0,1])$*;*

- *with probability* $1-\varepsilon$, $x_{n+1} = x_n$.

This Markov chain mixes to the uniform distribution at uniform geometric rate $(1-\varepsilon)$*:*

$$\sup_{x\in[0,1]} d_{TV}\big(\mathbb{P}(x_n \in \cdot | x_0 = x), \mathcal{U}([0,1])\big) \leq (1-\varepsilon)^n.$$

Proof. Let $x \in [0,1]$, $p = \mathcal{U}([0,1])$ the uniform distribution, and $p_n := \mathbb{P}(x_n \in \cdot | x_0 = x)$.

We will show that:

$$d_{TV}\left(p_n, p\right) \leq (1-\varepsilon)^n.$$

For $n = 1$, we have:

$$p_1 = \mathbb{P}(x_1 \in \cdot | x_0 = x) = \varepsilon p + (1-\varepsilon)\delta_x.$$

Then, for $n = 2$, we obtain:

$$\mathbb{P}(x_2 \in \cdot | x_0 = x, x_1) = \varepsilon p + (1-\varepsilon)\delta_{x_1}.$$

Taking the marginal with respect to $x_1|x_0$, we get:

$$\begin{aligned}
p_2 &= \mathbb{P}(x_2 \in \cdot | x_0 = x) \\
p_2 &= \int (\varepsilon p + (1-\varepsilon)\delta_{x_1}) \mathrm{d}p_1(x_1) \\
&= \varepsilon p + (1-\varepsilon) \int \delta_{x_1}(\varepsilon p(x_1) + (1-\varepsilon)\delta_x(x_1)) \mathrm{d}x_1 \\
&= \varepsilon p + \varepsilon(1-\varepsilon)p + (1-\varepsilon)^2 \delta_x.
\end{aligned}$$

A simple recursion on n shows that, for $n \geq 1$:

$$\begin{aligned}
p_n &= (\varepsilon + (1-\varepsilon)\varepsilon + ... + (1-\varepsilon)^{n-1}\varepsilon)p + (1-\varepsilon)^n \delta_x \\
&= (1 - (1-\varepsilon)^n)p + (1-\varepsilon)^n \delta_x,
\end{aligned}$$

which implies:

$$\begin{aligned}
d_{TV}\left(p_n, p\right) &= \sup_{A\in\mathcal{A}} \Big| p_n(A) - p(A) \Big| \\
&= (1-\varepsilon)^n \sup_{A\in\mathcal{A}} \Big| \delta_x(A) - p(A) \Big| \\
&\leq (1-\varepsilon)^n.
\end{aligned}$$

□

6.B.2 Implementation Details

The "kernel trick" enables an implementation of the non-parametric TD algorithm up to iteration n, which only uses the kernel matrix with entries $K_{i,j} := K(x_i, x_j)$, for $1 \leq i, j \leq n+1$.

Each value function V_k, for $1 \leq k \leq n$ belongs to the span of the basis of functions $(\Phi(x_j))_{1 \leq j \leq k}$:

$$V_k = \sum_{j=1}^{k} \alpha_{k,j} \Phi(x_j).$$

Hence V_k is represented in memory by the vector $(\alpha_{k,j})_{1 \leq j \leq k}$.

The TD iterations are equivalent to filling the lower-triangular matrix α:

$$\begin{cases} \alpha_{1,1} &= \rho_1 r(x_1) & \\ \alpha_{k,j} &= (1 - \rho_k \lambda) \alpha_{k-1,j} & \text{for } 1 \leq j < k \leq n \\ \alpha_{k,k} &= \rho_k r(x_k) + \rho_k \sum_{j=1}^{k-1} \alpha_{k-1,j} \left(\gamma K_{j,k+1} - K_{j,k} \right) & \text{for } 1 \leq k \leq n. \end{cases}$$

At inference time, for $x \in \mathcal{X}$, $V_k(x)$ can be computed from α and the vector $(K(x_j, x))_{1 \leq j \leq k}$:

$$V_k(x) = \sum_{j=1}^{k} \alpha_{k,j} K(x_j, x).$$

Finally, averaging can be performed by simple operations on α, which correspond to exchanging the indices of a triangular sum. Indeed, if:

$$V_n^{(e)} = \sum_{k=1}^{n} w_{k,n} V_{k-1},$$

for instance with $w_{k,n} := (1 - \rho\lambda)^{n-k} / \sum_{k=1}^{n} (1 - \rho\lambda)^{n-k}$, then, using that $V_0 = 0$:

$$\begin{aligned} V_n^{(e)} &= \sum_{k=2}^{n} w_{k,n} \sum_{j=1}^{k-1} \alpha_{k-1,j} \Phi(x_j) \\ &= \sum_{1 \leq j < k \leq n} w_{k,n} \alpha_{k-1,j} \Phi(x_j) \\ &= \sum_{j=1}^{n-1} \Phi(x_j) \sum_{k=j+1}^{n} w_{k,n} \alpha_{k-1,j} \\ &= \sum_{j=1}^{n-1} \alpha_{n,j}^{(e)} \Phi(x_j), \end{aligned}$$

with $\alpha_{n,j}^{(e)} := \sum_{k=j+1}^{n} w_{k,n} \alpha_{k-1,j}$.

This implementation requires the storage of $O(n^2)$ values and $O(n^2)$ computations to compute V_n. In our Python implementation, the limiting factor is the computation time of the kernel matrix. When $n \geq 1500$ and K_2 is used (empirically, the eigenvalues of the kernel matrix have a fast decrease), we use an incomplete Cholesky decomposition (Bach and Jordan, 2002) with maximal rank 100 to approximate the kernel matrix. It is computed online with a fast Cython implementation, and does not require the compute the whole kernel matrix. Overall, the CPU time for computing V_n for $n = 2000$ is approximately 20 seconds on a standard laptop. Running all the experiments of this chapter took a few hours.

Conclusion and Research Directions

Summary of the Thesis

Modern applications of control problems, such as polyarticulated robotics, are computationally challenging. One must deal with uncertainty, model misspecification, and large-dimensional systems, while many existing numerical methods are subject to the curse of dimensionality. Guided by such constraints, we have proposed new algorithms, or adapted existing ones, for solving different problems in control and reinforcement learning: assessing the stability of a trajectory, evaluating a policy, or approximating the optimal value function.

In this manuscript, we have tried to think in the same way about control and reinforcement learning problems, regardless of the differences in notations. In particular, we have progressively relaxed the degree of knowledge about the model: from a perfectly known model in Chapter 3, to a model known up to a certain order in Chapter 4, and finally to a model known only through samples in Chapters 5 and 6. In this sense, we have moved from a *control* paradigm to a *learning to control* paradigm. In the latter, the learning and control problems are interlaced, from the *model-based* setting, where the model is first learnt and then used as is to solve a control problem, to the *model-free* setting, where no model is explicitly learnt, as is the case in the last two chapters. In these chapters, we have used kernel methods, a set of tools primarily designed for learning problems, which are well-suited to sample-based algorithms and which often come with theoretical guarantees. Following the recent works of Diwale (2019) and Aubin-Frankowski (2021a), we believe kernel methods to be a promising direction for designing theoretically-grounded numerical methods in sample-based control.

For each algorithm considered, we have provided small-scale numerical examples, showing their potential merits and limitations. Further work is needed to make these methods readily applicable to large-scale, real-world applications.

Perspectives

Beyond direct generalizations mentioned as in the conclusion of each chapter, we have encountered throughout this work several open questions that we briefly describe hereafter.

Modeling the value function. The value function is an ubiquitous object in optimal control and reinforcement learning. In large-scale problems, it cannot be represented exactly, and must be somehow approximated. The most natural representation is probably linear, in some basis of functions or RKHS. However, contrary to many machine learning applications, the function to be approximated has little regularity. In particular, it is generally not smooth. This questions the relevance of modeling the value function in a space of smooth functions, like a Sobolev space. As briefly discussed in Chapter 5, it might be worth exploring the regularizing effect of a stochastic perturbation of the dynamics. Furthermore, as we have seen in Chapter 3, a max-plus linear representation might be better suited than a linear representation for the value function, given its interesting properties of compatibility with the Bellman operator.

Statistical hardness of control problems. When one is provided with n samples of a dynamics and cost, how hard is it to actually solve the optimal control problem up to a certain precision? And what is the precision of the approximation produced by a given algorithm? We have encountered this question in Chapter 5. It should be studied to provably exhibit a potential advantage of the method we proposed over the linear programming baseline. The number of samples required to reach a certain precision is called the *sample complexity* of a problem, and has been recently explored for the linear quadratic regulator by Dean et al. (2020). On the one hand, it is important to study lower-bounds on the number of observations, *i.e.*, do a worst-case analysis, depending on the intrinsic complexity of the control problem. This entails describing classes of statistically easy or hard problems. On the other hand, we want to derive upper-bounds, in order to evaluate a given sample-based method. To achieve this, one must relate the control problem to a statistical setting.

Relating the quality of the value function and the controller. A recurring question in control and reinforcement learning is whether a controller or policy derived from a given approximated value function will perform well. We have encountered this issue in our numerical experiments, where some apparently good value functions led to poor controllers or policies. More precisely, what is the relation between the quality of the value function, *e.g.*, in terms of distance to the optimal value function, or in terms of Bellman error, and the cost of the corresponding controller? Finding the right metrics is mostly an open problem. For instance, Lu et al. (2021b) questioned the relevance of the projected Bellman error for generating a good policy, and Fujimoto et al. (2022) showed that the Bellman error is a poor predictor of the distance to the optimal value function.

⋆

⋆ ⋆

This is Major Tom to Ground Control (...)
Though I'm past one hundred thousand miles
I'm feeling very still
And I think my spaceship knows which way to go...

David Bowie, *Space Oddity*, 1969.

Bibliography

Akian, M. and Fodjo, E. (2017). From a monotone probabilistic scheme to a probabilistic max-plus algorithm for solving Hamilton-Jacobi-Bellman equations. *arXiv preprint arXiv:1709.09049*. [Cited on p. 80.]

Akian, M., Gaubert, S., and Lakhoua, A. (2008). The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis. *SIAM Journal on Control and Optimization*, 47(2):817–848. [Cited on pp. 4, 10, 25, 63, 66, 67, 69, 70, 71, 74, 75, and 76.]

Allgower, E. L. and Georg, K. (2003). *Introduction to Numerical Continuation Methods*. SIAM. [Cited on p. 25.]

Andersson, J. A., Gillis, J., Horn, G., Rawlings, J. B., and Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36. [Cited on p. 24.]

Arnold, V. (1966). Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications à l'hydrodynamique des fluides parfaits. In *Annales de l'Institut Fourier*, volume 16, pages 319–361. [Cited on p. 37.]

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404. [Cited on pp. 54 and 104.]

Asmussen, S. (2003). *Applied Probability and Queues*, volume 2. Springer. [Cited on p. 123.]

Aubin, J.-P. and Cellina, A. (1984). *Differential Inclusions: Set-Valued Maps and Viability Theory*. Springer. [Cited on p. 86.]

Aubin-Frankowski, P.-C. (2021a). *Estimation and Control under Constraints through Kernel Methods*. PhD thesis, Université Paris Sciences et Lettres. [Cited on p. 161.]

Aubin-Frankowski, P.-C. (2021b). Interpreting the dual Riccati equation through the LQ reproducing kernel. *Comptes Rendus. Mathématique*, 359(2):199–204. [Cited on p. 55.]

Aubin-Frankowski, P.-C. and Gaubert, S. (2022). Tropical reproducing kernels and optimization. *arXiv preprint arXiv:2202.11410*. [Cited on p. 63.]

Baccelli, F., Cohen, G., Olsder, G. J., and Quadrat, J.-P. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*. John Wiley & Sons Ltd. [Cited on pp. 62 and 63.]

Bach, F. (2019). Max-plus matching pursuit for deterministic Markov decision processes. *arXiv preprint arXiv:1906.08524*. [Cited on pp. 68, 69, 71, 74, and 77.]

Bach, F. (2022). Information theory with kernel methods. *arXiv preprint arXiv:2202.08545*. [Cited on pp. 58 and 121.]

Bach, F. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48. [Cited on pp. 132 and 159.]

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the International Conference on Machine Learning*, pages 30–37. [Cited on p. 121.]

Baker, C. R. (1973). Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289. [Cited on pp. 121 and 125.]

Baldi, L. and Mourrain, B. (2021). On moment approximation and the effective Putinar's Positivstellensatz. *arXiv preprint arXiv:2111.11258*. [Cited on p. 106.]

Bar-Shalom, Y. and Tse, E. (1974). Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500. [Cited on p. 31.]

Barreto, A., Precup, D., and Pineau, J. (2011). Reinforcement learning using kernel-based stochastic factorization. *Advances in Neural Information Processing Systems*, 24. [Cited on p. 122.]

Barreto, A., Precup, D., and Pineau, J. (2016). Practical kernel-based reinforcement learning. *Journal of Machine Learning Research*, 17(1):2372–2441. [Cited on p. 122.]

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515. [Cited on p. 18.]

Bellman, R. (1957a). *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition. [Cited on p. 27.]

Bellman, R. (1957b). A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684. [Cited on pp. 2 and 8.]

Benveniste, A., Métivier, M., and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*, volume 22. Springer Science & Business Media. [Cited on p. 122.]

Berlinet, A. and Thomas-Agnan, C. (2011). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer. [Cited on p. 107.]

Bernstein, A. and Shimkin, N. (2008). Adaptive aggregation for reinforcement learning with efficient exploration: Deterministic domains. In *COLT*, pages 323–334. [Cited on pp. 76 and 77.]

Berry, D. A. and Fristedt, B. (1985). *Bandit Problems: Sequential Allocation of Experiments*, volume 5. Springer. [Cited on p. 29.]

Berthier, R., Bach, F., and Gaillard, P. (2020). Tight nonparametric convergence rates for stochastic gradient descent under the noiseless linear model. *Advances in Neural Information Processing Systems*, 33:2576–2586. [Cited on pp. 58 and 122.]

Bertsekas, D. P. (2011). *Dynamic Programming and Optimal Control, 3rd edition, volume ii*. Belmont, MA: Athena Scientific. [Cited on pp. 18, 27, and 33.]

Bertsekas, D. P. (2019). *Reinforcement Learning and Optimal Control*. Athena Scientific. [Cited on pp. 2 and 8.]

Betts, J. T. (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. SIAM. [Cited on p. 23.]

Bhandari, J., Russo, D., and Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. In *Conference on Learning Theory*, pages 1691–1692. [Cited on pp. 32, 121, 124, 125, 128, 129, 130, 131, and 149.]

Bhat, N., Farias, V., and Moallemi, C. C. (2012). Non-parametric approximate dynamic programming via the kernel method. *Advances in Neural Information Processing Systems*, 25. [Cited on p. 122.]

Bhatia, R. (2013). *Matrix Analysis*, volume 169. Springer Science & Business Media. [Cited on p. 125.]

Bock, H. and Plitt, K. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2):1603–1608. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984. [Cited on p. 24.]

Boltyanski, V. G., Gamkrelidze, R. V., Mishchenko, E. F., and Pontryagin, L. S. (1960). The maximum principle in the theory of optimal processes of control. *IFAC Proceedings Volumes*, 1(1):464–469. [Cited on p. 15.]

Bonnans, J.-F. (2019). Lecture notes on optimal control. *ENSTA Paris Tech and Optimization Master, Université Paris-Saclay*. [Cited on p. 24.]

Bonnans, J.-F., Gilbert, J. C., Lemaréchal, C., and Sagastizábal, C. A. (2006). *Numerical Optimization: Theoretical and Practical Aspects*. Springer Science & Business Media. [Cited on pp. 24 and 25.]

Borgwardt, K. M. and Kriegel, H.-P. (2005). Shortest-path kernels on graphs. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE. [Cited on p. 55.]

Borkar, V. S. (2009). *Stochastic Approximation: a Dynamical Systems Viewpoint*, volume 48. Springer. [Cited on p. 57.]

Borkar, V. S. and Meyn, S. P. (2000). The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469. [Cited on pp. 31, 57, 58, 122, and 127.]

Borkar, V. S. and Soumyanatha, K. (1997). An analog scheme for fixed point computation. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 44(4):351–355. [Cited on p. 129.]

Boscain, U. and Piccoli, B. (2005). An introduction to optimal control. *Contrôle Non Linéaire et Applications, Herman, Paris*, pages 19–66. [Cited on p. 17.]

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems*, 20. [Cited on p. 57.]

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311. [Cited on p. 122.]

Bourlès, H. and Kwan, G. K. (2013). *Linear Systems*. John Wiley & Sons. [Cited on pp. 1 and 7.]

Boyan, J. and Moore, A. (1994). Generalization in reinforcement learning: Safely approximating the value function. *Advances in Neural Information Processing Systems*, 7. [Cited on p. 120.]

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15. SIAM. [Cited on pp. 86 and 88.]

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press. [Cited on pp. 15, 25, 44, and 114.]

Bradtke, S. J. and Barto, A. G. (1996). Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1):33–57. [Cited on p. 120.]

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). OpenAI gym. *arXiv preprint arXiv:1606.01540*. [Cited on pp. 28, 77, and 87.]

Brogliato, B. (1999). *Nonsmooth Mechanics*, volume 3. Springer. [Cited on p. 104.]

Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. (2019). Neural temporal-difference learning converges to global optima. *Advances in Neural Information Processing Systems*, 32. [Cited on pp. 120 and 122.]

Caponigro, M., Ghezzi, R., Piccoli, B., and Trélat, E. (2018). Regularization of chattering phenomena via bounded variation controls. *IEEE Transactions on Automatic Control*, 63(7):2046–2060. [Cited on p. 18.]

Caponnetto, A. and De Vito, E. (2007). Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368. [Cited on pp. 122 and 127.]

Capuzzo Dolcetta, I. (1983). On a discrete approximation of the Hamilton-Jacobi equation of dynamic programming. *Applied Mathematics and Optimization*, 10(1):367–377. [Cited on p. 75.]

Carpentier, J. and Mansard, N. (2018a). Analytical derivatives of rigid body dynamics algorithms. In *Robotics: Science and Systems*. [Cited on pp. 92 and 96.]

Carpentier, J. and Mansard, N. (2018b). Multicontact locomotion of legged robots. *IEEE Transactions on Robotics*, 34(6):1441–1460. [Cited on p. 84.]

Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamiraux, F., Stasse, O., and Mansard, N. (2019). The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*. [Cited on pp. 39 and 95.]

Chandrashekar, L. and Bhatnagar, S. (2014). Approximate dynamic programming with (min;+) linear function approximation for Markov decision processes. In *53rd IEEE Conference on Decision and Control*, pages 1588–1593. IEEE. [Cited on p. 69.]

Chen, Z., Maguluri, S. T., Shakkottai, S., and Shanmugam, K. (2020). Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes. *Advances in Neural Information Processing Systems*, 33:8223–8234. [Cited on p. 129.]

Cheney, E. W. (2001). *Analysis for Applied Mathematics*, volume 1. Springer. [Cited on p. 125.]

Cheney, E. W. and Light, W. A. (2009). *A Course in Approximation Theory*, volume 101. American Mathematical Society. [Cited on p. 52.]

Chesi, G. (2004). Estimating the domain of attraction for uncertain polynomial systems. *Automatica*, 40(11):1981–1986. [Cited on p. 84.]

Chui, C. K. and Chen, G. (1987). *Kalman Filtering with Real-time Applications*. Springer Berlin, Heidelberg. [Cited on pp. 15 and 23.]

Cohen, G., Gaubert, S., and Quadrat, J.-P. (1999). Max-plus algebra and system theory: where we are and where to go now. *Annual Reviews in Control*, 23:207–219. [Cited on p. 63.]

Cohen, G., Gaubert, S., and Quadrat, J.-P. (2004). Duality and separation theorems in idempotent semimodules. *Linear Algebra and its Applications*, 379:395–422. [Cited on p. 68.]

Crandall, M. G., Evans, L. C., and Lions, P.-L. (1984). Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502. [Cited on p. 19.]

Crandall, M. G. and Lions, P.-L. (1983). Viscosity solutions of Hamilton-Jacobi equations. *Trans. Am. Math. Soc.*, 277(1):1–42. [Cited on pp. 105 and 112.]

Cristianini, N. and Shawe-Taylor, J. (2004). *Kernel Methods for Pattern Analysis*, volume 173. Cambridge University Press. [Cited on p. 121.]

Cucker, F. and Smale, S. (2002). On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39(1):1–49. [Cited on p. 127.]

Cucker, F. and Zhou, D. X. (2007). *Learning Theory: an Approximation Theory Viewpoint*, volume 24. Cambridge University Press. [Cited on pp. 122 and 127.]

Dai, B., He, N., Pan, Y., Boots, B., and Song, L. (2017). Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pages 1458–1467. [Cited on p. 122.]

Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. (2018). Finite sample analyses for TD(0) with function approximation. *Proceedings of AAAI'18/IAAI'18/EAAI'18*. [Cited on p. 121.]

Dayan, P. (1992). The convergence of TD(λ) for general λ. *Machine Learning*, 8(3):341–362. [Cited on p. 121.]

De Farias, D. P. and Van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865. [Cited on p. 115.]

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2020). On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679. [Cited on pp. 108 and 162.]

Défossez, A. and Bach, F. (2017). Adabatch: Efficient gradient aggregation rules for sequential and parallel stochastic gradient methods. *arXiv preprint arXiv:1711.01761*. [Cited on p. 129.]

Della Santina, C., Duriez, C., and Rus, D. (2021). Model based control of soft robots: A survey of the state of the art and open challenges. *arXiv preprint arXiv:2110.01358*. [Cited on pp. 104 and 118.]

Diehl, M., Bock, H. G., Diedam, H., and Wieber, P.-B. (2006). Fast direct multiple shooting algorithms for optimal robot control. In *Fast Motions in Biomechanics and Robotics*, pages 65–93. Springer. [Cited on p. 23.]

Diehl, M. and Gros, S. (2011). Numerical optimal control. *Optimization in Engineering Center (OPTEC)*. [Cited on pp. 23 and 25.]

Dietterich, T. and Wang, X. (2001). Batch value function approximation via support vectors. *Advances in Neural Information Processing Systems*, 14. [Cited on p. 122.]

Dieuleveut, A. (2017). *Stochastic Approximation in Hilbert Spaces*. PhD thesis, Paris Sciences et Lettres (ComUE). [Cited on p. 125.]

Dieuleveut, A. and Bach, F. (2016). Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399. [Cited on pp. 58, 122, 124, 125, 127, and 138.]

Diwale, S. S. (2019). *Kernel Methods and Model Predictive Approaches for Learning and Control*. PhD thesis, EPFL. [Cited on p. 161.]

Domingues, O. D., Ménard, P., Pirotta, M., Kaufmann, E., and Valko, M. (2021). Kernel-based reinforcement learning: A finite-time analysis. In *International Conference on Machine Learning*, pages 2783–2792. [Cited on p. 122.]

Dormand, J. R. and Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26. [Cited on p. 22.]

Duan, Y., Wang, M., and Wainwright, M. J. (2021). Optimal policy evaluation using kernel-based temporal difference methods. *arXiv preprint arXiv:2109.12002*. [Cited on p. 122.]

Durmus, A., Moulines, E., Naumov, A., Samsonov, S., and Wai, H.-T. (2021). On the stability of random matrix product with Markovian noise: Application to linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pages 1711–1752. [Cited on p. 130.]

Durrett, R. (2019). *Probability: Theory and Examples*, volume 49. Cambridge University Press. [Cited on p. 123.]

Evans, L. C. (2010). *Partial Differential Equations*, volume 19. American Mathematical Society. [Cited on pp. 19 and 20.]

Falcone, M. (1987). A numerical approach to the infinite horizon problem of deterministic control theory. *Applied Mathematics and Optimization*, 15(1):1–13. [Cited on p. 75.]

Falcone, M. and Ferretti, R. (2013). *Semi-Lagrangian Approximation Schemes for Linear and Hamilton—Jacobi Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA. [Cited on p. 75.]

Farahmand, A.-M., Ghavamzadeh, M., Szepesvári, C., and Mannor, S. (2016). Regularized policy iteration with nonparametric function spaces. *Journal of Machine Learning Research*, 17(1):4809–4874. [Cited on p. 122.]

Fasshauer, G. E. (2011). Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63. [Cited on p. 54.]

Featherstone, R. (1983). The calculation of robot dynamics using articulated-body inertias. *The International Journal of Robotics Research*, 2(1):13–30. [Cited on p. 38.]

Featherstone, R. and Orin, D. E. (2008). Handbook of robotics chapter 3: Dynamics. *Handbook of Robotics, Springer*. [Cited on p. 37.]

Febbo, H., Jayakumar, P., Stein, J. L., and Ersal, T. (2020). NLOptControl: A modeling language for solving optimal control problems. *arXiv preprint arXiv:2003.00142*. [Cited on p. 24.]

Finkel, R. and Bentley, J. (1974). Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9. [Cited on p. 77.]

Fleming, W. H. and McEneaney, W. M. (2000). A max-plus-based algorithm for a Hamilton–Jacobi–Bellman equation of nonlinear filtering. *SIAM Journal on Control and Optimization*, 38(3):683–710. [Cited on p. 64.]

Fleming, W. H. and Rishel, R. W. (2012). *Deterministic and Stochastic Optimal Control*, volume 1. Springer Science & Business Media. [Cited on pp. 28 and 112.]

Fleming, W. H. and Soner, H. M. (2006). *Controlled Markov Processes and Viscosity Solutions*, volume 25. Springer Science & Business Media. [Cited on pp. 25 and 75.]

Fujimoto, S., Meger, D., Precup, D., Nachum, O., and Gu, S. S. (2022). Why should I trust you, Bellman? the Bellman error is a poor replacement for value error. *arXiv preprint arXiv:2201.12417*. [Cited on p. 162.]

Fukumizu, K., Bach, F., and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99. [Cited on p. 125.]

Fuller, A. T. (1963). Study of an optimum non-linear control system. *Journal of Electronics and Control*, 15(1):63–71. [Cited on p. 18.]

Gaitsgory, V., Parkinson, A., and Shvartsman, I. (2017). Linear programming formulations of deterministic infinite horizon optimal control problems in discrete time. *Discrete & Continuous Dynamical Systems*, 22(10):3821–3838. [Cited on p. 108.]

Gaitsgory, V. and Quincampoix, M. (2009). Linear programming approach to deterministic infinite horizon optimal control problems with discounting. *SIAM Journal on Control and Optimization*, 48:2480–2512. [Cited on p. 108.]

Gamkrelidze, R. V. (1999). Discovery of the maximum principle. *Journal of Dynamical and Control Systems*, 5:437–451. [Cited on p. 15.]

Gaubert, S., McEneaney, W., and Qu, Z. (2011). Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms. In *2011 50th IEEE Conference on Decision and Control*, pages 1054–1061. IEEE. [Cited on p. 76.]

Gaubert, S. and Plus, M. (1997). Methods and applications of (max,+) linear algebra. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 261–282. Springer. [Cited on pp. 60, 62, and 67.]

Giesl, P. and Hafstein, S. (2015). Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems-Series B*, 20(8):2291–2331. [Cited on p. 84.]

Giftthaler, M., Neunert, M., Stäuble, M., Frigerio, M., Semini, C., and Buchli, J. (2017). Automatic differentiation of rigid body dynamics for optimal control and estimation. *Advanced Robotics*, 31(22):1225–1237. [Cited on p. 84.]

Glover, K., Sefton, J., and McFarlane, D. C. (1990). A tutorial on loop shaping using H-infinity robust stabilization. *IFAC Proceedings Volumes*, 23(8):117–126. [Cited on p. 84.]

Gonçalves, V. M. (2021). Max-plus approximation for reinforcement learning. *Automatica*, 129:109623. [Cited on p. 80.]

Grünewälder, S., Lever, G., Baldassarre, L., Pontil, M., and Gretton, A. (2012). Modelling transition dynamics in MDPs with RKHS embeddings. In *International Conference on Machine Learning*. [Cited on p. 122.]

Guo, X. and Hernández-Lerma, O. (2009). Continuous-time Markov decision processes. In *Continuous-Time Markov Decision Processes*. Springer. [Cited on p. 28.]

Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288. [Cited on p. 124.]

Heinonen, M. and d'Alché Buc, F. (2014). Learning nonparametric differential equations with operator-valued kernels and gradient matching. *arXiv preprint arXiv:1411.5172*. [Cited on p. 55.]

Henrion, D. (2014). Optimization on linear matrix inequalities for polynomial systems control. *Lecture notes at the 35th International Summer School on Automatic Control, Grenoble, France*. [Cited on pp. 40, 43, 44, 50, and 106.]

Henrion, D., Lasserre, J.-B., and Löfberg, J. (2009). Gloptipoly 3: Moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779. [Cited on p. 44.]

Hernández-Hernández, D., Hernández-Lerma, O., and Taksar, M. (1996). The linear programming approach to deterministic optimal control problems. *Applicationes Mathematicae*, 24(1):17–33. [Cited on p. 104.]

Hernández-Lerma, O. and Lasserre, J. B. (2012). *Discrete-time Markov Control Processes: Basic Optimality Criteria*, volume 30. Springer Science & Business Media. [Cited on p. 66.]

Hespanha, J. P. (2018). *Linear Systems Theory*. Princeton Press. [Cited on p. 15.]

Hu, B. and Syed, U. (2019). Characterizing the exact behaviors of temporal difference learning algorithms using Markov jump linear system theory. *Advances in Neural Information Processing Systems*, 32. [Cited on p. 121.]

Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 11. [Cited on p. 55.]

Jaakkola, T., Jordan, M., and Singh, S. (1993). Convergence of stochastic iterative dynamic programming algorithms. *Advances in Neural Information Processing Systems*, 6. [Cited on p. 121.]

Jallet, W., Bambade, A., Mansard, N., and Carpentier, J. (2022). Constrained Differential Dynamic Programming: A primal-dual augmented Lagrangian approach. *hal-03597630*. working paper or preprint. [Cited on p. 14.]

Johansen, T. A. (2000). Computation of Lyapunov functions for smooth nonlinear systems using convex optimization. *Automatica*, 36(11):1617–1626. [Cited on p. 84.]

Jones, M. and Peet, M. M. (2019). Relaxing the Hamilton Jacobi Bellman equation to construct inner and outer bounds on reachable sets. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2397–2404. IEEE. [Cited on p. 105.]

Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear Estimation*. Prentice Hall. [Cited on p. 15.]

Kalman, R. E. (1960). On the general theory of control systems. In *Proceedings First International Conference on Automatic Control, Moscow, USSR*, pages 481–492. [Cited on p. 15.]

Kamoutsi, A., Sutter, T., Esfahani, P. M., and Lygeros, J. (2017). On infinite linear programming and the moment approach to deterministic infinite horizon discounted optimal control problems. *IEEE Control Systems Letters*, 1(1):134–139. [Cited on p. 105.]

Klenke, A. (2013). *Probability Theory: A Comprehensive Course*. Springer Science & Business Media. [Cited on p. 122.]

Koppel, A., Warnell, G., Stump, E., Stone, P., and Ribeiro, A. (2020). Policy evaluation in continuous MDPs with efficient kernelized gradient temporal difference. *IEEE Transactions on Automatic Control*, 66(4):1856–1863. [Cited on pp. 122 and 123.]

Korda, M., Henrion, D., and Jones, C. N. (2017). Convergence rates of moment-sum-of-squares hierarchies for optimal control problems. *Systems & Control Letters*, 100:1–5. [Cited on p. 106.]

Korda, N. and La, P. (2015). On TD(0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In *International Conference on Machine Learning*, pages 626–634. [Cited on p. 121.]

Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor, J. L. (2016). *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM. [Cited on p. 104.]

Kwakernaak, H. and Sivan, R. (1969). *Linear Optimal Control Systems*, volume 1072. Wiley. [Cited on pp. 15, 21, and 22.]

Lakshminarayanan, C. and Szepesvari, C. (2018). Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *International Conference on Artificial Intelligence and Statistics*, pages 1347–1355. [Cited on p. 121.]

Lamiraux, F. and Mirabel, J. (2014). *Humanoid Path Planner*. Available from `https://humanoid-path-planner.github.io/hpp-doc/`. [Cited on p. 37.]

Lasserre, J.-B. (2010). *Moments, Positive Polynomials and their Applications*, volume 1. World Scientific. [Cited on pp. 40, 41, 105, and 107.]

Lasserre, J.-B. (2015). *An Introduction to Polynomial and Semi-Algebraic Optimization*, volume 52. Cambridge University Press. [Cited on pp. 40, 43, 50, 104, and 106.]

Lasserre, J.-B., Henrion, D., Prieur, C., and Trélat, E. (2008). Nonlinear optimal control via occupation measures and lmi-relaxations. *SIAM Journal on Control and Optimization*, 47(4):1643–1666. [Cited on pp. 5, 11, 49, 104, 106, 107, and 118.]

Lattimore, T. and Szepesvári, C. (2020). *Bandit Algorithms*. Cambridge University Press. [Cited on p. 29.]

LaValle, S. M. and Kuffner, J. J. (2001). Rapidly-exploring random trees: progress and prospects. *Algorithmic and Computational Robotics: New Directions*, pages 293–308. [Cited on p. 84.]

LeVeque, R. J. (1992). *Numerical Methods for Conservation Laws*, volume 214. Springer. [Cited on pp. 20 and 25.]

Levin, D. A. and Peres, Y. (2017). *Markov Chains and Mixing Times*, volume 107. American Mathematical Society. [Cited on p. 130.]

Li, W. and Todorov, E. (2007). Iterative linear quadratic regulator design for nonlinear biological movement systems. *International Journal of Control*, 80(9):1439 – 1453. [Cited on p. 21.]

Liberzon, D. (2011). *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press. [Cited on pp. 13, 14, 16, 20, 21, 23, 70, 85, 93, 94, and 105.]

Lieberman, G. M. (1996). *Second Order Parabolic Differential Equations*. World scientific. [Cited on p. 112.]

Lions, P.-L. (2015). Cours sur le contrôle des modèles dynamiques. *École polytechnique*. [Cited on p. 19.]

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444. [Cited on p. 55.]

Long, J., Han, J., and E, W. (2021). An L^2 analysis of reinforcement learning in high dimensions with kernel and neural network approximation. *arXiv preprint arXiv:2104.07794*. [Cited on p. 122.]

Lu, F., Mehta, P. G., Meyn, S. P., and Neu, G. (2021a). Convex Q-learning. In *2021 American Control Conference (ACC)*. [Cited on p. 108.]

Lu, F., Mehta, P. G., Meyn, S. P., and Neu, G. (2021b). Convex Q-learning. In *2021 American Control Conference (ACC)*, pages 4749–4756. IEEE. [Cited on p. 162.]

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. (1980). On-line computational scheme for mechanical manipulators. *Trans. ASME J. Dynamic Syst., Measurement, and Control*, 102(2):69–76. [Cited on p. 38.]

Lyapunov, A. M. (1992). The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534. [Cited on p. 46.]

Mairal, J. and Vert, J.-P. (2018). Machine learning with kernel methods. *Lecture Notes, January*, 10. [Cited on pp. 52 and 54.]

Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415. [Cited on p. 77.]

Marteau-Ferey, U., Bach, F., and Rudi, A. (2020). Non-parametric models for non-negative functions. *Advances in Neural Information Processing Systems*, 33:12816–12826. [Cited on pp. 5, 11, 59, 104, 107, and 113.]

Maslov, V. P. (1973). *Operational Methods*. Mir Publishers Moscow. [Cited on p. 64.]

Mayne, D. (1966). A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95. [Cited on p. 21.]

McEneaney, W. M. (2003). Max-plus eigenvector representations for solution of nonlinear H infinity problems: basic concepts. *IEEE Transactions on Automatic Control*, 48(7):1150–1163. [Cited on pp. 4, 10, 63, 66, 67, 69, 71, and 76.]

Mehta, P. and Meyn, S. (2009). Q-learning and pontryagin's minimum principle. In *Proceedings of the 48h IEEE Conference on Decision and Control*, pages 3598–3605. IEEE. [Cited on p. 71.]

Meyn, S. (2022). *Control Systems and Reinforcement Learning*. Cambridge University Press. [Cited on pp. 2 and 8.]

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7(12). [Cited on p. 121.]

Mockus, J. (2012). *Bayesian Approach to Global Optimization: Theory and Applications*, volume 37. Springer Science & Business Media. [Cited on p. 93.]

Moerland, T. M., Broekens, J., and Jonker, C. M. (2020). Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*. [Cited on p. 31.]

Mou, W., Pananjady, A., and Wainwright, M. J. (2020). Optimal oracle inequalities for solving projected fixed-point equations. *arXiv preprint arXiv:2012.05299*. [Cited on p. 121.]

Munos, R. (2000). A study of reinforcement learning in the continuous case by the means of viscosity solutions. *Machine Learning*, 40(3):265–299. [Cited on p. 25.]

Munos, R. and Moore, A. (2002). Variable resolution discretization in optimal control. *Machine Learning*, 49(2-3):291–323. [Cited on pp. 25 and 76.]

Murray, R. M., Li, Z., and Sastry, S. S. (2017). *A Mathematical Introduction to Robotic Manipulation*. CRC press. [Cited on pp. 37 and 104.]

Nagaraj, D., Wu, X., Bresler, G., Jain, P., and Netrapalli, P. (2020). Least squares regression with Markovian data: Fundamental limits and algorithms. *Advances in Neural Information Processing Systems*. [Cited on pp. 122 and 130.]

Nemirovski, A. (2004). Interior point polynomial time methods in convex programming. *Lecture notes*. [Cited on p. 114.]

Nesterov, Y. and Nemirovskii, A. (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM. [Cited on p. 88.]

Novak, E. (2006). *Deterministic and Stochastic Error Bounds in Numerical Analysis*. Springer. [Cited on p. 108.]

Novak, E., Ullrich, M., Woźniakowski, H., and Zhang, S. (2018). Reproducing kernels of Sobolev spaces on $\mathbb{R}^d$ and applications to embedding constants and tractability. *Analysis and Applications*, 16(05):693–715. [Cited on p. 121.]

Olver, F. W. J., Lozier, D. W., Boisvert, R. F., and Clark, C. W. (2010). *NIST Handbook of Mathematical Functions*. Cambridge University Press. [Cited on p. 132.]

Ormoneit, D. and Sen, Ś. (2002). Kernel-based reinforcement learning. *Machine Learning*, 49(2):161–178. [Cited on pp. 122 and 123.]

Papachristodoulou, A., Anderson, J., Valmorbida, G., Prajna, S., Seiler, P., Parrilo, P. A., Peet, M. M., and Jagt, D. (2021). *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. Available from `https://github.com/oxfordcontrol/SOSTOOLS`. [Cited on p. 43.]

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32. [Cited on pp. 32, 92, and 96.]

Paulsen, V. I. and Raghupathi, M. (2016). *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press. [Cited on p. 106.]

Pillaud-Vivien, L. (2020). *Apprentissage par Noyaux Reproduisants: Descente de Gradient Stochastique et Estimation de Laplacien*. PhD thesis, Université Paris Sciences et Lettres. [Cited on p. 33.]

Pillaud-Vivien, L., Rudi, A., and Bach, F. (2018a). Exponential convergence of testing error for stochastic gradient methods. In *Conference on Learning Theory*, pages 250–296. [Cited on p. 122.]

Pillaud-Vivien, L., Rudi, A., and Bach, F. (2018b). Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. *Advances in Neural Information Processing Systems*, 31. [Cited on p. 58.]

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855. [Cited on p. 128.]

Pontryagin, L. S., Boltyanski, V. G., and Gamkrelidze, R. V. (1974). *Théorie Mathématique des Processus Optimaux*. Éditions Mir. [Cited on p. 16.]

Powell, W. B. and Ma, J. (2011). A review of stochastic algorithms with continuous value function approximation and some new approximate policy iteration algorithms for multidimensional continuous applications. *Journal of Control Theory and Applications*, 9(3):336–352. [Cited on p. 28.]

Puterman, M. L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons. [Cited on pp. 28 and 33.]

Putinar, M. (1993). Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42:969–984. [Cited on p. 41.]

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. (2020). Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*. [Cited on p. 25.]

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. (2005). Graph kernels for chemical informatics. *Neural Networks*, 18(8):1093–1110. Neural Networks and Kernel Methods for Structured Domains. [Cited on p. 55.]

Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528. [Cited on pp. 23 and 24.]

Reiss, R.-D. (2012). *A Course on Point Processes*. Springer Science & Business Media. [Cited on p. 122.]

Ross, I. M. and Karpenko, M. (2012). A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, 36(2):182–197. [Cited on p. 24.]

Rudi, A. and Ciliberto, C. (2021). PSD representations for effective probability models. *Advances in Neural Information Processing Systems*, 34:19411–19422. [Cited on p. 59.]

Rudi, A., Marteau-Ferey, U., and Bach, F. (2020). Finding global minima via kernel approximations. *arXiv preprint arXiv:2012.11978*. [Cited on pp. 59, 106, 107, 108, 111, 113, and 114.]

Rudin, W. (1987). *Real and Complex Analysis, 3rd Ed.* McGraw-Hill, Inc., USA. [Cited on p. 139.]

Safonov, M. G. (2012). Origins of robust control: Early history and future speculations. *Annual Reviews in Control*, 36(2):173–181. [Cited on p. 31.]

Samuelson, P. A. (1972). The general saddlepoint property of optimal-control motions. *Journal of Economic Theory*, 5(1):102–120. [Cited on p. 23.]

Schapire, R. E. and Warmuth, M. K. (1996). On the worst-case analysis of temporal-difference learning algorithms. *Machine Learning*, 22(1):95–121. [Cited on p. 121.]

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer. [Cited on p. 55.]

Schölkopf, B. and Smola, A. J. (2001). *Learning with Kernels*. MIT Press. [Cited on p. 52.]

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Kernel Methods in Computational Biology*. MIT Press. [Cited on p. 55.]

Schwartz, L. (1981). *Cours d'Analyse*, volume 1. Hermann. [Cited on p. 111.]

Sethian, J. A. (1999). *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, volume 3. Cambridge University Press. [Cited on p. 25.]

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press. [Cited on pp. 30 and 32.]

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press. [Cited on p. 52.]

Siciliano, B., Khatib, O., and Kröger, T. (2008). *Springer Handbook of Robotics*, volume 200. Springer. [Cited on p. 37.]

Simon, I. (1978). Limited subsets of a free monoid. In *19th Annual Symposium on Foundations of Computer Science*, pages 143–150. IEEE Computer Society. [Cited on p. 63.]

Singh, S., Chen, M., Herbert, S. L., Tomlin, C. J., and Pavone, M. (2018). Robust tracking with model mismatch for fast and safe planning: an SOS optimization approach. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 545–564. Springer. [Cited on p. 84.]

Slotine, J.-J. E. and Li, W. (1991). *Applied Nonlinear Control*, volume 199. Prentice Hall Englewood Cliffs, NJ. [Cited on pp. 46, 85, and 127.]

Sobol', I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comp. Math. Math. Phys.* [Cited on p. 115.]

Srikant, R. and Ying, L. (2019). Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pages 2803–2830. [Cited on pp. 121 and 131.]

Stasse, O., Flayols, T., Budhiraja, R., Giraud-Esclasse, K., Carpentier, J., Mirabel, J., Del Prete, A., Souères, P., Mansard, N., Lamiraux, F., Laumond, J.-P., Marchionni, L., Tome, H., and Ferro, F. (2017). TALOS: A new humanoid research platform targeted for industrial applications. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 689–695. IEEE. [Cited on p. 39.]

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(Nov):67–93. [Cited on p. 127.]

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44. [Cited on pp. 31, 120, and 121.]

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems*, pages 1038–1044. [Cited on p. 95.]

Sutton, R. S. (2015). Introduction to reinforcement learning with function approximation. In *Tutorial at the Conference on Neural Information Processing Systems*, page 33. [Cited on p. 120.]

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press. [Cited on pp. 13, 26, 28, 30, 31, 66, 71, 106, 120, and 124.]

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12. [Cited on p. 31.]

Tallec, C., Blier, L., and Ollivier, Y. (2019). Making deep Q-learning methods robust to time discretization. In *International Conference on Machine Learning*, pages 6096–6104. [Cited on p. 75.]

Tarres, P. and Yao, Y. (2014). Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9):5716–5735. [Cited on p. 129.]

Tedrake, R., Manchester, I. R., Tobenkin, M., and Roberts, J. W. (2010). LQR-trees: Feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052. [Cited on pp. 47, 84, and 94.]

Tobenkin, M. M., Manchester, I. R., and Tedrake, R. (2011). Invariant funnels around trajectories using sum-of-squares programming. *IFAC Proceedings Volumes*, 44(1):9218–9223. [Cited on pp. 94, 95, 96, and 97.]

Topcu, U. and Packard, A. (2007). Stability region analysis for uncertain nonlinear systems. In *IEEE Conference on Decision and Control*, pages 1693–1698. [Cited on p. 88.]

Trélat, E. (2005). *Contrôle Optimal: Théorie & Applications*, volume 36. Vuibert Paris. [Cited on pp. 13, 16, 23, 24, 25, and 104.]

Trélat, E. (2012). Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications*, 154(3):713–758. [Cited on p. 104.]

Trélat, E. and Zuazua, E. (2015). The turnpike property in finite-dimensional nonlinear optimal control. *Journal of Differential Equations*, 258(1):81–114. [Cited on p. 23.]

Tsitsiklis, J. N. and Van Roy, B. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690. [Cited on pp. 32, 120, 121, 125, and 126.]

Vacher, A., Muzellec, B., Rudi, A., Bach, F., and Vialard, F.-X. (2021). A dimension-free computational upper-bound for smooth optimal transport estimation. In *Conference on Learning Theory*, pages 4143–4173. [Cited on p. 59.]

van Brunt, B. (2004). *The Calculus of Variations*. Springer New York. [Cited on pp. 1 and 7.]

Verschueren, R., Frison, G., Kouzoupis, D., Frey, J., van Duijkeren, N., Zanelli, A., Novoselnik, B., Albin, T., Quirynen, R., and Diehl, M. (2022). acados—a modular open-source framework for fast embedded optimal control. *Mathematical Programming Computation*, 14(1):147–183. [Cited on p. 24.]

Vinter, R. (1993). Convex duality and nonlinear optimal control. *SIAM Journal on Control and Optimization*, 31(2):518–538. [Cited on pp. 104, 105, and 114.]

Von Stryk, O. (1993). Numerical solution of optimal control problems by direct collocation. In *Optimal Control*, pages 129–143. Springer. [Cited on p. 24.]

Wahba, G. (1990). *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM. [Cited on pp. 55 and 132.]

Weidmann, J. (2012). *Linear Operators in Hilbert Spaces*, volume 68. Springer Science & Business Media. [Cited on p. 136.]

Wiener, N. (1948). *Cybernetics or Control and Communication in the Animal and the Machine*. MIT Press (2019 re-edition). [Cited on pp. 1 and 7.]

Williams, C. and Seeger, M. (2000). Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press. [Cited on p. 56.]

Woodworth, B., Bach, F., and Rudi, A. (2022). Non-convex optimization with certificates and fast rates through kernel sums of squares. *arXiv preprint arXiv:2204.04970*. [Cited on p. 118.]

Xu, T., Wang, Z., Zhou, Y., and Liang, Y. (2020). Reanalysis of variance reduced temporal difference learning. *arXiv preprint arXiv:2001.01898*. [Cited on p. 121.]

Yang, T., Li, Y.-F., Mahdavi, M., Jin, R., and Zhou, Z.-H. (2012). Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems*, volume 25. [Cited on p. 56.]

Yu, H. and Bertsekas, D. P. (2010). Error bounds for approximations from projected linear equations. *Mathematics of Operations Research*, 35(2):306–329. [Cited on p. 121.]

Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. *IEEE Transactions on Automatic Control*, 26(2):301–320. [Cited on p. 1.]

RÉSUMÉ

L'apprentissage par renforcement désigne pour un agent le fait d'apprendre à agir dans un environnement inconnu, de façon à maximiser sa récompense sur le long terme. Il trouve son origine dans le domaine du contrôle optimal, ainsi que dans certains travaux en psychologie. L'augmentation des capacités de calcul et l'utilisation de méthodes d'approximation comme les réseaux de neurones ont permis des succès récents notamment pour la résolution des jeux, sans pour autant systématiquement fournir des garanties théoriques. Quant au domaine du contrôle optimal, pour lequel un modèle de l'environnement est fourni, il a connu des développements théoriques solides dès les années 1960, avec des outils numériques qui ont fait leurs preuves dans de nombreuses applications industrielles. Néanmoins, la résolution numérique de problèmes de contrôle non-linéaires de grande dimension, problèmes qui sont notamment rencontrés en robotique, reste aujourd'hui relativement ouverte.
Dans cette thèse, nous développons et analysons des algorithmes efficaces, si possible avec des garanties théoriques, pour le contrôle et l'apprentissage par renforcement. Nous montrons que, même s'ils sont formulés différemment, ces deux problèmes sont très proches. Nous nous intéressons d'abord à la discrétisation des processus de décision Markoviens déterministes à état continu, en adaptant une méthode développée pour le contrôle en temps continu. Puis nous proposons une méthode d'estimation rapide de régions de stabilité applicable à des systèmes dynamiques de grande dimension imparfaitement connus. Nous généralisons ensuite un algorithme de résolution de problèmes de contrôle issu de l'optimisation polynomiale, aux systèmes non-polynomiaux et connus à partir d'un nombre fini d'observations. Pour cela, nous utilisons une représentation comme somme de carrés des fonctions positives lisses issue des méthodes à noyaux. Enfin, nous analysons un algorithme classique en apprentissage par renforcement, l'algorithme des différences temporelles, dans sa version non-paramétrique. Nous soulignons ainsi le lien entre l'algorithme des différences temporelles et l'algorithme de descente de gradient stochastique, pour lequel de nombreux résultats de convergence sont connus.

MOTS CLÉS

contrôle optimal ; apprentissage par renforcement ; méthodes numériques ; approximation max-plus ; fonctions de Lyapunov ; espaces à noyaux reproduisants ; sommes de carrés ; estimation non-paramétrique.

ABSTRACT

Reinforcement learning describes how an agent can learn to act in an unknown environment in order to maximize its reward in the long run. It has its origins in the field of optimal control, as well as in some works in psychology. The increase in computational power and the use of approximation methods such as neural networks have led to recent successes, in particular in the resolution of games, yet without systematically providing theoretical guarantees. As for the field of optimal control, for which a model of the environment is provided, it has known solid theoretical developments since the 1960s, with numerical tools that have proven useful in many industrial applications. Nevertheless, the numerical resolution of high dimensional nonlinear control problems, which are typically encountered in robotics, remains relatively open today.
In this thesis, we develop and analyze efficient algorithms, when possible with theoretical guarantees, for control and reinforcement learning. We show that, even though they are formulated differently, these two problems are very similar. We first focus on the discretization of continuous state deterministic Markov decision processes, by adapting a method developed for continuous time control. Then we propose a method for fast estimation of stability regions applicable to imperfectly known high-dimensional dynamical systems. We then generalize an algorithm for solving control problems derived from polynomial optimization, to non-polynomial systems known through a finite number of observations. For this, we use a sum-of-squares representation of smooth positive functions from kernel methods. Finally, we analyze a classical algorithm in reinforcement learning, the temporal-difference learning algorithm, in its non-parametric version. In particular, we insist on the link between the temporal-difference learning algorithm and the stochastic gradient descent algorithm, for which many convergence results are known.

KEYWORDS

optimal control; reinforcement learning; numerical methods; max-plus approximation; Lyapunov functions; reproducing kernel Hilbert spaces; sums-of-squares; non-parametric estimation.