Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Reinforcement Learning Through the Lens of Koopman Operators

The Infinite-Dimensional Framework

Ph.D. candidate
Francesco Zanini

Advisor
prof. Alessandro Chiuso

Director & Coordinator
prof. Andrea Neviani

Ph.D. School in
Information Engineering

Department of
Information Engineering
University of Padova
2022

To Nicolas Bourbaki, the mathematician who mastered nearly all aspects of the field, emphasising rigor in place of conjecture

Actually, to my brothers

Sommario

Il Reinforcement Learning è ad oggi una delle più attive aree di ricerca dell'Intelligenza Artificiale. Ciò è dovuto sia ai suoi successi applicativi, sia alle diverse sfide teoriche che presenta, trovandosi all'intersezione fra Teoria del Controllo e Machine Learning. Sebbene il paradigma sia piuttosto semplice, il quale consiste in un agente che interagisce con l'ambiente per ottenere reward, ha aperto la strada ad approcci molto diversi. Questi sono distinti in due grandi categorie: model-based, che si basa su previsioni date da un modello dell'ambiente, e model-free, che affronta direttamente il problema della ricerca della policy; entrambi con pregi e difetti. La ricerca attuale sta cercando di unire queste filosofie, ottenendo un algoritmo che eredita i vantaggi di entrambe.
Parallelamente, e finora in modo essenzialmente indipendente, negli ultimi anni si è assistito a un progressivo aumento della letteratura sul Koopman operator. Quest'ultimo consente una descrizione lineare di un qualsiasi sistema dinamico non lineare, basandosi però su uno spazio di funzioni, che risulta essere infinitamente dimensionale. I vantaggi della descrizione lineare hanno portato a una grande diffusione di questa strategia: prima per l'analisi dei flussi non lineari, e recentemente anche per il problema del controllo. Chiaramente i vantaggi della caratterizzazione lineare sono bilanciati dallo svantaggio di dover gestire una descrizione infinito-dimensionale del sistema. Quest'ultima non è in pratica esprimibile, quindi in letteratura sono emerse diverse tecniche di approssimazione finito-dimensionali, anche per stimare l'operatore di Koopman da osservazioni della dinamica. Tuttavia, queste si basano sulla definizione a priori di uno spazio a dimensione finita, in cui l'operatore viene approssimato, cosa che rende estremamente vincolante l'utilizzo di queste tecniche senza informazione a priori sul sistema.
In questa Tesi viene presentato un nuovo approccio per la stima del Koopman operator, basato sulla teoria dei Reproducing Kernel Hilbert Spaces. Quest'ultimi sono effettivamente spazi di Hilbert, quindi infinito-dimensionali, che però inducono una soluzione finito-dimensionale del problema di ricostruzione, grazie al Representer theorem. La relazione tra il Koopman operator e i Reproducing Kernel Hilbert Spaces è ampiamente discussa dal punto di vista della teoria degli operatori. In particolare, l'adozione degli RKHS permette di rilassare l'ipotesi di uno spazio finito-dimensionale fissato a priori per la stima, in quanto l'unica informazione a priori nel problema di stima è rappresentata dalla scelta del kernel, che caratterizzerà poi le funzioni di base di cui la ricostruzione è una combinazione lineare.
Questo particolare approccio, dato dall'unione del Koopman operator e degli RKHS, risulta essere particolarmente conveniente per affrontare il problema proposto dal Reinforcement Learning. Infatti si dimostra in questo lavoro come una formulazione naturale

della value-function, che è la quantità chiave nel framework del Reinforcement Learning, possa essere data tramite iterazioni successive del Koopman operator. Essendo il reward l'unica cosa che conta, la propagazione di questa particolare funzione attraverso l'operatore di Koopman fornisce un modo efficace di affrontare il problema del Reinforcement Learning. Quest'ultimo configura infatti un modello che non si basa sulla propagazione dello stato, ma solamente sul reward: può quindi essere considerato un approccio a metà strada tra una prospettiva model-free e una model-based.

Abstract

Reinforcement Learning is nowadays one of the most active research areas in Artificial Intelligence. This is due both to its practical successes, and to the theoretical challenges it poses, lying in between Control Theory and Machine Learning. Although the paradigm is very simple, and it consists of an agent which interact with an environment to collect reward, it paved the way for very diverse approaches. These are distinguished in two main categories: model-based, which relies on predictions given by a model of the environment, and model-free, which tackles directly the policy search problem; both with merits and defects. Current research is trying to merge these perspective, yielding an algorithm which inherits benefits of both.

In parallel, and until now practically independently, recent years have seen a progressive increase of literature on the Koopman operator framework. The latter allows for a linear description of any nonlinear dynamical system, although relying on a space of function, which however is infinite-dimensional. The advantages of the linear description have led to a great diffusion of this perspective: first for the analysis of nonlinear flows, and very recently also for the control perspective. Clearly the benefits of the linear characterisation are balanced by the drawback of having to handle an infinite-dimensional description of the system. The latter is not feasible in practice, therefore different finite-dimensional approximation techniques emerged form the literature, also to learn the Koopman operator from data. However these are based on the a priori definition of a finite-dimensional space, in which the operator will be learnt, so that they are unsuccessful without significant prior knowledge.

In this Dissertation a new approach to estimate the Koopman operator is presented, relying on Reproducing Kernel Hilbert Spaces theory. The latter are indeed Hilbert spaces, therefore infinite-dimensional, however they yield a finite-dimensional solution of the reconstruction problem, thanks to the Representer theorem. The connections between Koopman operator framework and Reproducing Kernel Hilbert Spaces are widely discussed, from the perspective of operator theory. In particular, by framing the learning problem in RKHS, it is possible to relax the assumption of a fixed finite-dimensional space for the estimation, and the only prior knowledge embedded in the problem is done through the kernel, which shapes the functions yielding the estimate.

The latter approach, resulting from the combination of the Koopman operator framework and RKHS, turns out to be particularly suitable for dealing with the Reinforcement Learning problem. In this work it is shown how a natural formulation of the value-function, which is the key quantity of the Reinforcement Learning setting, can be given by subsequent iterations of the Koopman operator. As the reward is all that matters, the

propagation of this particular function through the Koopman operator gives a concise way of addressing the Reinforcement Learning problem. The latter indeed provides a model which is not based on the propagation of the state, but relies only on the reward: therefore it can be regarded as an approach lying midway between a model-free and a model-based perspective.

Contents

Notation

X	State space
A	Input space
Y	Output space
x	State
t	Continuous time
f	Vector field
f_{o}	Output map
a	Input
y	Output
$\mathfrak{g}$	Gravitational constant (Earth)
$\mathfrak{l}$	Rod length
c	Coefficient
$\mathcal{U}^t$	Koopman operator (continuous)
γ, η	Parameters
ψ	Observable
d	Dimension of state space
Φ	Flow
Ψ	Observable space
$\mathfrak{F}$	Field
$\mathcal{M}$	Metric space
L^p	p-Lebesgue space of functions
$\mathcal{C}^0$	Space of continuous functions
$\mathcal{L}$	Infinitesimal Koopman Generator
$\mathcal{G}$	Vector space
$\mathcal{O}$	Operator
g	Element of a vector space
λ	Element of the spectrum
φ	Eigenfunction
v	Koopman mode
S	Functional
H	Hilbert space
h	Element of the Hilbert
s	Representer
$\mathfrak{S}$	σ-algebra
ρ, ν, μ	Measures

$\mathcal{P}$	Perron-Frobenius operator
$\mathbb{1}$	Characteristic function
Ω	Sample space
W	Wiener process
k	Discrete time
N	Dimension of the Koopman approximation
$\mathfrak{P}$	Projection
D	Dictionary of observables
Ψ_D	Space of function spanned by the dictionary
U	Finite-dimensional Koopman operator
M	Number of datapoints
$\overline{x}, \overline{y}$	Datapoints
l^p	p-Lebesgue space of sequences
K	Kernel
ϕ	Element of a basis
$\mathcal{S}$	Kernel Integral Operator
α, β	Coefficients
c	Cost
$\mathcal{Z}$	left-shift operator
p	Dimension of input space
A, B	dynamics matrices
Q, R, Q_f	Cost matrices
π	Policy
Π	Space of policies
V_π	Value-function
θ	Policy parameter
Θ	Policy space
l	Dimension of the policy parameter
r	Reward function
z	State-parameter variable
$X\Theta$	State-parameter space
w	State noise
ε	Observable noise
$\mathfrak{c}$	Constant
p	Dimension of input space

1 Introduction

This Dissertation originates from the union of different fields, which combined together form a single, coherent picture. In particular, a new approach to deal with the Reinforcement Learning problem is proposed, which derives from the integration of the theory of Reproducing Kernel Hilbert Spaces into the Koopman Operator framework. The resulting algorithm yields a simple method to estimate the value-function and its gradient, so that the control parameter can be easily improved. The latter method is built step by step, thus starting with a thorough review of the foundations, proceeding to address the simpler problem of estimation of a dynamical system, and finally dealing with the control problem. Throughout this discourse, other original contributions are presented. The problem of choosing the sampling time to collect data from a dynamical system is also addressed, as well as the propagation of the uncertainty in the estimate of the dynamics.

Reinforcement Learning is nowadays one of the most active research areas in artificial intelligence. Its rapid diffusion in recent years can probably be explained by the fact that it combines two pillars of scientific research: *control theory* and *machine learning*. From the former, it inherits the paradigm, the final purpose, and largely shares the setting. The objective of the Reinforcement Learning algorithms is indeed to devise a *policy*, which is basically the same concept as the controller, in order to obtain the best performance from the resulting behaviour of the system with which they interact. From the latter, it draws the methodologies and the techniques, as well as the language. Indeed the search for policy relies on finite samples coming from observations of the system, with which it is possible to derive estimates of the quantities involved, and to build complexity bounds.
Growing on the shoulders of these giants, the Reinforcement Learning framework have been able to achieve unimaginable goals and to overcome hitherto inaccessible limits.

One of the most celebrated examples is given by the *AlphaGo* algorithm by Silver et al., 2016, which defeated the European champion Fan-Hui $5-0$ in a regular tournament match, the first time a computer program has defeated a human professional player in even games. Reinforcement Learning techniques are nowadays widely applied in other fields as well, like robotics (Kober, Bagnell, and Peters, 2013), and very recently also Natural Language Processing (Luketina et al., 2019). Also branches of science with a less significant computational part, like Psychology and Neuroscience, benefited from connections with Reinforcement Learning: many core algorithms developed to address this setting were also discovered to operate inside the brain, and vice-versa the study of the cerebral patterns have been of inspiration for actual algorithms (Sutton and Barto, 2018).

The success of Reinforcement Learning is also due to the fact that all algorithms descend from a simple but specific picture, which defines the setting of the Reinforcement Learning problem. The general framework involves an agent which iteratively interacts with an environment, by selecting actions. As a consequence of the chosen action, the agent moves into a new state, and receives a reward signal. The goal is to obtain as much reward as possible, despite the uncertainty about the environment. Nonetheless the future state of the agent is affected by its current action, consequently characterising opportunities and potential in later stages. It is clear that in order to establish the optimal choice it is necessary the ability to foresee the delayed effects of the interplay between the environment and the chosen policy. This changes the focus of the agent from the reward to the value-function. Indeed while the agent and the environment are the actors in the scene, the actual mathematical elements which are indispensable to the Reinforcement Learning theory are: the *policy* π, which defines the agent behaviour with respect to environment, and in general is a mapping from the set of observations available to the agent to actions, so that actions of the agent are specified for each state; the *reward* r, which is a scalar-valued signal that defines the objective of the Reinforcement Learning agent, and it is provided at each time step by the environment, thus specifying which are the good and bad events for the agent; the *value-function* V_π, which plays the same role of the reward but considering the long run, so that it gives a cumulative description of the performance of the agent over more steps, when following a certain policy; and the *model of the environment*, which is a mathematical description of the environment, allowing to predict the future responses of the environment to specific state and actions. Note that even if the reward function and the value-function represent similar concept, there is a significant difference. The reward is indeed an intrinsic element of the Reinforcement Learning setting, and is therefore directly provided by the environment;

the value-function is instead built by the agent for the sole purpose of collecting more reward. The objective of the Reinforcement Learning framework is defined by the reward, however it is the maximisation of the value-function which allows to achieve the highest performances, as the problem of interactive decision making develops over several steps of interaction, so that predictions about the future becomes a key factor. Sutton and Barto, 2018 explicitly say that "the most important component of almost all Reinforcement Learning algorithms [...] is a method for efficiently estimating values. The central role of value estimation is arguably the most important thing that has been learned about Reinforcement Learning over the last six decades". The value-function will indeed serve as the final performance measure to maximise. Therefore the goal of the generic Reinforcement Learning algorithm would be to find the policy π^* for which the value-function yields the maximum value. The aim of this work will consequently be to develop a new methodology for estimating the latter function under a certain policy, what in *Dynamic Programming* literature is called *policy evaluation*; and at the same time to solve the *policy improvement* task, i.e., to provide a simple way to obtain a better policy than the current one, with respect to the partial ordering given by the value-function.

Reinforcement Learning is often divided into two major categories: model-based Reinforcement Learning, and model-free Reinforcement learning. The former prescribe to learn a model of the environment, and then use the resulting predictive capabilities to design a control policy which has (possibly) optimal performances, according to the corresponding estimation of the value-function. The latter instead seeks a direct improvement of the policy through available data, without building any model of the transition dynamics. These two perspectives have been analysed in the literature and some systematic merits and defects have been identified. In particular, model-based algorithms are often said to be more sample-efficient but yield a worse asymptotic behaviour; while model-free ones requires more samples to converge to a solution, but the latter attains usually higher performances (Tu and Recht, 2019a).
While the model-based perspective is surely closer to the classical control framework, it is worth pointing out that also the control community started to develop methods which did not require the identification of a model of the plant, called *direct data-driven control*. The growing literature on this topic indicates that the model-free approach may be a viable alternative to the well-established framework which envisage the estimation of the plant first, and the design of the controller based on the latter estimate. These works nonetheless maintain some differences with the Reinforcement Learning setting which are peculiar of the control language, like the description in frequency domain of most of

the involved quantities (Selvi et al., 2021; Cerone, Regruto, and Abuabiah, 2017; Piga, Formentin, and Bemporad, 2018; Formentin, 2012; Breschi, Sassella, and Formentin, 2022).

Recently, there have been a collective effort from the machine learning community to develop algorithms which combines the strengths of both approaches while mitigating their weaknesses (Racanière et al., 2017; Gu et al., 2016; Feinberg et al., 2018).

One of the most celebrated algorithms in history of Reinforcement Learning is *MuZero* by Schrittwieser et al., 2020, an extension of the aforementioned AlphaGo, which however does not make use of samples from an expert and can handle imperfect dynamics. The latter is accomplished indeed by building a model of the environment, but in a peculiar way, so that this model depends on the rewards of the transitions, and not on the transitions themselves. In particular, the state of the environment x is first mapped into a lower-dimensional representation $[x]$, from which the lower-dimensional representation of the next state $[x']$ is predicted, together with the corresponding reward r. All these steps are managed by an *Artificial Neural Network*, whose weights must be updated with respect to the target. The peculiarity of this approach is that these weights are adjusted only according to the reward signal, and not on the observation of the next state. In this way the learnt lower-dimensional feature space is free to represent arbitrary characteristics of the state, and is not bound by having to explain actual transitions. The model is then driven by the reward alone, which makes it goal-oriented, and that may explain its success with respect to competitors. To back up this intuition, consider the trivial example of a constant reward, so that no learning is actually necessary, since every policy attain the exact same reward at each step. The environment can be arbitrary complicated, and this would obviously have no impact on the performances. A model-based algorithm which explicitly takes into account transitions would try to model the complicated dynamics, even though it is completely useless. An algorithm which focuses on the reward would instead immediately realise that predictions are actually trivial - although nothing changes in this example, since it is not possible to improve performances. Of course also a model-free algorithm would quickly realise that the reward is always the same, as they are not based on predictions, therefore no model of the environment is involved. However, as mentioned before, model-free algorithms are known to suffer from sample inefficiency, precisely because the model turns out to be a convenient and compact description of the interactions with the environment. This leads to the claim that MuZero is an algorithm somewhere in between the two perspectives.

In this Dissertation, an algorithm with a similar principle is proposed, which indeed considers only the propagation of the reward function, and does not learn from the state

transitions. In particular, the latter approach deals with continuous action and space states, which is a setting closer to the control community than to the Reinforcement Learning one. Although there are works in Reinforcement Learning with continuous action and state spaces, it is definitely not the most popular setting, not least because of the more difficult challenges it poses. For instance, if the working space is continuous, it is impossible to resort even to asymptotic result which are often evoked for the finite-states setting, which helps to prove convergence of algorithms relying on the fact that every state-action pair is visited infinitely many times under the assumption of an infinite data budget. In order to achieve this, the method presented here exploits the Koopman operator framework, which has been neglected until recently, but which has lately been making a rather big impact in control theory, as evidenced by the many new papers on the subject (Mauroy, Mezić, and Yoshihiko, 2020). The latter will allow the estimation of the system to focus only on reward, and not to chase transitions.
The Koopman operator framework is an alternative description of a general nonlinear dynamical system, which is translated in an *infinite-dimensional* but *linear* fashion. The latter dates back to the seminal work by Koopman, 1931, but it is only in more recent years that it has established itself as a useful tool for the analysis of dynamical systems. First rediscovered and widely used by the fluid dynamics community (Arbabi and Mezić, 2017b), its popularity has since been consolidated thanks to the simple estimation methods derived from it (Tu et al., 2014; Williams, Kevrekidis, and Rowley, 2015), and the possibility of exploiting theorems on spectrum analysis (Mauroy, 2021; Mezić, 2020). Instead of characterising a dynamical system through the derivative over time of the state variable, as it usually done in system theory, the Koopman operator framework considers an infinite-dimensional space of functions, and it gives a description of the evolution of any function $\psi(\cdot)$ in this space, when propagated through the dynamics under study. Therefore it characterises the composition of the function given as input and the state-transition map $f(\cdot)$, which is equivalent to the function given as output $\psi_+(\cdot)$. The latter is expressed as:

$$\mathcal{U}^t[\psi](t_0) = \psi(x(t_0 + t)) = \psi_+(\cdot), \tag{1.1}$$

where it is evident that system is expressed through the composition of a generic function ψ, called *observable*, and the dynamics $f(\cdot)$.

The latter description can be given in both continuous and discrete time, depending on the underlying system. Clearly, equivalently to what happen with the description in state space, the continuous-time description is more general, because it can provide

the discrete-time description by constraining the time variable to assume predetermined values and considering the integral between these instants. However, in this work, the operator is learned from a finite amount of data coming from transitions of the system, as prescribed by the Reinforcement Learning setting. These data necessarily provides discrete-time information about the system, because they are physically collected at a certain sampling rate. Given that the Koopman operator returns a linear characterisation of a generic nonlinear system, it is not difficult to turn the above system into its continuous version, however the most widespread approaches yield a reconstruction in discrete-time. This is why an entire chapter of this Dissertation is dedicated to the study of how the sampling time influences the properties of the system.
A common belief is that a finer time discretisation always leads to better estimation of the system behaviour, also for the control cost. However, this will only prove to be true with an unlimited data budget. With finite data, a higher temporal resolution means that more data is collected within fewer episodes. This inevitably leads to the question on how to optimally choose the time discretisation for the task at hand. In practice, there are always limitations on how much data can be collected, stored and processed. The practitioner hence faces a fundamental trade-off: a high temporal resolution leads to a better approximation of the continuous-time system from discrete measurements, whereas collecting data along a larger number of trajectories leads to lower variance in the estimation with respect to stochasticity in the system. This is indeed true for any system with stochastic dynamics, even if the learner has access to exact (noiseless) measurements of the system's state. Data efficiency can be improved by leveraging a precise understanding of the trade-off between approximation error and statistical estimation error in long term value estimation — two factors that react differently to the level of temporal discretisation.
In particular, the behaviour of a simple Monte Carlo estimator with a fixed data budget has been investigated. As explained above, the most important quantity for Reinforcement Learning is the value-function, therefore the problem of value-function estimation has been considered. By taking the simplest possible dynamics, i.e. a Langevin system, the analysis showed exactly the existence of a trade-off in the choice of the sampling interval. By taking it too small, the single trajectory is well described by the zero-order hold approximation, so that the cost (or the reward) over the long run is accurate. However this comes at the price of reducing the number of trajectories that are evaluated, since the data budget is fixed, which are different from each other because the dynamics are stochastic. Instead by taking many trajectories, the noise due to the stochasticity can be managed, but the cost of the single trajectory would be severely biased, as only few

samples are used for the approximation. The presence of a trade-off implies the existence of an optimal value for the sampling interval, optimally balancing these two phenomena, which is discussed in Chapter 3.
As a consequences of the above results, data can always be thought of as sampled at the optimal sampling time, so that it is impossible to extract more information from the system with a finite data budget, and consequently the discrete-time approximation yield the best possible description. In light of this, the Koopman operator estimation is exclusively discussed in the discrete-time setting.
In order to develop the methodology to tackle the Reinforcement Learning problem, the Koopman operator must indeed be estimated from data, since the latter allows to build a model of the environment, which it is necessary for predictions. In order to do that, the setting of Reproducing Kernel Hilbert Spaces have been considered.

Hilbert spaces satisfying certain additional properties are known as Reproducing Kernel Hilbert Spaces, and RKHS theory is normally described as a transform theory between Reproducing Kernel Hilbert Spaces and positive semi-definite functions $K(\cdot, \cdot)$, called kernels: every RKHS has a unique kernel, and certain problems posed in RKHSs are more easily solved by involving the kernel (Wahba, 2003). Reproducing Kernel Hilbert Spaces are indeed very convenient spaces in which to frame the estimation problem because they are infinite-dimensional, but under specific conditions, the solution of a regularised regression problem lies in a particular finite-dimensional space. They are particularly important in the field of statistical learning theory because of the celebrated representer theorem, which states that every function in an RKHS that minimises an empirical risk functional can be written as a linear combination of the kernel function evaluated at the training points. This is a practically useful result as it effectively simplifies the estimation problem from an infinite-dimensional to a finite-dimensional optimisation problem.
The latter reduction can be exploited to learn a finite-dimensional approximation of the Koopman operator. The latter is learnt by minimising the prediction error for the available datapoints, i.e., in a regression framework. However, in order to make the problem meaningful, the finite-dimensional reduction of the hypothesis space have always been considered to be given or to be predetermined, which limits the ability of estimation or requires strong a priori knowledge. By framing the problem in RKHS, the prior knowledge which is required to be embedded in the problem is just the one given by the kernel, which is a very flexible paradigm and can be adapted to different problems, yet it is still possible to obtain a finite-dimensional solution. In Chapter 6 the estimation of Koopman operators in RKHS is discussed, which does not solve the control problem

yet. However it can already be seen how the estimation of the system relies on the observation of the selected observable. The system is indeed represented through the Koopman operator in RKHS, given by $U_{\text{DMD}}^{(K)}$, and is able to propagate any observable through its values on the training points, as:

$$\widehat{\psi\left(f\left(\cdot\right)\right)} = K\left(\cdot, \overline{x}\right)\left[K\left(\cdot, \overline{x}\right) + \sigma^2\right]^{-1} U_{\text{DMD}}^{(K)} \psi\left(\overline{x}\right), \tag{1.2}$$

so that the propagation can be specified for a particular observable. The last chapters of the Dissertation leverage the intuition that the reward can be though as an observable, in order to build an approach indeed driven by the reward function. The above equation can indeed by rewritten as:

$$\widehat{r\left(f\left(\cdot\right)\right)} = K\left(\cdot, \overline{x}\right)\left[K\left(\cdot, \overline{x}\right) + \sigma^2\right]^{-1} U_{\text{DMD}}^{(K)} r\left(\overline{x}\right), \tag{1.3}$$

which yield the estimate of the evolution of the reward based on its observations on the experienced transition. Note how this approach is indeed aligned with the initial objective of building a model-based algorithm which however relies only the reward samples, and not targeting transitions. By recalling that the value-function is just the sum of the rewards over the predicted trajectory, and having an actual way to predict those rewards, it is easy to understand that this solves the task of policy evaluation, for a fixed policy generating the system under analysis. The choice of a policy induces indeed an autonomous system, as the control input will depend only on the state. For a one-step prediction, the output are not required to lie in the same space as the state variable, as for standard kernel methods. However, since the same regression will be iterated to yield predictions for arbitrary future steps, the latter becomes a mandatory assumption. The regression must be consistent with itself at the next step, so the state-transition function must be an endomorphism.

The latter framework is extended to the control problem through the assumption of a parameterised policy class, which is very common in the Reinforcement Learning setting with continuous state and action space. This assumption will still make it possible to treat the system as autonomous, by making it dependent on the control parameter, which actually changes the dynamics. If a parameterised system needs to be learnt, a parameterised Koopman operator needs to be defined. Under the dynamics defined by the control parameter θ then, the Koopman operator will maintain its propagation capabilities as:

$$\widehat{r\left(f\left(\cdot\right)\right)} = K\left(\cdot, \overline{x}\right)\left[K\left(\cdot, \overline{x}\right) + \sigma^2\right]^{-1} U_{\text{DMD}}^{(\theta)} r\left(\overline{x}\right), \tag{1.4}$$

In particular, by relying on a result concerning the solution of differential equation, in Chapter 8 it is explained how in the description of a parameterised system, the state and the parameters basically play the same role. This allows to consider an augmented state for the estimation problem in z, composed by states and parameters together, so that a parameterised Koopman operator can be learnt. The latter is given by:

$$V_\theta(\cdot) = \sum_{\tau=1}^{\overline{\tau}} K(\cdot, \overline{z})\, U_{\mathrm{EDMD}}^{(\theta)} \beta_\tau, \tag{1.5}$$

where β_τ are the coefficients for the subsequent propagations of the rewards. Assuming the kernel function is differentiable with respect to the control parameter, an expression for the gradient of the value-function is trivially derived. The control problem can then be solved through a gradient ascent algorithm following the above formula, which will yield a parameter with a better performance, according to the current estimation of the value-function. With every new observation the approximation of the value-function can be improved, and in turn the gradient.
The fact that the procedure is framed in the language of Reproducing Kernel Hilbert Spaces allows to understand the proposed algorithm through a Bayesian perspective, and in particular to consider confidence intervals (Wahba, 1983). Every successive observable after the first step is indeed estimated as a Gaussian Process, which has a well-defined covariance description. The latter perspective makes it possible to equip the estimation of the value-function through the proposed procedure with confidence bounds, expressing the region where the true function lies with high probability. This is a peculiar feature among Reinforcement Learning algorithms, which makes it clear that the proposed approach is unique and still has room for improvement, as there may be other uncommon hidden properties still to be explored.
Since experience without theory is blind, but theory without experience is mere intellectual play (Bertalanffy, 1962), the formal derivations of the aforementioned results are always complemented by illustrative examples.

The present work is structured as follows:

- A detailed introduction to the Koopman operator framework is presented in Chapter 2, which explains all its relevant properties and provides all the tools needed to understand the rest of the work. This part can be considered as background material, and indeed consist of knowledge coming from relevant works in the field.

- The innovative results on the analysis of the sampling time for observing continuous-

time system are given in Chapter 3. This chapter consists entirely of original contributions, which in particular are the exact characterisation of the mean-squared error for a Monte Carlo estimator of the value function in the case of a Langevin dynamics; and the corresponding derivation of closed-form solution for the optimal sampling time.

- Chapter 4 describes the current approaches to estimate the Koopman operator from data. The formalisation of the finite-dimensional restriction for the original Koopman operator is introduced, along with the preexisting techniques on which the development of the proposed method is based. These notions comes from well-established literature on the Koopman operator.

- The Reproducing Kernel Hilbert Space theory is illustrated in Chapter 5. The arguments are presented from an operator theory perspective, in order to facilitate connections with the Koopman operator framework. The contents of this chapter are again background material, without which it would not be possible to understand how the proposed method works.

- The formal connection between Koopman operators and Reproducing Kernel Hilbert Spaces is given in Chapter 6, leading to the core of the proposed approach, which so far will deal with estimation only. A detailed bridge between kernel methods and Koopman estimation comes to life in the form of a new algorithm which extends and generalises the previous ones. The latter allows also to understand better the dual perspective taken by the well-known methods. The whole chapter represents an original contribution.

- The control problem is introduced in Chapter 7, with particular emphasis on the algorithms exploiting the Koopman operator framework. A review on the current control approaches based on the Koopman linearisation is provided, both from a theoretical and a practical standpoint. The most relevant and structured works have been selected from the ever-growing literature.

- In Chapter 8 the original method previously developed for estimation is finally extended to the control setting. This extension require imperceptible changes from the previous version, as the control parameter is incorporated into the system description and basically treated as a state. This section represents an original contribution, which in particular is given by the derivation of a Policy Gradient algorithm in order to tackle the Reinforcement Learning task. The latter exploits

the Koopman operator framework, which allows the final estimate to be based on the reward only.

- The last chapter presents additional features of the proposed approach. In particular, the difficult problem of propagating uncertainty of the estimated dynamical system is considered, which can be easily handled by the proposed method. The observable are indeed propagated as Gaussian Process, which have a well-known covariance description. A refinement of the proposed method is also presented, which extend the procedure to the case in which the knowledge of the reward function is not available, but only its samples from the experienced transitions, as prescribed by the Reinforcement Learning setting. Also Chapter 9 represents an original contribution in its entirety.

- Conclusions are drawn in Chapter 10. Strengths and weaknesses of the proposed approach are also discussed, along with open problems and possible extensions.

2

Koopman operator

The Koopman operator is the fundamental mathematical tool on which all the work presented in this Dissertation is based.
In the following a motivation for its introduction, the formal definition and its most important properties are provided. Some preliminary notions on infinite-dimensional operators are also introduced. Particular attention is paid to the eigendecomposition, as well as the Koopman description of stochastic systems.
The contents are mainly based on the work by Mauroy, Mezić, and Yoshihiko, 2020.

2.1 A linear framework

Classical system theory is built around the concept of *state*, which clearly has a central role in characterising a system (Strogatz, 2000; Wiggins, 2003). Its purpose is indeed to describe the quantities of interest for the system dynamics, which is in fact done through the definition of the state variable. The state is the subject of the autonomous variations of the system, while also being affected by inputs, and ultimately producing outputs. Consequently, a dynamical system is usually described through a function, which characterises the evolution of the state variable with respect to time. The latter variable is set to lie in a predefined space, the *state space*.
In the *continuous-time* setting, this corresponds to specify the derivative of a generic state in the aforementioned space for any time instant, thus naturally leading to a representation through differential equations. This led to the usual view of dynamical systems in *state-space* form.
The state space is denoted as X, the input space as A and the output space as Y. The

notation used in this works reads as:

$$\begin{cases} \dot{x}(t) = \frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), a(t), t) \\ y(t) = f_{\mathrm{o}}(x(t)) \end{cases} \tag{2.1}$$

where $t \in \mathbb{R}$ represents time, $f : X \times A \times \mathbb{R} \to X$ is the vector field and $f_{\mathrm{o}} : X \to Y$ is the output map.

For the whole manuscript, the state will be assumed measurable, i.e., the output map is the identity function: the latter is a rather standard assumption in many works which deal with the Koopman operator framework, as well as in the Reinforcement Learning literature. Only *time-invariant* systems will be considered, for which the vector field does not have an explicit dependence on time. Moreover, until Chapter 7, autonomous system are taken into account, which does not allow for an input to change the dynamics. The latter can be thought in the general scenario as if the input was constantly zero, i.e., $a(t) = 0$, $\forall t \in \mathbb{R}$.

One of the simplest and most studied case of a dynamical system is the swinging of a clock pendulum (Baker and Blackburn, 2008), whose dynamics are given by:

$$\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} = -\frac{\mathfrak{g}}{\mathfrak{l}} \sin(x(t)) \tag{2.2}$$

in which the state $x \in [-\pi, \pi]$ represents the angle between the direction of the rod and the vertical axis, i.e., the one along which gravity acts. The dynamics depend on the local gravitational field of Earth $\mathfrak{g}$, and the length of the rod $\mathfrak{l}$, and describe the evolution over time t. There are no exogenous inputs - hence it is an *autonomous* system - and the state can be fully observed at any time, so it coincides with the output.

Already in this simple example it turns out that there is no explicit solution for the dynamics, although the space has been parameterised such that the dynamics can be written in their simplest form, which is the one given by (2.2), conveniently expressing the angle as a state. This means that there is no closed-form solution for the differential equation in (2.2), and so there is not an explicit expression for the *flow* of the dynamics, i.e., the evolution of the state from a given state and time.

The problem however can be simplified by assuming that x is close to zero. By recalling that $\sin(x) = x + O(x^3)$, equation (2.2) is approximated with the dynamics for the harmonic oscillator, given by:

$$\frac{\mathrm{d}^2 x(t)}{\mathrm{d}t^2} = -\frac{\mathfrak{g}}{\mathfrak{l}} x(t) \tag{2.3}$$

under the "small oscillations" hypothesis.
As a result, the latter has a closed-form solution which can be expressed by:

$$x(t) = c_1 \sin\left(t\sqrt{\frac{\mathfrak{g}}{\mathfrak{l}}}\right) + c_2 \cos\left(t\sqrt{\frac{\mathfrak{g}}{\mathfrak{l}}}\right) \tag{2.4}$$

where the coefficients c_1 and c_2 depend on the initial conditions. Equation (2.4) completely characterise the dynamics of the state x over time, given initial conditions on angular position and angular velocity, $x(t_0) = x_0$ and $\dot{x}(t_0) = \dot{x}_0$.
Note that the nonlinearity given by the sine in the differential equation makes the problem significantly harder, preventing an explicit solution from being found. Linear differential equations instead can be solved exactly, making the linear setting very appealing. The Koopman operator provides a way to *globally linearise* nonlinear systems.
The Koopman operator (or *composition* operator) framework gives an alternative description of any dynamical system, in a function space. The evolution of the system is not described through the dynamics of the state variable, but rather through functions taking the state as argument, which change over time. This means that, for every function given as an input of the Koopman operator $\mathcal{U}^t$, a new function is returned, which is equal to the former function whose argument is the evolution of the state after some time t, for any state. Since the Koopman operator describes the evolution of any function for any time, it provides a complete characterisation of the dynamical system.
The main advantage of this framework is that the description of the dynamics is always linear. Therefore even the nonlinear dynamics given in (2.2) can be characterised by a linear operator. The latter is given by:

$$x(t_0 + t) = \mathcal{U}^t\left[x(t_0)\right], \tag{2.5}$$

in the case that the propagated function is the identity.
The formulation above may appear inconclusive, as it would need infinitely many functions to provide an exact linear description of the dynamics, the operator being infinite-dimensional. The following example, proposed in Brunton and Kutz, 2019, illustrates a very simple case in which the dynamics can be characterised with a finite number of functions, so that the global linearisation provided by the Koopman operator can be better understood.

Example 2.1.1 (Finite-dimensional Koopman operator). Consider the nonlinear dynamical

system given by:

$$\begin{cases} \dot{x}_1(t) = \eta x_1(t) \\ \dot{x}_2(t) = \gamma\left(x_2(t) - x_1^2(t)\right) \end{cases} \tag{2.6}$$

For $\gamma < \eta < 0$ the system exhibits a slow attracting manifold given by $x_2(t) = x_1^2(t)$. By describing instead the system through the following functions:

$$\psi_1(x) = x_1; \quad \psi_2(x) = x_2; \quad \psi_3(x) = x_1^2; \tag{2.7}$$

the dynamics become linear. Indeed:

$$\begin{cases} \dot{\psi}_1(x) = \eta x_1(t) = \eta \psi_1(x) \\ \dot{\psi}_2(x) = \gamma\left(x_2(t) - x_1^2(t)\right) = \gamma\left(\psi_2(x) - \psi_3(x)\right) \\ \dot{\psi}_3(x) = 2x_1(t)\,\dot{x}_1(t) = 2\eta\psi_3(x) \end{cases} \tag{2.8}$$

which yield the following simple linear system,

$$\begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{bmatrix} = \begin{bmatrix} \eta & 0 & 0 \\ 0 & \gamma & -\gamma \\ 0 & 0 & 2\eta \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix}. \tag{2.9}$$

In the augmented description, $\psi_3 = \psi_1^2$ is an invariant manifold, i.e., trajectories which starts there are constrained to always stay on the manifold. This can also be seen as a constraint to recover the original system in (2.6), since the lifted dynamics in (2.8) are capable of describing a more general behaviour, that is not captured by the original bi-dimensional characterisation. The dynamics in (2.9) in fact makes sense even if they are initialised with $\psi_3 \neq \psi_1^2$. Trajectories matching the behaviour of the original system are therefore the ones constrained on the invariant manifold $\psi_3 = \psi_1^2$. This issue is explored in more detail by Iacob et al., 2021.

The introduction of this framework dates back to the seminal work by Koopman, 1931 and from early on aroused considerable interest in statistical mechanics (Gaspard, 1998). It is only in recent times, however, that the Koopman operator proved itself an effective tool for studying geometric properties of nonlinear dynamical systems, allowing for a different perspective on the analysis of differential dynamics which helped the community to gain a deeper understanding of the subject (Mezić and Banaszuk, 2004; Mauroy, Mezić, and Yoshihiko, 2020).

The *Perron-Frobenius* operator, whose left-adjoint operator is the Koopman operator, has historically been considered more suitable for the purpose of investigating properties

of dynamical systems, and indeed it enabled the development of set-oriented methods and the study of coherent set (Dellnitz and Junge, 2002; Dellnitz, Klus, and Ziessler, 2017). However, recent works focusing on the analysis of the spectrum of these operators have revealed that the Koopman operator may have "better-behaved" eigenfunctions, which makes it more appropriate for non-conservative systems (Mauroy, Mezić, and Yoshihiko, 2020).
The latter is usually the case for controlled systems, making the Koopman operator more appealing for the control problem. It also leads to a more convenient framework for estimating the latter operator from data, which will be crucial for the development in the next chapters.

2.2 Definition of the Koopman operator and main properties

Consider an ordinary and time-invariant differential equation given by:

$$\dot{x}(t) = \frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t)) \tag{2.10}$$

where $f : X \to X,\ X \subseteq \mathbb{R}^d$.
The vector field f induces a flow map $\Phi^{t-t_0} : X \to X$ describing solutions of (2.10), for any initial condition. If $x(t)$ is a solution of the *initial value* problem (also called *Cauchy* problem) given by

$$\begin{cases} \dot{x}(t) = f(x(t)) \\ x(t_0) = x_0 \end{cases}, \tag{2.11}$$

then the same solution is described through the flow map as

$$x(t) = \Phi^{t-t_0}(x_0). \tag{2.12}$$

Although it would be more correct to define the flow on the time interval $t - t_0$ as done in (2.12), it is customary to assume that $t_0 = 0$ so that the flow is given by $\Phi^t : X \to X$. In the case of a time-invariant system indeed the flow depends only on the difference, therefore this is done without loss of generality. This convention is also adopted here, unless explicitly specified.
The Koopman operator focuses on the orbit of the dynamical system described by (2.10) with respect to a generic function $\psi : X \to \mathbb{R}$ of interest, called *observable* (Mauroy, Mezić, and Yoshihiko, 2020). The operator then describes the evolution of the whole space of observables under the action provided by the dynamics.

Definition 2.2.1 (Observable)**.** An observable is a scalar-valued function taking values on the state space, i.e., $\psi : X \to \mathbb{R}$.

The only requirement for stating the definition of the Koopman operator is to have a complete and normed vector space, whose general characterisation is given by the Banach space. The latter is indeed a vector space that allows the computation of a complete norm for its elements.

Definition 2.2.2 (Banach space)**.** A Banach space is a complete normed space $(\Psi, \|\cdot\|_\Psi)$, which is a pair consisting of a vector space Ψ over a scalar field $\mathfrak{F}$ and a distinguished complete norm $\|\cdot\|_\Psi : \Psi \to \mathbb{R}$. For the latter to be complete, it is required for the induced metric space $\mathcal{M}$ that every Cauchy sequence of points in $\mathcal{M}$ has a limit which is also in $\mathcal{M}$.

Then the Koopman operator is defined as the composition of the observable and the flow induced by the dynamical system. The following definition is general and does not require any other assumption (Mauroy, Mezić, and Yoshihiko, 2020).

Definition 2.2.3 (Koopman operator)**.** Let Φ^t be the flow induced by a dynamical system. Consider a Banach space Ψ of observables $\psi : X \to \mathbb{R}$ closed under composition, i.e., $\psi \circ \Phi^t \in \Psi$. The family of Koopman operators $\mathcal{U}^t$ associated with the flows $\Phi^t : X \to X$, $t \in \mathbb{R}^+$, is defined as:

$$\mathcal{U}^t [\psi] (x) := \psi \circ \Phi^t (x) = \psi \left(\Phi^t (x) \right), \tag{2.13}$$

$\forall \psi \in \Psi$, $\forall x \in X$.

The Koopman operator provides an alternative description of a dynamical system through the evolution of the function space of observables. This perspective allows different techniques to be exploited in the analysis of the dynamics. According to the properties which characterise the underlying system, the associated composition operator inherit some related properties and translates structural behaviour, such as the one of a *semigroup*.

Definition 2.2.4 (Semigroup)**.** A semigroup is an algebraic structure consisting of a set, together with an associative binary operation.

In particular, the flow $\Phi^t (\cdot)$ on the set X is a semigroup action of the additive group of real numbers on X, if the mapping $\Phi : X \times \mathbb{R} \to X$ is such that:

$$\Phi^s \left(\Phi^t (x) \right) = \Phi^{t+s} (x), \tag{2.14}$$

$\forall x \in X$, $\forall t, s \in \mathbb{R}$.
The following proposition shows that the semigroup property translates directly into the Koopman operator.

Proposition 2.2.5 (Semigroup property)**.** *If the flow $\Phi^t(\cdot)$ induced by a dynamical system on the set X is a semigroup with respect to $t \in \mathbb{R}$, then the family of Koopman operators associated with the same dynamical system is a semigroup of operators.*

Proof. Exploiting the semigroup property for the flow, it is easy to derive the semigroup property for the Koopman operator,

$$\mathcal{U}^{t+s}[\psi](x) = \psi\left(\Phi^{t+s}(x)\right) = \psi\left(\Phi^t\left(\Phi^s(x)\right)\right) = \mathcal{U}^t[\psi(\Phi^s(x))] = \mathcal{U}^t\left[\mathcal{U}^s[\psi]\right](x), \quad (2.15)$$

thus proving the statement. □

Other properties translates from the flow to the Koopman operator. For continuity, for example, the following hold:

- If $\Phi^t(\cdot)$ is continuous with respect to t, then $\mathcal{U}^t$ is strongly continuous in $L^2\left(\tilde{X}\right)$, where $\tilde{X} \subseteq X$ is a forward-invariant set;
- If $\Phi^t(\cdot)$ is uniformly Lipschitz continuous with respect to t, then $\mathcal{U}^t$ is strongly continuous in $C^0\left(\tilde{X}\right)$, where $\tilde{X} \subseteq X$ is a forward-invariant set.

The Koopman semigroup of operators is strongly continuous if the following holds:

$$\lim_{t \downarrow 0} \left\| \mathcal{U}^t[\psi] - \psi \right\|_\Psi = 0, \quad (2.16)$$

where $\|\cdot\|_\Psi$ is the norm given by the Banach space Ψ.
If this is the case, then it is possible to define the *infinitesimal Koopman generator* $\mathcal{L} : \tilde{\Psi} \to \Psi$, with $\tilde{\Psi}$ a dense set in Ψ. Indeed the limit:

$$\lim_{t \downarrow 0} \frac{\mathcal{U}^t[\psi] - \psi}{t} = \mathcal{L}[\psi] \quad (2.17)$$

exists $\forall \psi \in \tilde{\Psi}$, provided (2.16) holds.
The infinitesimal Koopman generator is the analogous of the time derivative of the observables under the action of the dynamics. Another characterisation of the infinitesimal generator of the Koopman operator can be given in terms of the associated differential

equation (2.10), as:

$$\mathcal{L}[\psi](x_0) = \lim_{t \downarrow 0} \frac{\mathcal{U}^t[\psi](x_0) - \psi(x_0)}{t} \tag{2.18}$$

$$= \lim_{t \downarrow 0} \frac{\psi(x(t)) - \psi(x_0)}{t} \tag{2.19}$$

$$= \left.\frac{\partial \psi}{\partial x}\right|_{x=x_0} \left.\frac{\mathrm{d}x}{\mathrm{d}t}\right|_{t=0} = \nabla\psi(x_0) f(x_0) \tag{2.20}$$

which provides a connection between the Koopman operator framework and the vector field $f(\cdot)$ of the underlying dynamics.

It should be noted that the semigroup property and hence the existence of an infinite-dimensional operator rely on the fact that the considered dynamics are *time-invariant*, as in (2.10). If the vector field were to be time-varying and therefore explicitly dependent on time, $f(t, x(t))$, this is no longer true. Indeed the *Cauchy-Lipschitz* - or *Picard-Lindelöf* - theorem does not guarantee that for a fixed state x_0 there is a unique solution (Agarwal and Lakshmikantham, 1993). For *time-varying* systems, the uniqueness of the trajectory requires to specify also the time in which the system is on that state, $x(t_0) = x_0$. Indeed by starting in the same initial state at different initial times, different trajectories are observed. This, in turn, makes the flow - and consequently the Koopman operator - explicitly dependent on the initial time, and not only on the *time interval*, as in (2.12). A solution of a time-varying initial value problem could be written as:

$$x(t) = \Phi^{(t_0, t)}(x_0), \tag{2.21}$$

and also the corresponding Koopman operator would be parameterised by two time instants, i.e.,

$$\mathcal{U}_{t_0}^t[\psi](x) := \psi \circ \Phi^{(t_0, t)}(x_0) = \psi\left(\Phi^{(t_0, t)}(x_0)\right). \tag{2.22}$$

The analysis of time-varying dynamical system is inherently more general and therefore more complex, and it is out of the scope of this Dissertation. As a matter of fact, it is a context that has not yet been explored in the literature, with only few exceptions (Guého, Singla, and Majji, 2021; Müller, Otto, and Radons, 2017; Zhang et al., 2019).

The description of a dynamical system through the Koopman operator comes at the price of turning the finite-dimensional state space to an infinite-dimensional space of functions. This however brings the benefit of having a global linear description of the dynamical system associated with the operator. To show the linearity of the generic Koopman operator, recall the following definition.

Definition 2.2.6 (Linear operator)**.** Let $\mathcal{G}$ and $\tilde{\mathcal{G}}$ be vector spaces over a field $\mathfrak{F}$. An

operator $\mathcal{O} : \mathcal{G} \to \tilde{\mathcal{G}}$ is linear if

$$\mathcal{O}\left[c_1 g + c_2 \tilde{g}\right] = c_1 \mathcal{O}\left[g\right] + c_2 \mathcal{O}\left[\tilde{g}\right] \tag{2.23}$$

$\forall g, \tilde{g} \in \mathcal{G}$, $\forall c_1, c_2 \in \mathfrak{F}$.

The next proposition proves the linearity of the Koopman operator without any other assumption.

Proposition 2.2.7 (Linearity of Koopman operator)**.** *The family of Koopman operators is linear.*

Proof. By following the definition, it holds that:

$$\mathcal{U}^t\left[c_1\psi_1 + c_2\psi_2\right] = (c_1\psi_1 + c_2\psi_2)\circ\Phi^t = c_1\psi_1\circ\Phi^t + c_2\psi_2\circ\Phi^t = c_1\mathcal{U}^t\left[\psi_1\right] + c_2\mathcal{U}^t\left[\psi_2\right] \tag{2.24}$$

as required. □

One of the most important tool in analysing linear operators is the eigendecomposition. Therefore it is useful to characterise the spectrum and the eigenfunctions of the Koopman operator.

2.3 Eigendecomposition

The spectrum analysis of linear operators is quite easy in the case of a finite-dimensional operator. The same concept is still meaningful even in the infinite-dimensional case, in which case, however, there is a significant increase in complexity. While the spectrum of a linear operator on a finite-dimensional vector space is precisely the set of eigenvalues, an operator defined on an infinite-dimensional space may have additional elements in its spectrum, and may have no eigenvalues.
The latter is defined from the notion of *resolvent set*, which is essential to introduce the spectrum of general linear operators.

Definition 2.3.1 (Resolvent set)**.** Let Ψ be a Banach space and $\tilde{\Psi}$ dense in Ψ; let also $\mathcal{O} : \tilde{\Psi} \to \Psi$ be a linear operator and $\mathcal{I}$ denote the identity operator. The resolvent set of $\mathcal{O}$ is defined as:

$$\mathrm{Res}\left(\mathcal{O}\right) := \left\{\lambda \in \mathbb{C} \mid \mathcal{O} - \lambda\mathcal{I} : \tilde{\Psi} \to \Psi \text{ has a bounded everywhere-defined inverse}\right\} \tag{2.25}$$

i.e., $\lambda \in \mathbb{C}$ is in the resolvent set if there exists a bounded operator $\bar{\mathcal{O}} : \Psi \to \tilde{\Psi}$ such that:

$$\bar{\mathcal{O}}\left(\mathcal{O} - \lambda\mathcal{I}\right) = \mathcal{I}_{\tilde{\Psi}}; \quad \left(\mathcal{O} - \lambda\mathcal{I}\right)\bar{\mathcal{O}} = \mathcal{I}_{\Psi}, \tag{2.26}$$

meaning that the inverse must defined in the whole Ψ.

The spectrum of a linear operator is then a generalisation of the set of eigenvalues of a matrix. From Definition 2.3.1 it becomes trivial to define the spectrum of a linear operator, which is given by all the elements of $\mathbb{C}$ which do not belong to the resolvent set.

Definition 2.3.2 (Spectrum of a linear operator)**.** The spectrum of a linear operator $\mathcal{O} : \tilde{\Psi} \to \Psi$ with Ψ Banach space and $\tilde{\Psi}$ dense in Ψ is defined as:

$$\mathrm{Sp}\left(\mathcal{O}\right) := \mathbb{C} \setminus \mathrm{Res}\left(\mathcal{O}\right), \tag{2.27}$$

i.e., a complex number λ is in the spectrum if it is not in the resolvent set.

If the analysis is restricted to *bounded* linear operators, the inverse operator - when it exists - is surely linear and therefore it is also bounded, from the *bounded inverse theorem* (Narici and Beckenstein, 2010). In particular, in the bounded case, three different subsets of the spectrum can be distinguished:

- The pair $(\varphi, \lambda) \in \Psi \times \mathbb{C}$ with $\varphi \neq 0$, satisfying:

  $$\mathcal{O}\left[\varphi\right](\cdot) = \lambda\varphi(\cdot) \tag{2.28}$$

  is an eigenpair of $\mathcal{O}$, with λ eigenvalue and φ eigenfunction. If λ is an eigenvalue, then $\mathcal{O} - \lambda\mathcal{I}$ is trivially not injective, therefore $\lambda \notin \mathrm{Res}\left(\mathcal{O}\right)$ so that $\lambda \in \mathrm{Sp}\left(\mathcal{O}\right)$. The set of all eigenvalues is called *point spectrum* of $\mathcal{O}$, and it is denoted as $\mathrm{Sp_p}\left(\mathcal{O}\right)$.

- The set of all $\lambda \in \mathrm{Sp}\left(\mathcal{O}\right)$ for which $\mathcal{O} - \lambda\mathcal{I}$ is not surjective and and the range of $\mathcal{O} - \lambda\mathcal{I}$ is dense in Ψ is called the *continuous spectrum*, and it is denoted as $\mathrm{Sp_c}\left(\mathcal{O}\right)$.

- The set of all $\lambda \in \mathrm{Sp}\left(\mathcal{O}\right)$ for which $\mathcal{O} - \lambda\mathcal{I}$ is not surjective and and the range of $\mathcal{O} - \lambda\mathcal{I}$ is not dense in Ψ is called the *residual spectrum*, and it is denoted as $\mathrm{Sp_r}\left(\mathcal{O}\right)$.

The Koopman operator in general has a non-empty continuous spectrum, which is typically associated with chaotic systems, i.e., systems which are highly sensitive to initial conditions.

However, by focusing on the point spectrum, the Koopman eigenvalues and eigenfunctions can be defined as follows.

Definition 2.3.3 (Eigenpair)**.** The pair $(\varphi_\lambda, \lambda)$ is composed of an eigenfunction φ_λ and the corresponding eigenvalue λ of the Koopman operator associated with the semi-group of maps $\Phi^t(\cdot)$ if they satisfy

$$\mathcal{U}^t\left[\varphi_\lambda(\cdot)\right] = \varphi_\lambda\left(\Phi^t(\cdot)\right) = e^{\lambda t}\varphi_\lambda(\cdot), \tag{2.29}$$

$\varphi_\lambda \in \Psi \setminus 0$, $\lambda \in \mathbb{C}$.

If the semigroup of Koopman operators is strongly continuous, the same definition can be given in terms of the infinitesimal Koopman generator $\mathcal{L}$, as:

$$\mathcal{L}\left[\varphi_\lambda(\cdot)\right] = \lambda\varphi_\lambda(\cdot). \tag{2.30}$$

By exploiting the identity in (2.20), it is also possible to relate the eigenvalue equation with the dynamics in (2.10), through:

$$f(\cdot)\nabla\varphi_\lambda(\cdot) = \lambda\varphi_\lambda(\cdot). \tag{2.31}$$

These definitions refers to the eigenvalues belonging to the point spectrum $\mathrm{Sp_p}(\mathcal{U})$ of the Koopman operator, which is in general an infinite set. Indeed if λ_1, $\lambda_2 \in \mathrm{Sp_p}(\mathcal{U})$, then $c_1\lambda_1 + c_2\lambda_2 \in \mathrm{Sp_p}(\mathcal{U})$ provided that the associated eigenfunction $\varphi_{\lambda_1}^{c_1}\varphi_{\lambda_2}^{c_2} \in \Psi$.
As will be clear in Chapter 5, the possibility of characterising the space in which the Koopman operator acts through its eigenfunctions, reveals useful properties of the latter space. Therefore a characterisation of the space with eigenfunction is sought in the following.
As is usually done, the Zermelo-Fraenkel set theory is considered as a foundation for everything that has been said so far and will be said later. In particular, the extended set of axioms is assumed to hold, which also includes the axiom of choice (Zermelo, 1904; Fraenkel, Bar-Hillel, and Levy, 1973).

Axiom 2.3.4 (Axiom of choice)**.** *For any set F of nonempty sets, there exists a choice function $\mathfrak{f}$ that is defined on F and maps each set of F to an element of the set.*

By choosing to rely on the axiom of choice, every normed space admits a basis. Therefore also the Banach space Ψ of functions admits a basis (Jech, 2008).
The Koopman operator eigenfunctions may provide a basis for the space of observables Ψ, under some technical assumptions. For instance, this holds true for many *Hermitian*

operators, such as *Strum-Liouville* operators, and *Hamiltonian* operators (Kusse and Westwig, 2006). For the Koopman operator, a sufficient conditions is that the dynamics are integrable and defined in a compact space (Mauroy, Mezić, and Yoshihiko, 2020). In a more specific case, if the dynamics and Ψ are such that the Koopman operator is bounded, compact and self-adjoint - which is the case for measure-preserving dynamical systems, as it will be discussed in Section 2.4 - then the *Hilbert-Schmidt* theorem holds, and the eigenfunctions are guaranteed to form an orthonormal basis with countable non-zero eigenvalues (Renardy and Rogers, 2004; Royden and L, 1988). Therefore any observable in the Banach space Ψ can be expressed in terms of eigenfunctions. This expansion leads to the definition of *Koopman modes*.

Definition 2.3.5 (Koopman modes)**.** Let the span of Koopman eigenfunctions $\{\varphi_{\lambda_i}\}_{i=1}^{\infty}$ densely fills the space $\tilde{\Psi} \subset \Psi$. Then the Koopman mode expansion of $\psi \in \tilde{\Psi}$ is given by

$$\psi(\cdot) = \sum_{i=1}^{\infty} v_i \varphi_{\lambda_i}(\cdot) \tag{2.32}$$

where the coefficients v_i are the Koopman modes related to the observable ψ.

By exploiting both the eigendecomposition in (2.29) and the Koopman mode expansion in (2.32), it is possible to understand the role of eigenfunctions and Koopman modes in the evolution of the dynamics:

$$\mathcal{U}^t[\psi(\cdot)] = \mathcal{U}^t\left[\sum_{i=1}^{\infty} v_i \varphi_{\lambda_i}(\cdot)\right] = \sum_{i=1}^{\infty} v_i \mathcal{U}^t[\varphi_{\lambda_i}(\cdot)] = \sum_{i=1}^{\infty} v_i e^{\lambda_i t} \varphi_{\lambda_i}(\cdot). \tag{2.33}$$

Note that in the equation above the propagation of the system obtained through the evolution of the observable is simply given by the exponential functions governed by the eigenvalues. The Koopman modes and the eigenfunctions are indeed fixed with respect to time, being associated with invariant geometric properties of the system. The impact of the eigendecomposition can also be appreciated from the simple nature of propagation in (2.33). In the case of the identity function, the Koopman mode expansion characterise the evolution of the different initial conditions, as in (2.5). The latter intuition is the rationale behind Dynamic Mode Decomposition, a framework for estimating the Koopman operator, described in detail in Chapter 4.

2.4 Duality

Duality is a powerful concept (Gowers, Barrow-Green, and Leader, 2010) and it may be useful to gain a deeper understanding of the Koopman operator framework. It will also

turn out to be useful in the next chapters, and especially in Chapter 5, when *Reproducing Kernel Hilbert Spaces* and *integral operators* will play an important role.
It is well-known that any vector space $\mathcal{G}$ has a corresponding dual vector space consisting of all linear forms on $\mathcal{G}$, together with the vector-space structure of pointwise addition and scalar multiplication by constants. Since the Koopman operator acts on a Banach space, its dual is a space of linear forms. For completeness, the definition of linear forms is recalled here.

Definition 2.4.1 (Linear form)**.** A linear form S is a linear map from a vector space $\mathcal{G}$ to its field of scalars $\mathfrak{F}$. It is denoted as linear functional when the vector space is a space of functions.

As has already been pointed out, the dual space of a vector space can be defined as the space of linear forms. Note that the following definition is general and holds for any vector space.

Definition 2.4.2 (Dual space)**.** Given any vector space $\mathcal{G}$ over a field $\mathfrak{F}$, the dual space $\mathcal{G}^*$ is defined as the set of all linear forms $S : \mathcal{G} \to \mathfrak{F}$. The dual space itself becomes a vector space over $\mathfrak{F}$ when equipped with an addition and a scalar multiplication operations.

Given that the Koopman operator maps the Banach space Ψ in itself, then the dual operator must act on Ψ^*, which is here defined to be the set of all functionals S from Ψ to $\mathfrak{F}$. The dual operator satisfies:

$$S\left(\mathcal{U}\left[\psi\right]\right) = \mathcal{U}^*\left[S\right]\left(\psi\right), \tag{2.34}$$

$\forall S \in \Psi^*$, $\forall \psi \in \Psi$; where time dependency has been omitted to simplify the notation.
A meaningful characterisation of the dual operator can be achieved through the *Riesz representation* theorem, which allows to describe a functional through the inner product between a function, called *representer*, and the argument of the functional. The latter is a result of paramount importance in learning theory, and it will be exploited again in Chapter 5. The action of any functional is characterised through the inner product in a suitable Hilbert space.
As it will be clear later on, an important subset of Ψ^* is the one composed of bounded functionals.

Definition 2.4.3 (Bounded functional)**.** A functional $S : \Psi \to \mathfrak{F}$ defined over a generic Banach space Ψ is said to be bounded if there exists $\mathfrak{c} \in \mathfrak{F}$ such that $\forall \psi \in \Psi$, it holds $|S\left[\psi\right]| \leq \mathfrak{c} \left\|\psi\right\|_{\Psi}$.

Since it is now necessary to deal with inner products, Banach space is too general a concept, which must instead be specialised in a Hilbert space (Hewitt and Stromberg, 1965). The latter are the standard generalisation of Euclidean vector spaces to infinite-dimensional ones.

Definition 2.4.4 (Hilbert space)**.** A Hilbert space H is a vector space endowed with an inner product with values in a field $\mathfrak{F}$ which induces a complete metric space $\mathcal{M}$, with the distance function derived by the inner product.

Under these premises, it is then possible to state the Riesz theorem (Fréchet, 1907; Riesz, 1907; Halmos, 1950).

Theorem 2.4.5 (Riesz)**.** *Let H be a Hilbert space whose inner product $\langle h, \tilde{h} \rangle_H$ is linear in the first argument and antilinear in the second argument. For every bounded linear functional $S \in H^*$, there exists a unique $s \in H$, called the Riesz representation of S, such that:*

$$S(h) = \langle h, s \rangle_H, \tag{2.35}$$

$\forall h \in H$.

The proof of the Riesz theorem can be found in many textbooks, and different versions are available. The one in the book by Haaser and Sullivan, 1971 is very general and relies on properties of the *Riemann-Stieltjes* integral.
By restricting again the set of functionals to the one taking as input continuous function, i.e., $\mathcal{G}^* = C^0\left(\tilde{X}\right)^*$ with $\tilde{X} \subseteq X$, the Riesz representation theorem provides a direct characterisation of bounded linear functionals as integrals with respect to suitable measures. If the measure given by the representer is absolutely continuous, the latter can be expressed as an integral over the standard *Lebesgue* measure. This stems from the *Radon-Nikodym* theorem, which allows to characterise a general measure with respect to another one (Nikodým, 1930).

Theorem 2.4.6 (Radon-Nikodym)**.** *Let $(X, \mathfrak{S})$ be a measurable space consisting of a set X and a σ-algebra $\mathfrak{S}$, on which two σ-finite measures ρ and ν are defined. If ν is absolutely continuous with respect to ρ, then there exists a $\mathfrak{S}$-measurable function $\mu : X \to [0, \infty)$, such that for any measurable set $\tilde{X} \subseteq X$, it holds:*

$$\nu\left(\tilde{X}\right) = \int_{\tilde{X}} \mu \, \mathrm{d}\rho. \tag{2.36}$$

Under the assumption of the Riesz theorem and of the Radon-Nikodym one, and by considering ρ in Theorem 2.4.6 as the standard Lebesgue measure, any functional $S : C^0\left(\tilde{X}\right) \to \mathfrak{F}$ can be expressed through a density function $\mu \in L^1$, as:

$$S(\psi) = \int_{\tilde{X}} \psi(x)\, \mu(x)\, \mathrm{d}x, \tag{2.37}$$

and it is possible to define an operator $\mathcal{P} : L^1 \to L^1$ such that

$$\mathcal{U}^*[S](\psi) = \int_{\tilde{X}} \psi(x)\, \mathcal{P}[\mu](x)\, \mathrm{d}x. \tag{2.38}$$

The operator defined above is the Perron-Frobenius operator (or *transfer* operator). The latter is related with the propagation of densities according to the flow induced by the dynamical system. This is shown by the following equation:

$$\int_{\tilde{X}} \mathcal{P}[\mu](x)\, \mathrm{d}x = \int_{\Phi^{-1}\left(\tilde{X}\right)} \mu(x)\, \mathrm{d}x, \tag{2.39}$$

which corresponds to the propagation of the indicator function $\mathbb{1}_{\tilde{X}}(x)$. The Koopman operator corresponds instead to the propagation of observables, which are not densities. In particular, if μ is a probability measure, the Perron-Frobenius operator provides the evolution of an initial distribution through the dynamical system, while the Koopman operator characterise the evolution of the expectation of a specific function - the observable - over the initial distribution. The latter concept will be made precise in the following section.
The relation in (2.39) can be used as a definition for the Perron-Frobenius operator in the general space of integrable functions. A particular characteristic of a dynamical system makes the two operator interchangeable: the *measure-preserving* property.

Definition 2.4.7 (Measure-preserving dynamical system)**.** A measure-preserving dynamical system is defined as a measurable space and a measure-preserving transformation on it. In particular, given a measurable space $(X, \mathfrak{S})$, let μ be a measure on $\mathfrak{S}$. A transformation $f : X \to X$ is measure-preserving if

$$\mu\left(f^{-1}\left(\tilde{X}\right)\right) = \mu\left(\tilde{X}\right), \tag{2.40}$$

$\forall \tilde{X} \in \mathfrak{S}$. In this case, μ is called an invariant measure for f.

When the Perron-Frobenius operator satisfies the requirements of the Perron-Frobenius theorem - which typically happens in finite-dimensional spaces, and was later generalised to infinite-dimensional spaces by (Schaefer, 1966) - the eigenfunction associated with the

largest eigenvalue gives the invariant measure.
If a system is measure preserving, the invariant measure $\bar{\mu}$ can be used to define the space $\Psi = \Psi^* = L^2_{\bar{\mu}}(X)$ equipped with the norm $\|\psi\|_\Psi = \int |\psi|^2 \, \mathrm{d}\bar{\mu}$. The two operator-theoretic description are equivalent in this case. Moreover they are isometries, i.e.,

$$\|\mathcal{U}[\psi]\| = \|\mathcal{P}[\psi]\| = \|\psi\| \tag{2.41}$$

and, more generally, they satisfy $\mathcal{U}\mathcal{U}^* = \mathcal{U}\mathcal{P} = \mathcal{I}$. Additionally, when they are invertible, they are *unitary* (Mauroy, Mezić, and Yoshihiko, 2020).

2.5 Koopman for stochastic dynamical system

Many phenomena in nature can be modelled as dynamical systems, describing the evolution of the state over time. However, often it turns out that there is an intrinsic random component which affects the latter evolution, or the model is not detailed enough to explain all the complex mechanisms acting at the different levels. Indeed fluctuations are classically referred to as *stochastic* when their suspected origin implicates the action of a very large number of variables or degrees of freedom. For instance, the action of many water molecules on the motion of a large protein can be seen as noise.
It is therefore of primary importance to understand how the description of a system through the Koopman operator extends to the stochastic case.
If a deterministic dynamical system is usually described by a differential equation as in (2.10), its stochastic version is described by a *stochastic differential equation* (SDE). The latter is a differential equation in which one or more of the terms is a stochastic process, so that the solution is also a stochastic process.
Recall the following definition by Pavliotis, 2014.

Definition 2.5.1 (Stochastic process)**.** Let $\overline{T}$ be an ordered set, $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space, and $(X, \mathfrak{S})$ a measurable space. A stochastic process is a collection of random variables $x = \left\{x(t) \mid t \in \overline{T}\right\}$ such that for each fixed $t \in \overline{T}$, x is a random variable from $(\Omega, \mathcal{F}, \mathbb{P})$ to $(X, \mathfrak{S})$. The set Ω is known as the sample space, where X is the state space of the stochastic process x.

Typically, SDEs are ordinary differential equation driven by white Gaussian noise, which is formalised through the Wiener process (also called Brownian motion). That makes the latter process the most important stochastic continuous-time process. Brownian motion is a zero mean process with independent Gaussian increments.

Definition 2.5.2 (Independent increments)**.** A process x has independent increments if for every sequence $t_0 < t_1 < \cdots < t_n$, the random variables $x(t_1) - x(t_0), x(t_2) - x(t_1), \ldots, x(t_n) - x(t_{n-1})$, are independent.

Hence a Wiener process is defined as follows.

Definition 2.5.3 (Wiener process)**.** A Wiener process $W(t) : \mathbb{R}^+ \to \mathbb{R}$ is a real-valued stochastic process with almost surely (a.s.) continuous paths such that $W(0) = 0$, it has independent increments, and for every $t > s \geq 0$, the increment $W(t) - W(s)$ has a Gaussian distribution with mean 0 and variance $t - s$, i.e., the density of the random variable $W(t) - W(s)$ is

$$\Delta W(x \mid t, s) = (2\pi(t-s))^{-\frac{1}{2}} \exp\left(-\frac{x^2}{2(t-s)}\right). \tag{2.42}$$

A stochastic differential equation in general takes the form:

$$\mathrm{d}x(t) = u(x(t))\,\mathrm{d}t + \overline{\sigma}(x(t))\,\mathrm{d}W(t) \tag{2.43}$$

where $W(t)$ is a $\tilde{d}$-dimensional Brownian motion, $u : X \to X$, $X \subseteq \mathbb{R}^d$ is the *drift*, and $\overline{\sigma} : X \to \mathbb{R}^{d \times \tilde{d}}$ is the *diffusion*. To obtain the same form as (2.10), it is possible to rewrite the equation as:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t)) + \sigma(x(t))\frac{\mathrm{d}W(t)}{\mathrm{d}t} \tag{2.44}$$

where, however, the derivative of the Wiener process cannot be mathematically defined. It is useful to think about it as a Gaussian white noise, i.e., a zero-mean Gaussian process with correlation function given by $\delta(t-s)\,I$.
The precise interpretation of (2.43) is given by the integral equation for $x(t)$:

$$x(t) = \int_{t_0}^{t} u(x(s))\,\mathrm{d}s + \int_{t_0}^{t} \overline{\sigma}(x(s))\,\mathrm{d}W(s) + \mathfrak{c} \tag{2.45}$$

but a formal understanding would require to define the stochastic integral against $W(s)$, which is out of the scope of this Dissertation.
In order to introduce the Koopman operator for the stochastic dynamical systems, it is necessary to understand how to define the flow of the system, which is the solution of the initial value problem for a generic $x \in X$.
For the following lemma, the process $x(t)$ must have continuous paths (Gard and Gard, 1988; Kallenberg, 2002).

Lemma 2.5.4. *Under the assumption that the initial value problem given by:*

$$\begin{cases} \mathrm{d}x\left(t\right) = u\left(x\left(t\right)\right)\mathrm{d}t + \overline{\sigma}\left(x\left(t\right)\right)\mathrm{d}W\left(t\right) \\ x\left(t_0\right) = x_0 \quad a.s. \end{cases} \tag{2.46}$$

has a global forward-in-time solution $x\left(t\right),\ \forall t \in \mathbb{R}^+$ *with distribution* $\rho\left(t, x_0\right)$*, let* μ *be a probability measure on a* σ*-algebra* $\mathfrak{S}$ *of subsets of* X*. Define by:*

$$\rho_\mu^t\left(\tilde{X}\right) := \int_X \rho\left(t, \tilde{X}\right)\mu\left(\mathrm{d}x\right), \tag{2.47}$$

$\forall \tilde{X} \in \mathfrak{S}$*. Then a stochastic process is a solution to the initial value problem given by:*

$$\begin{cases} \mathrm{d}x\left(t\right) = u\left(x\left(t\right)\right)\mathrm{d}t + \overline{\sigma}\left(x\left(t\right)\right)\mathrm{d}W\left(t\right) \\ \rho\left(t, x_0\right) = \mu \end{cases} \tag{2.48}$$

if and only if $\rho\left(t, x\right) = \rho_\mu^t$*.*

Denote then as $\Phi^{t-t_0}\left(x_0\right)$ the solution process with initial condition $x\left(t_0\right) = x_0$ and let $\rho\left(t, x_0\right)$ its probability distribution. Analogously to the deterministic case, the sought relation is something that resembles $\mathcal{U}^t\left[\psi\left(\cdot\right)\right] = \psi\left(\Phi^t\left(\cdot\right)\right)$, for a solution of the stochastic differential equation starting from $t_0 = 0$. Let $\psi\left(\cdot\right) \in L^\infty\left(X\right)$, then for $t \in \mathbb{R}^+$ and fixed $x \in X$, $\psi\left(\rho\left(t, x\right)\right)$ is a bounded and Borel-measurable function, mapping $C^0\left(\left(\mathbb{R}^+, X\right)\right)$ into $\mathbb{R}$, therefore it is possible to take the expectation of $\psi\left(\rho\left(t, x\right)\right)$ wit respect to $\rho\left(t, x\right)$. This motivates the following definition by Klus et al., 2020.

Definition 2.5.5 (Stochastic Koopman operator)**.** Let $\psi \in L^\infty\left(X\right)$ and $x\left(t\right)$ an observable and the solution of the initial value problem with $x\left(t_0\right) = x_0$ almost surely. Define for all $t \in \mathbb{R}^+$ the stochastic Koopman operator $\mathcal{U}^t : L^\infty\left(X\right) \to L^\infty\left(X\right)$ as:

$$\mathcal{U}^t\left[\psi\left(x\right)\right] = \mathbb{E}_{\rho(t,x)}\left[\psi\left(\Phi^t\left(x\right)\right)\right], \tag{2.49}$$

$\forall \psi \in L_X^\infty\left(X\right)$ and $x \in X$.

If $\bar{\mu}$ is a stationary measure for the process $x\left(t\right)$, and therefore the system is a measure-preserving one, the Koopman operator can be extended from $L_X^\infty\left(X\right)$ to the Hilbert space $L_X^2\left(X\right)$ with the standard inner product $\langle h, \tilde{h}\rangle = \int_X h\left(x\right)\tilde{h}\left(x\right)\mathrm{d}\mu\left(x\right)$. The important thing to notice is that the Koopman operator propagates the given observable through a stochastic dynamical system by returning its expectation. This property will be crucial in what follows, and in particular in the derivation of the Koopman

formulation of the value-function for the Reinforcement Learning problem. The action of the Koopman operator is indeed constrained to yield a deterministic function, and therefore to marginalise out the source of randomness from the system.

3

Temporal resolution

Oftentimes, dynamical systems are dealt with in the discrete-time form, even if they model continuous processes. The latter occurs because of the sampling operation: this constrains data coming from the system to yield a discrete-time description. It is common belief that the best thing to do is to sample as quickly as possible, which would be closer to a continuous description. In the present Chapter it is shown that this is not true, in the case of a fixed data budget, both through theoretical derivation and empirical evidences. The choice of the sampling time leads indeed to a fundamental trade-off between approximation error and variance of the estimate.
The contents of this Chapter represent original contribution, and in particular are taken from the work by **Zanini, Francesco** et al., 2023.

3.1 Discrete samples from a continuous process

In the previous chapter the Koopman operator was presented in its continuous time formulation, which is indeed the most general one. The latter can be useful for analysis, when the continuous-time dynamics of the modelled phenomenon are explicitly available, in order to discover geometric properties of the underlying system, for instance through the eigendecomposition. However, from the next chapter onward, the emphasis will be on how to estimate the Koopman operator from data. In the Reinforcement Learning problem in fact the agent only witness transitions, and its efforts to improve performance should be based entirely on these. The Koopman operator view does not change the challenges faced in learning a nonlinear system from only discrete snapshots of the dynamics.
As explained in the next chapter, this means that it will be impossible to learn an infinite-dimensional operator, as only a finite amount of data will be available. In a similar way, also learning a continuous-time system would not be possible without leveraging some

prior knowledge, since there is no difference in the sampled transitions for every system that in those discrete points produces the same values, as it will be clear in Chapter 5. The latter problem suggest to consider the estimation of a discrete-time system, which would be aligned to the available data, although yielding a less general description of the dynamics. This is indeed what will be considered from the next chapter onward. However, first it is imperative to understand which sampling interval should be chosen to collect the aforementioned data. Usually, in Reinforcement Learning or Optimal Control this problem is neglected, and data points are assumed to arrive at a predetermined fixed clock cycle. However in many cases the practitioner has the ability of changing the sampling frequency at which data are collected. The discussion in the following sections shows that this choice has an impact on the estimation of quantities related to the system. A common belief is that a finer time discretisation always leads to better estimation of the system properties, however this is not true if a finite data budget is considered. The latter is of course a meaningful assumption, since in practice, there are always limitations on how much data can be collected, stored and processed.

There is a sizable literature on Reinforcement Learning in continuous-time systems (Doya, 2000; Lee and Sutton, 2021; Lewis, Vrabie, and Vamvoudakis, 2012; Bahl et al., 2020; Kim, Shin, and Yang, 2021; Yildiz, Heinonen, and Lähdesmäki, 2021). But these previous works have largely focused on deterministic dynamics, and do not investigate trade-offs in temporal discretisation that allow for its optimization. A smaller body of work has considered learning continuous-time control under stochastic (Baird, 1994; Bradtke and Duff, 1994; Munos and Bourgine, 1997; Munos, 2006), or bounded (Lutter et al., 2021) perturbations, but with a focus on making standard learning methods more robust to small time scales (Tallec, Blier, and Ollivier, 2019a), again without explicitly managing the temporal discretisation level. There have also been works that characterize the effects of temporal truncation in infinite horizon problems (Jiang et al., 2016; Droge and Egerstedt, 2011). Despite these prevailing topics in the literature, in the following analysis it can be appreciated that managing temporal discretisation offers substantial improvements not captured by these previous studies.

In order to understand how sampling time affects the estimation of relevant quantities of the system, a very simple setting is considered, composed of a Monte Carlo estimator for the value-function of a Langevin system. The latter will translate in the a linear system in discrete-time, which is thought to come from a linear controlled system, with a static feedback from the state. This particular setting is called Linear Quadratic Regulator and it is presented in Chapter 7. The LQR setting is a standard framework in control theory and it gives rise to a fundamental Optimal Control problem (Lindquist, 1990), which

has proven itself to be a challenging scenario for Reinforcement Learning algorithms (Tu and Recht, 2018; Krauth, Tu, and Recht, 2019). The stochastic LQR considers linear systems driven by additive Gaussian noise with a quadratic form for the cost, which is sought to be minimised by means of a feedback controller. Although it is a well-understood scenario and a closed-form of the optimal controller is known thanks to the separation principle (Georgiou and Lindquist, 2013), only recently the statistical properties of the long-term cost have been investigated (Bijl et al., 2016). The work in our paper also closely related to the now sizable literature on Reinforcement Learning in LQR systems (Bradtke, 1992; Krauth, Tu, and Recht, 2019; Tu and Recht, 2019a; Dean et al., 2018; Dean et al., 2020; Fazel et al., 2018; Gu et al., 2016). These existing works uniformly focused on the discrete time setting, although the benefits of managing spatial rather than temporal discretisation has been considered (Sinclair, Banerjee, and Yu, 2019; Cao and Krishnamurthy, 2020). Wang, Zariphopoulou, and Zhou, 2020 studied the continuous-time LQR setting but it focused on the exploration problem rather than the temporal discretisation.

The task of choosing the temporal resolution can also be understood as an experimental design problem (Chaloner and Verdinelli, 1995). By choosing the time-discretisation, the experimenter determines how to allocate measurements for a given data budget. What is perhaps surprising of the following analysis is that, for any fixed design, there is a constant approximation error (bias) that persists even when the number of data points becomes infinite. At the same time, the bias can also be managed by scarifying estimation error (variance).

3.2 Policy evaluation in continuous LQR

In the classical continuous-time Linear Quadratic Regulator (LQR), a state variable $x(t) \in \mathbb{R}^d$ evolves over time $t \geq 0$, according to the following equation:

$$\mathrm{d}x(t) = Ax(t)\,\mathrm{d}t + Ba(t)\,\mathrm{d}t + \overline{\sigma}\mathrm{d}W(t), \tag{3.1}$$

in which the drift is assumed to be linear with respect to both x and a.

The dynamical model is fully specified by the matrices $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{d\times p}$ and the diffusion coefficient $\overline{\sigma} \in \mathbb{R}^{d\times\tilde{d}}$, which is independent from x. The control input $a(t) \in \mathbb{R}^p$ is given by a fixed policy π, and $W(t)$ is a Wiener process. As will always be assumed, the state variable $x(t)$ is fully observed.

For simplicity, dynamics are considered to start at $x(0) = 0 \in \mathbb{R}^d$ (Abbasi-Yadkori and Szepesvári, 2011; Dean et al., 2020). The quadratic cost functional J which defines the

LQR problem (see Chapter 7) is defined for symmetric matrices $Q \in \mathbb{R}^{d\times d}$, which is positive semi-definite, $R \in \mathbb{R}^{p\times p}$, which is positive definite, a *system horizon* $0 < \overline{\tau} \leq \infty$ and a discount factor $\gamma \in (0, 1]$, as:

$$J_{\overline{\tau}} = \int_0^{\overline{\tau}} \gamma^t \left[x(t)^\top Q x(t) + a(t)^\top R a(t) \right] \mathrm{d}t. \tag{3.2}$$

In what follows, the class of admissible controllers given by static feedback of the state will be considered, i.e.:

$$\pi(x(t)) = Cx(t), \tag{3.3}$$

where $C \in \mathbb{R}^{p\times d}$ is the static control matrix yielding the control input. As will be recalled later, it is well known that in infinite horizon systems with discounting, the Optimal Control is of the form (3.3). Since the envisaged objective is the one of policy evaluation, the specific choice of the policy plays no particular role in the following, therefore the LQR dynamics in (3.1) are further reduced to a linear stochastic dynamical system described by a Langevin equation. Using the definitions $\tilde{A} := A + BC$ and $\tilde{Q} := Q + C^\top RC$, both the state dynamics and the cost can be expressed in a more compact form, as:

$$\mathrm{d}x(t) = \tilde{A}x(t) + \overline{\sigma}\xi(t), \qquad J_{\overline{\tau}} = \int_0^{\overline{\tau}} \gamma^t x(t)^\top \tilde{Q} x(t)\, \mathrm{d}t, \tag{3.4}$$

in which the formally incorrect simplification of considering the derivative of a Wiener process as a zero-mean Gaussian white noise $\xi(t)$ has been undertaken. The latter is indeed not measurable with nonzero probability, however this does not play any role with respect to what follows.

The expected cost $V_{\overline{\tau}}$ is the expectation of the cost w.r.t. the Wiener process, i.e. $V_{\overline{\tau}} = \mathbb{E}[J_{\overline{\tau}}]$, and it represent the quantity to be estimated. In what follows, equation (3.4) is analysed, therefore only the compact representation through the Langevin process is considered.

Also, from now on, an explicit distinction will be made between the *finite-horizon setting*, where $\overline{\tau} < \infty$, $\gamma \leq 1$ and the cost is $V_{\overline{\tau}}$; and the *infinite-horizon setting* where $\overline{\tau} = \infty$, $\gamma < 1$ and the cost is V_∞. Note indeed that in the infinite-horizon case, a discount factor turns out to be necessary to yield a finite value-function, since with stochastic dynamics the running cost will never be exactly zero, and integrating for an infinite time the result would necessary be infinite.

3.2.1 Monte Carlo estimation

The main objective of policy evaluation is to estimate the expected cost from discrete-time observations. To this end, a uniform discretisation of the interval $[0, T]$ is considered, with increment Δt, resulting in $N = T/\Delta t$ time points, defined as $t_k := k\Delta t$ for $k \in \{0, 1, \ldots, N-1\}$.
Here, the *estimation horizon* T is chosen by the practitioner, in such a way that $T < \infty$ and $T \leq \overline{\tau}$ - where for simplicity it is assumed that $T/\Delta t$ is an integer. With the N points sampled from one trajectory, a standard way to approximate the integral in (3.4) is given by the *Riemann sum estimator*:

$$\hat{J}(\Delta t) = \sum_{k=0}^{N-1} \gamma^{t_k} \Delta t x(t_k)^\top \tilde{Q} x(t_k). \tag{3.5}$$

In turn, in order to estimate $V_{\overline{\tau}}$, M independent trajectories with cost estimates $\hat{J}_1, \ldots, \hat{J}_M$ are averaged, resulting in the *Monte Carlo estimator* given by:

$$\hat{V}_M(\Delta t) = \frac{1}{M} \sum_{i=1}^{M} \hat{J}_i(\Delta t) = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=0}^{N-1} \gamma^{t_k} \Delta t x(t_k)^\top \tilde{Q} x(t_k) \tag{3.6}$$

The ultimate goal of this Chapter is to understand the mean-squared error (MSE) of the Monte Carlo estimator in (3.6), for a fixed system specified by $\tilde{A}$, $\overline{\sigma}$ and $\tilde{Q}$. This is done with the objective to inform an optimal choice of the step-size parameter Δt, under the assumption of a a fixed data budget $\mathbb{B} = M \cdot N$.
Note indeed that one degree of freedom can be exploited to choose M and N. For simplicity, in the finite-horizon setting, the estimation grid is required to be chosen so that the full episode $[0, \overline{\tau}]$ is covered, which leads to the constraint $T = \overline{\tau} = N \cdot \Delta t$. The MSE is characterised as the least-squares error surface as a function of Δt and $\mathbb{B}$, given by:

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}) = \mathbb{E}\left[\left(\hat{V}_M(\Delta t) - V_T\right)^2\right]. \tag{3.7}$$

In the infinite horizon setting, i.e. $\overline{\tau} = \infty$, the *estimation horizon* T is a free variable chosen by the experimenter which determines the number of trajectories through $M = \frac{\mathbb{B}}{N} = \frac{\mathbb{B}\Delta t}{T}$. The mean-squared error for the infinite horizon setting is then given as a function of Δt, $\mathbb{B}$, and T:

$$\mathrm{MSE}_\infty(\Delta t, \mathbb{B}, T) = \mathbb{E}\left[(\hat{V}_M(\Delta t) - V_\infty)^2\right]. \tag{3.8}$$

3.3 Characterizing the Mean-Squared Error

In what follows the objective is then to characterize the least-squares error of the Monte Carlo estimator as a function of the step size Δt and the total data budget $\mathbb{B}$, as well as the estimation horizon T in the infinite horizon setting. These results will uncover a fundamental trade-off for choosing an *optimal* step size that leads to a minimal mean-squared error.

One-Dimensional Langevin Process In order to simplify the exposition while preserving the main ideas, first the results for the 1-dimensional case will be presented. The analysis for the vector case exhibits the same quantitative behavior but is significantly more involved. Therefore, for each setting, primary importance will be given to the scalar case, for which it is easier to interpret the results, and then the vector case will also be discussed.

Let then $x(t) \in \mathbb{R}$ be the scalar state variable that evolves according to following the Langevin equation:

$$\mathrm{d}x(t) = ux(t)\,\mathrm{d}t + \sigma \mathrm{d}W(t). \tag{3.9}$$

In equation (3.9), $u \in \mathbb{R}$ is the drift coefficient and $W(t)$ is a Wiener process with scalar parameter $\overline{\sigma} > 0$. As already seen, the latter is the prototypical setting for evaluating a fixed deterministic policy in the linear quadratic regulator (LQR) framework.

Throughout the whole chapter, a *stable* - or marginally stable - closed-loop is assumed for the system under analysis, i.e., $u \leq 0$. The realisation of the stochastic process, given as a sample path in episode $i = 1, \ldots, M$, is denoted as $x_i(t)$, where it is always assumed that the starting state is $x(0) = 0$ and time is $t \in [0, T]$. The expected cost then can be written as:

$$V_{\overline{\tau}} = \mathbb{E}\left[\int_0^{\overline{\tau}} \gamma^t r_i^2(t)\,\mathrm{d}t\right] = \int_0^{\overline{\tau}} \gamma^t q \mathbb{E}\left[x_i^2(t)\right]\mathrm{d}t, \tag{3.10}$$

in which $r_i(t) = q x_i^2(t)$ is the quadratic cost function for a fixed $q > 0$.

Note that J_i is a random variable with respect to the stochasticity of the system evolution; while the expected cost V is the expectation $V = \mathbb{E}[J_i]$, which is deterministic.

The Riemann sum that approximates the cost attained by the system in episode $i \in [M]$, for the scalar case, is given by:

$$\hat{J}_i(\Delta t) = \sum_{k=0}^{N-1} \Delta t q x_i^2(t_k). \tag{3.11}$$

Given data from M episodes, the Monte-Carlo estimator has the same formulation as in the general case:

$$\hat{V}_M(\Delta t) = \frac{1}{M}\sum_{i=1}^{M} \hat{J}_i(\Delta t)\,. \tag{3.12}$$

Since the square of the cost parameter q^2 factors out of the mean-squared error in (3.7) and (3.8), the latter is taken as $q = 1$ in what follows, without loss of generality.

3.3.1 Finite-horizon, undiscounted

Recall that in the finite-horizon setting we set the system horizon $\overline{\tau}$ and estimation horizon T to be the same. This implies that the estimation grid covers the full episode, i.e. $\Delta t \cdot N = T = \overline{\tau}$. Perhaps surprisingly, the mean-squared error of the Riemann estimator for the Langevin system (3.9) can be computed in closed form. The result takes its simplest form in the finite-horizon, undiscounted setting where $\gamma = 1$ and $\overline{\tau} < \infty$. This result is summarized in the next theorem.

Theorem 3.3.1 (Finite-horizon undiscounted MSE)**.** *In the finite-horizon and undiscounted setting, the mean-squared error of the Monte Carlo estimator is given by:*

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}) = E_1(\Delta t, T, u) + \frac{E_2(\Delta t, T, u)}{\mathbb{B}}, \tag{3.13}$$

where

$$E_1(\Delta t, T, u) = \frac{\overline{\sigma}^4\left(-2u\Delta t + e^{2u\Delta t} - 1\right)^2\left(e^{2uT} - 1\right)^2}{16u^4\left(e^{2u\Delta t} - 1\right)^2}, \tag{3.14}$$

$$E_2(\Delta t, T, u) = \frac{\overline{\sigma}^4 T\left[\Delta t\left(e^{2uT} - 1\right)\left(4e^{2u\Delta t} + e^{2uT} + 1\right) - \left(e^{2u\Delta t} - 1\right)\left(e^{2u\Delta t} + 4e^{2uT} + 1\right)T\right]}{2u^2\left(e^{2u\Delta t} - 1\right)^2}. \tag{3.15}$$

The proof involves computing the closed-form expressions for the second and forth moments of the random trajectories $x_i(t)$ and is provided in appendix (A.2 and A.3.1). While perhaps daunting at first sight, the beauty of the result is that it *exactly* characterizes the error surface as a function of the step size Δt and the budget $\mathbb{B}$.

In principle, for any fixed $\mathbb{B}$, it is possible to optimize Δt in order to minimize the mean-squared error by searching over possible step-sizes $\Delta t_m = T/m$ for $m = 1, \ldots, \mathbb{B}$, provided knowledge of the system parameters u, $\overline{\sigma}$ and fixed horizon T. On the other hand, the practical scope of this procedure is somewhat limited.

On the upside, as it is shown next, the underlying trade-offs can be characterized and

understood closely in several different regimes. In Section 3.4, it is evidenced through numerical experiments how these insights translate into simulations for linear and non-linear systems.
In the case of marginal stability ($u = 0$), a simpler form of the MSE emerges which is easier to interpret, therefore the particular case of $u = 0$ is considered separately. Taking the limit $u \to 0$ of the previous expression gives the following result:

Corollary 3.3.2 (MSE for marginally stable system)**.** *Assume a marginally stable system, $u = 0$. Then the mean-squared error of the Monte-Carlo estimator is*

$$\mathrm{MSE}_T\left(\Delta t, \mathbb{B}\right) = \frac{\overline{\sigma}^4 T^2}{4} \cdot \Delta t^2 + \frac{\overline{\sigma}^4 T^5}{3} \cdot \frac{1}{\Delta t \mathbb{B}} + \frac{\overline{\sigma}^4 T^2(-2T^2 + 2\Delta t T - \Delta t^2)}{3\mathbb{B}}. \tag{3.16}$$

The first part of the expression can be understood as a Riemann sum *approximation error*, which is controlled by the Δt^2 term. The second part corresponds to the *variance term*, that decreases with the number of episodes as $\frac{1}{M} = \frac{T}{\mathbb{B}\Delta t}$. The remaining terms are lower-order terms for small Δt and large $\mathbb{B}$.
For a fixed data budget $\mathbb{B}$, the step size Δt can be chosen to balance these two terms (up to lower-order terms in $\mathbb{B}$), as:

$$\Delta t^*\left(\mathbb{B}\right) := \underset{\Delta t > 0}{\arg\min}\, \mathrm{MSE}_T\left(\Delta t, \mathbb{B}\right) \approx T\left(\frac{2}{3\mathbb{B}}\right)^{1/3}. \tag{3.17}$$

From this, the optimal number of episodes can be computed, yielding:

$$M^* \approx \frac{\mathbb{B}\Delta t}{T} = \left(\frac{2}{3}\right)^{1/3} \mathbb{B}^{2/3}. \tag{3.18}$$

Note that under the assumption $\mathbb{B} \gg 1$, it also hold true that $M^* \gg 1$. This is in agreement with the implicit requirement that Δt is big enough to consider at least one whole trajectory, i.e. $\Delta t > T/\mathbb{B}$.
Consequently, the mean-squared error for the optimal choice of Δt (up to lower-order terms in $\mathbb{B}$) is

$$\mathrm{MSE}_T\left(\Delta t^*, \mathbb{B}\right) \approx 3\left(3/2\right)^{1/3} \overline{\sigma}^4 T^4 \mathbb{B}^{-2/3}. \tag{3.19}$$

In other words, the optimal error rate, as a function of the data budget, is $O\left(\mathbb{B}^{-2/3}\right)$.
Next, a similar form for Δt^* is obtained for the general case where $u \leq 0$.

Corollary 3.3.3 (Optimal step size)**.** *For $\mathbb{B} \gg 1$, the optimal step-size (up to lower-order*

terms in $\mathbb{B}$) is given by:

$$\Delta t^*(\mathbb{B}) \approx \left(-\frac{T\left(4uT - e^{4uT} + e^{2uT}(8uT-4)+5\right)}{2u^2(e^{2uT}-1)^2} \right)^{1/3} \mathbb{B}^{-1/3}. \tag{3.20}$$

Moreover, $\mathrm{MSE}_T(\Delta t^*, \mathbb{B}) \leq O\left(\mathbb{B}^{-2/3}\right)$.

Proof. From Theorem 3.3.1, it is possible to compute the leading terms in Δt of the least-squares error as:

$$E_1(\Delta t, T, u) = \frac{\overline{\sigma}^4\left(e^{2uT}-1\right)^2}{8u^2}\Delta t^2 + O\left(\Delta t^3\right), \tag{3.21}$$

$$\frac{E_2(\Delta t, T, u)}{\mathbb{B}} = -\frac{\overline{\sigma}^4 T\left(4uT - e^{4uT} + e^{2uT}(8uT-4)+5\right)}{8u^4}\cdot\frac{1}{\Delta t\mathbb{B}} + O(1/\mathbb{B}); \tag{3.22}$$

then solving for the optimal Δt^* yields the result. □

3.3.2 Discounted cost

Adding discounting ($\gamma < 1$) in the finite-horizon setting does not fundamentally change the results, however, the exact expression for the mean-squared error gets significantly more involved. In the regime where Δt is small and $\mathbb{B}$ is large, a Taylor expansion characterizes the error surface as follows:

$$\begin{aligned}\mathrm{MSE}_T(\Delta t, \mathbb{B}, \gamma) \approx{}& \frac{\overline{\sigma}^4 T}{\log(\gamma)(u+\log(\gamma))(2u+\log(\gamma))^2}\cdot\frac{1}{\Delta t\mathbb{B}} + \\ &+ \frac{\overline{\sigma}^4\gamma^{2T}\left(e^{2uT}-1\right)^2}{16u^2}\cdot\Delta t^2 + \\ &+ \frac{\gamma^T\left(e^{2uT}-1\right)\left(\gamma^T\left(e^{2uT}(2u+\log(\gamma))-\log(\gamma)\right)-2u\right)}{48u^2}\cdot\Delta t^3 + \\ &+ \frac{\overline{\sigma}^4}{144}\cdot\Delta t^4\end{aligned} \tag{3.23}$$

The approximation shows only the lowest order terms for $1/(\Delta t\mathbb{B})$, γ^T and Δt. For details on the derivation, the reader is referred to Lemma A.3.1 in appendix (A.3.2).
The above result shows that the main trade-off between Δt and $\mathbb{B}$ persists also for the discounted objective, as long as γ^T is treated as a constant relative to Δt^2 and $1/(\Delta t\mathbb{B})$. In the limit where γ^T becomes small - e.g. $\gamma^T = o(\Delta t^4)$ - the nature of the trade-off changes in that the approximation error improves to $O(\Delta t^4)$. This can be understood from the fact that under geometric discounting combined with a decaying process, the

sum of $N = T/\Delta t$ estimation errors does not suffer a factor N, thereby removing a factor of $1/(\Delta t)$ from the (non-squared) approximation error - see the appendix (A.1) for a more detailed explanation.

Vector Case Also for the vector case ($d > 1$) it is possible to exactly characterise the mean-squared error of the Monte-Carlo estimator for the value-function of the Langevin system in (3.9). The closed-form computations will however require to assume that the matrix $\tilde{A}$ governing the behaviour of the system is diagonalisable, and stable. The latter is a rather mild assumption, as it is sufficient for the system in (3.1) to be controllable to ensure satisfiability of this condition. Controllability in fact translates into the possibility of freely adjusting the eigenvalues of the closed-loop matrix $\tilde{A}$ through the choice of the controller K. This means that it is always possible to choose eigenvalues to be distinct from each other, so that $\tilde{A}$ is diagonalisable.
The explicit form of the mean-squared error, although actually computable, is given by a long expression which is not easy to interpret, and is therefore deferred to the appendix (A.4). The following theorem summarizes the result for the vector case in the form of a Taylor expansion for small Δt and large $\mathbb{B}$.

Theorem 3.3.4 (MSE in vector case)**.** *Assume $\tilde{A}$ is diagonalisable, with eigenvalues $\lambda_1, \ldots, \lambda_n$. The mean-squared error of the Monte-Carlo estimator in the finite-horizon, undiscounted setting, is*

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}) = E_1(\Delta t, T, \lambda_1, \ldots, \lambda_n) + \frac{E_2(\Delta t, T, \lambda_1, \ldots, \lambda_n)}{\mathbb{B}} \tag{3.24}$$

where

$$E_1(\Delta t, T, \lambda_1, \ldots, \lambda_n) = (\bar{\mathfrak{c}}_1 + \mathfrak{c}_1(\lambda_1, \ldots, \lambda_n) O(T)) \bar{\sigma}^4 T^2 \Delta t^2 + O\left(\Delta t^3\right) \tag{3.25}$$

$$\frac{E_2(\Delta t, T, \lambda_1, \ldots, \lambda_n)}{\mathbb{B}} = (\bar{\mathfrak{c}}_2 + \mathfrak{c}_2(\lambda_1, \ldots, \lambda_n) O(T)) \bar{\sigma}^4 \frac{T^5}{\Delta t \mathbb{B}} + O(1/\mathbb{B}) \tag{3.26}$$

The proof with the exact derivation of the constants $\bar{\mathfrak{c}}_1$, $\mathfrak{c}_1(\lambda_1, \ldots, \lambda_n)$, $\bar{\mathfrak{c}}_2$, $\mathfrak{c}_2(\lambda_1, \ldots, \lambda_n)$ can be found in appendix (A.4.1).
Note that the terms composing the MSE are very similar to the ones obtained in the scalar analysis. Indeed by comparing them with the expressions in (3.21) and (3.22), all the terms has the same order for $\Delta t, \mathbb{B}$ and T. The only difference is that in the vector case, cumbersome eigenvalue-dependent constants are involved, whereas in the scalar case the result can more easily be expressed in terms of the system parameter u.

Since the optimal choice for Δt is given by balancing the trade-off between the two terms above, E_1 for the approximation error and E_2 for the variance term, its expression is analogous to the scalar case, as shown by the following corollary.

Corollary 3.3.5 (Optimal step size in vector case)**.** *Under the assumption that* $\mathbb{B} \gg 1$*, the optimal step-size for the vector case is given by:*

$$\Delta t^* (\mathbb{B}) = \left(\frac{\bar{\mathfrak{c}}_1 + \mathfrak{c}_1 (\lambda_1, \dots, \lambda_n) O(T)}{\bar{\mathfrak{c}}_2 (\lambda_1, \dots, \lambda_n) + \mathfrak{c}_2 (\lambda_1, \dots, \lambda_n) O(T)} \right)^{1/3} T \mathbb{B}^{-1/3} + o\left(\mathbb{B}^{-1/3}\right) \tag{3.27}$$

The constants in Corollary 3.3.5 are clearly the same as in Theorem 3.3.4.
Clearly, by removing the diagonalisability assumption, the computations become more and more tedious. However, general bounds holding for the general case of a vector Langevin process with a stable matrix $\tilde{A}$ are provided in appendix (A.4.3). These results show that the mean-squared error lies in between two expressions with the same order in Δt and $\mathbb{B}$, whose difference depends only on T, and on the eigenvalues of the matrix. Both the lower and upper bounds are convex functions of Δt, narrowing down the behaviour of the step size in this general case. In particular, the lower bound can always be expressed in terms of the mean-squared error for the scalar case, emphasizing the importance of examining this special case.
Although the convex behaviour is only proven for the case of a Langevin system, the experimental results provided in Section 3.4 exhibit a similar trade-off for general nonlinear stochastic systems.
From the present analysis, it is possible to derive guidelines on how to set the step-size even for the case of nonlinear and unknown dynamics. Although the sharp order in B for the optimal step-size holds for the case of linear dynamics only, we empirically show in Section 3.4 that a similar trade-off carries on to nonlinear dynamics, and $\Delta t = \mathfrak{c} T \mathbb{B}^{-1/3}$ is a solid choice for the more general setting. While the constant c depends on the controlled dynamics (therefore on both the free dynamics and the policy), $\mathfrak{c}$ could be estimated with a small budget, in order to properly scale the value of Δt for a large-scale experiment. This approach does not require the knowledge of the dynamics beforehand, nonetheless it provides a systematic way of setting the step size Δt for any given scenario.

3.3.3 Infinite-horizon setting

The main characteristic of the finite-horizon setting is the trade-off between approximation and estimation error. Recall that in the infinite-horizon setting ($\overline{\tau} = \infty$), the

estimation horizon $T < \infty$ becomes a free variable that is chosen by the experimenter to define the measurement range $[0, T]$. Consequently the mean-squared error of the Monte-Carlo estimator suffers an additional *truncation error* from using a finite Riemann sum with $N = T/\Delta t$ terms as an approximation to the infinite integral that defines the cost V_∞.
More precisely, the expected cost can be decomposed as a part which is actually approximated, and the truncation error, as:

$$V_\infty = V_T + V_{T,\infty}, \tag{3.28}$$

where $V_T = \int_0^T \gamma^t \mathbb{E}\left[x^2(t)\right] \mathrm{d}t$ as before, and

$$V_{T,\infty} = \int_T^\infty \gamma^t \mathbb{E}\left[x^2(t)\right] \mathrm{d}t = \frac{\overline{\sigma}^2 \gamma^T}{2u}\left(\frac{1}{\log(\gamma)} - \frac{e^{2uT}}{\log(\gamma) + 2u}\right). \tag{3.29}$$

The integral is a direct calculation based on Lemma A.2.1, provided in the appendix (A.2). Thus, the mean-squared error becomes:

$$\mathrm{MSE}_\infty(\Delta t, \mathbb{B}, T) = \mathbb{E}\left[\left(\hat{V}_M(\Delta t) - V\right)^2\right] \tag{3.30}$$

$$= \mathrm{MSE}_T(\Delta t, \mathbb{B}) - 2\mathbb{E}\left[\hat{V}_M(\Delta t) - V_T\right] V_{T,\infty} + V_{T,\infty}^2, \tag{3.31}$$

where $\mathrm{MSE}_T(\Delta t, \mathbb{B}) = \mathbb{E}\left[(\hat{V}_M(\Delta t) - V_T)^2\right]$ is the same mean-squared error of the discounted finite-horizon setting.
Note that the term $V_{T,\infty}^2$ is neither controlled by a small step-size Δt nor by a large data budget $\mathbb{B}$, hence results in the truncation error from finite estimation. Fortunately, the geometric discounting ensures that $V_{T,\infty}^2 = O\left(\gamma^{2T}\right)$, which is not unexpected given that the term constitutes the tail of the geometric integral. In particular, by setting $T = \mathfrak{c} \cdot \log(\mathbb{B})/\log(1/\gamma)$ for large enough $\mathfrak{c} > 1$, it suffices to ensure that the truncation error is below the estimation variance. The latter result is summarised in the next theorem.

Theorem 3.3.6 (Infinite-horizon, discounted MSE)**.** *In the infinite-horizon, discounted setting, the mean-squared error of the Monte-Carlo estimator is given by:*

$$\mathrm{MSE}_\infty(\Delta t, \mathbb{B}, T) = \overline{\sigma}^4 T\, \mathfrak{c}(u, \gamma) \cdot \frac{1}{\Delta t \mathbb{B}} + \frac{\overline{\sigma}^4}{144} \cdot \Delta t^4 + O\left(\Delta t^5\right) + O\left(\mathbb{B}^{-1}\right) \tag{3.32}$$

where $\mathfrak{c}(u, \gamma) = \frac{1}{\log(\gamma)(u+\log(\gamma))(2u+\log(\gamma))^2}$ *and it is assumed that* $\gamma^T = o\left(\Delta t^4\right)$.

It follows that the optimal choice for the step-size is

$$\Delta t^* (\mathbb{B}, T) \approx (36\, T\, \mathfrak{c}\, (u, \gamma)\, /\mathbb{B})^{1/5}, \tag{3.33}$$

while the minimal least-squares error is

$$\mathrm{MSE}_\infty (\Delta t^*, T, \mathbb{B}) \leq O\left((T\, \mathfrak{c}\, (u, \gamma)\, /\mathbb{B})^{4/5} + \gamma^{2T}\right) \tag{3.34}$$

Lastly, note that if γ^T is treated as a constant, the cross term $\mathbb{E}\left[\hat{V}_M (\Delta t) - V_T\right] V_{T,\infty}$ in (3.31) introduces a dependence of order $O\left(\Delta t \gamma^{2T}\right)$ to the mean-squared error. In this case, the overall trade-off becomes $\mathrm{MSE}_\infty (\Delta t, \mathbb{B}, T) \approx O\left(1/(\Delta t \mathbb{B}) + \gamma^{2T}(1 + \Delta t)\right)$, and the optimal step-size is $\Delta t^* \approx \mathbb{B}^{-1/2}$.

Vector case As happened before, the mean-squared error for the vector case can be explicitly computed in closed-form assuming that A is diagonalizable. Once again the result reflects the same behaviour as in the scalar case. Conveniently, the MSE in Theorem 3.3.6 has been expressed with sharp terms in Δt and $\mathbb{B}$, while confining the dependence on the system parameter u within the constant $\mathfrak{c}$, and the impact of higher-order terms in T within $V_{T,\infty}$. This allows to state exactly the same result for the vector case, in which the constant will now depend on the eigenvalues of the matrix $\tilde{A}$, as well as the discount factor γ. These are provided in full detail in appendix (A.4.2).

Corollary 3.3.7. *For $\tilde{A}$ diagonalisable, with eigenvalues given by $\lambda_1, \ldots, \lambda_n$, the mean-squared error of the Monte-Carlo estimator in the infinite-horizon, discounted setting is*

$$\mathrm{MSE}_\infty (\Delta t, \mathbb{B}, T) = \mathfrak{c}_3 (\lambda_1, \ldots, \lambda_n, \gamma)\, \overline{\sigma}^4 \frac{T}{\Delta t \mathbb{B}} + \frac{\overline{\sigma}^4}{144} \Delta t^4 + O\left(\Delta t^5\right) + O\left(\mathbb{B}^{-1}\right), \tag{3.35}$$

under the assumption that $\gamma^T = o\left(\Delta t^4\right)$.

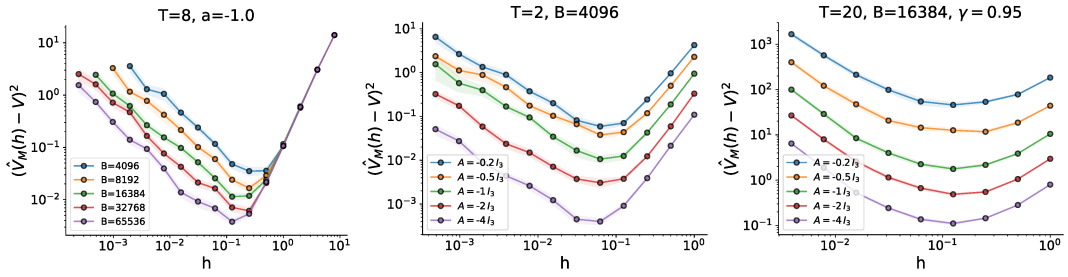
The different terms in Corollary 3.3.7 correspond to the estimation error, the approximation error and the truncation error as in the scalar case. The optimal step size choice exhibits the same dependence on T and $\mathbb{B}$ as in the scalar case, but with a different constant depending on the eigenvalues. Lastly, the general case for Langevin processes with a stable matrix $\tilde{A}$ is discussed in Appendix A.4.3.

3.4 Towards nonlinear systems: a numerical study

The trade-off identified in the previous analysis suggests that there exists an optimal choice for the temporal resolution in policy evaluation. The next goal is then to verify the derived trade-off in several simulated dynamical systems. While the theoretical analysis assumes linear transitions and quadratic cost, in the following empirical evaluation it is shown that such a behaviour for the sampling time also exists in nonlinear systems. For the sake of completeness the experimental setup is composed of simple linear quadratic systems mirroring the setting of Section 3.2, as well as several standard benchmarks environment in RL from OpenAI Gym (Brockman et al., 2016) and MuJoCo (Todorov, Erez, and Tassa, 2012). The results confirm the theoretical derivations and highlight the importance of choosing an appropriate step-size for policy evaluation.

3.4.1 Linear Quadratic Systems

First, numerical experiments on the Langevin dynamical systems are performed, in order to demonstrate the exact trade-off discovered in the previous section. The results of this experiment are shown in Fig. 3.1. For all systems, the parameters $\overline{\sigma}^2 = 1$ and $\tilde{Q} = I$ are fixed. The lines in the plot represent the sample mean $\left(\hat{V}_M(\Delta t) - V\right)^2$ and the shading represent standard error. Each data point was averaged over 50 independent runs in the scalar case and 40 in the vector case. A clear trade-off can be observed in all three plots of Figure 3.1.



(a) $n = 1$, finite horizon

(b) $n = 3$, finite horizon

(c) $n = 3$, infinite horizon

Figure 3.1: Mean-squared error trade-off in linear quadratic systems of different dimension n. The left-most plot shows the dependence of the optimal step-size on the data budget; as expected the optimal step-size decreases with more data. Middle and right plot show the MSE for different drift matrices A. Note that the optimal step-size h exhibits only a mild dependence on the scale of A.

In particular Figure 3.1a shows the MSE for the one-dimensional system with $T = 8$ and

$u = -1$. The ground truth V is calculated analytically using equation (A.37). The figure illustrates how the error changes varying the data budget, as $\mathbb{B} = \{2^{12}, 2^{13}, 2^{14}, 2^{15}, 2^{16}\}$, and also shows the improvement that can be obtained by increasing the latter budget. Indeed as $\mathbb{B}$ increases, both the error and and the optimal step size Δt^* decrease. This result aligns with the analysis shown in Theorem 3.3.1 and Corollary 3.3.3.
Fig. 3.1b and 3.1c show the experimental results for both undiscounted finite-horizon and discounted infinite-horizon d-dimensional systems. In order to compute V for the undiscounted finite-horizon system, the Riccati Differential Equation has been numerically solved using backwards induction as is standard practice. For the discounted infinite-horizon system, the solution of Lyapunov equation has been retrieved using a standard solver. Note that in the latter experiments the dimension is $d = 3$.
By fixing all parameters and running experiments on the system $\tilde{A} = \mathfrak{c}I_3$, where $\mathfrak{c} \in \{-0.2, -0.5, -1, -2, -4\}$, stable dynamics are produced for the closed loop, with different eigenvalues.
Results in both plots suggest that the impact of these eigenvalues of $\tilde{A}$ on Δt^* is mild and that the eigenvalue-dependent constants in Corollary 3.3.5 do not significantly affect the optimal step-size Δt^*. However, the eigenvalues do influence the values of the MSE achieved in each system: the latter decreases as the magnitude of the eigenvalue increases. In the infinite horizon system, the horizon needs to be large enough to manage truncation error while simultaneously being small such that multiple rollouts can be performed. In the experiments γ has been chosen large, so that a good estimate of V can be learnt. Also the effective horizon is set as $T = 1/(1-\gamma)$.

3.4.2 Nonlinear systems

Here it is empirically shown that the trade-off discovered for linear quadratic systems carries over to nonlinear systems, even with more complex cost functions. Several simulated nonlinear environments are considered: two from OpenAI Gym (Brockman et al., 2016), i.e., Pendulum and BipedalWalker, and six from MuJoCo (Todorov, Erez, and Tassa, 2012), including InvertedDoublePendulum, Pusher, Swimmer, Hopper, HalfCheetah and Ant. Note that the original environments have a fixed temporal discretisation δt, pre-chosen by the designer, which is not the same among different tasks.
To measure the effect of Δt, first all environments have been modified to run at a small discretised time $\delta t = 0.001$ as a proxy to the underlying continuous-time systems. Then, a nonlinear policy, parameterized by a neural network, is trained for each system, through the DAU algorithm (Tallec, Blier, and Ollivier, 2019b). The latter policy is used to gather data from the continuous-time proxy at intervals of $\delta t = 0.001$, which are then

down-sampled for different Δt based on the ratio of Δt and δt. This allowed to handle uniformly all environments, thus yielding the behaviour of the MSE with respect to the sampling time Δt in the same interval.

The policy employed in the environment is stable, in the sense that it produces reasonable behavior (e.g., the pendulum stays mostly upright; Ant walks forward) and does not cause early termination of episodes.

The results of the MSE of Monte-Carlo value estimation are shown in Fig. 3.2. Similar

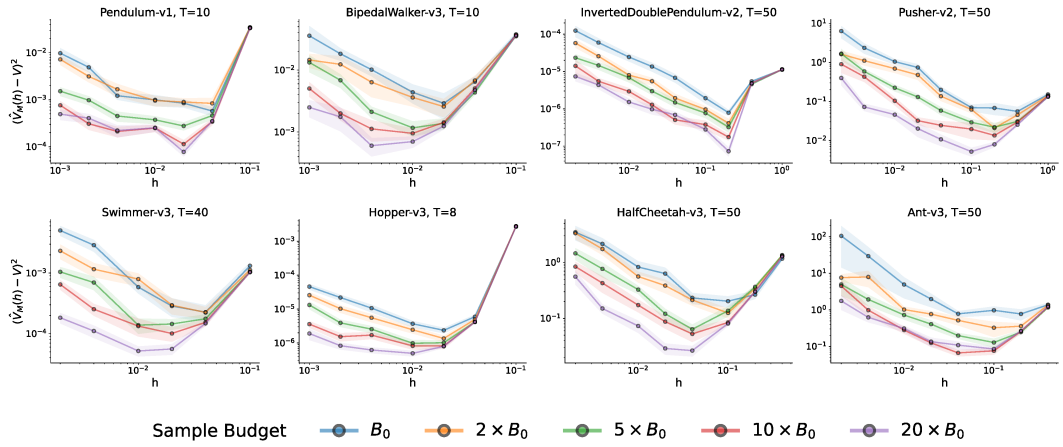


Figure 3.2: MSE of Monte-Carlo value estimation in nonlinear systems. The line and shaded region denote the sample mean and its standard error of $\left(\hat{V}_M(\Delta t) - V\right)^2$, from 30 random runs. T is the system (and estimation) horizon in physical time (seconds). $\mathbb{B}_0$ denotes the environment-dependent base sample budget, chosen such that it gives a full episode for the smallest Δt. In almost all environments the optimal step-size depends on the data budget (with the 'InvertedDoublePendulum-v2' being the only exception). In particular, the MSE as a function of Δt shows a clear minimum for choosing the optimal step-size, which generally decreases as the data budget increases.

to the linear systems case, the data budget $\mathbb{B}$ has been varied to understand how the MSE changes with the discretisation Δt. With a slight abuse of notation V and $\hat{V}$ are used to refer to the true and estimated sum of rewards instead of the cost. All these environments fall under the finite-horizon undiscounted setting. The horizon T of these experiments is chosen to be the physical time of 1000 steps in the original environments with the default δt.

These system are stochastic in the starting state, while having deterministic dynamics. Despite the different settings with respect to the theoretical analysis, a clear trade-off is evident in all systems.

4

Learning Koopman from data

In order to learn the Koopman operator from data, two simplifications are considered: the reconstruction will yield a discrete-time model, and the operator will be approximated with a finite-dimensional version. In particular, the latter is a necessary assumption in order to deal with a meaningful problem in which a finite amount of data are provided. This Chapter then discusses first the discrete-time setting and the associated Koopman operator, highlighting the differences with the continuous-time version in Chapter 2. Then the general framework for the finite-dimensional approximation is presented (Mauroy, Mezić, and Yoshihiko, 2020), and it is furthermore specialised for the two most widespread approaches to approximate the Koopman operator: Dynamic Mode Decomposition (Tu et al., 2014) and Extended Dynamic Mode Decomposition (Williams, Kevrekidis, and Rowley, 2015).

4.1 From discrete samples to discrete systems

In the previous chapter a fundamental trade-off has been discovered and analysed, which led to the characterisation of an optimal sampling time to collect observations from a continuous-time dynamical system. Although the previous result has only been shown to hold true in the case of a Monte Carlo estimator for the purpose of reconstructing the value-function of the system, there is supporting evidence to suggest that a similar trade-off would appear also for other estimators, e.g. for a *maximum likelihood estimator* identifying the parameters of the system (Aït-Sahalia, 2002).
In light of the result above, in what follows it will always be assumed to have data sampled at the best possible fixed time interval for the task at hand. This allows the estimation to be carried out in a simpler way, because under these premises the dynamical system to be estimated is totally independent from time, and maps any point of the state space into its evolution after the fixed sampling time, which is equal for all samples.

Indeed from the perspective of *discrete-time* dynamical system, dynamics evolve at separate points in time, because time is considered a discrete variable. Hence the state of the system is defined only for these discrete values, and there is no evolution in between, meaning that the time variable belongs to a set which is not dense in $\mathbb{R}$, e.g. $\mathbb{N}$.
Moreover, there is no characterisation of the vector field, again for the same reason that a differential description would not be defined. The only possible characterisation is the integral description, which gives the state at the next time step from the knowledge of the current state. By assuming a fixed interval between time steps, this is the same characterisation given by the flow, so there is no distinction in this setting:

$$x_{k+1} = f(x_k) \tag{4.1}$$
$$= \Phi(x_k), \tag{4.2}$$

meaning that the characterisation of the flow is the same as the one given by the discrete vector field.
Assuming that the discrete-time version is a sampled version of continuous-time dynamics, it is clear that the discrete-time description of a dynamical system is inherently simpler than the continuous-time characterisation, as from the latter it is always possible to recover the former by integrating the differential dynamics in the fixed time interval. Consequently, also the Koopman description of the same system will be less general when considered in discrete time than in continuous time.
However the estimation relies only on samples from the systems, which inherently yields a discrete description of the system under analysis, so that the direct derivation of a continuous-time model will just leverage some kind of prior knowledge in order to fill the gaps between samples.
In the field of *System Identification*, the main task is to reconstruct the dynamics of the system, given discrete samples of the transitions (Söderström and Stoica, 1988; Ljung, 1999; James et al., 2013). This is classically done through the *Prediction Error Minimisation* method, while assuming a fixed model class which should represent the underlying system. Standard parameterised classes are given by the *ARX*, *ARMAX*, *OE* and the *Box-Jenkins* models, which indeed allows to reconstruct both discrete and continuous dynamics. These models constrain the representation of the system to make the problem feasible, so that it is possible to generalise the estimation to yield continuous dynamics. One of the most widespread non-parametric alternatives is represented by *kernel methods*, which leverage a different kind of prior knowledge that can be specifically adapted to the problem under study and allows for greater flexibility. This approach will be reviewed in Chapter 5.

The identification through a structured model, which maintain versatility but restrict the hypothesis class for the regression, allows indeed the estimation to yield a continuous time dynamics, generalising over time. This is particularly clear in the parameterised case, in which discrete-time samples are used to estimate parameters, and the same model with the same parameters can be made dependent on time, so that a prediction for any time interval can be obtained, even if the regression made use only of data coming from a fixed sampling time.
For instance, a very strict assumption on the model would be to consider the true dynamics as linear, in which case the relationship between the vector field and the flow is known and given by an exponential mapping. This allows to first reconstruct the discrete-time flow, and then to generalise it to any time interval, through the description of the vector field. Given that the Koopman description of any system is always linear, the latter approach is exploited in Mauroy and Gonçalves, 2020 in order to retrieve the vector field for general non linear systems.
Nonetheless the two perspective of discrete-time and continuous-time system in this case yield the same amount of information, which is the one given by the discrete samples, and the apparent greater generality of the continuous-time model is just due to the prior knowledge. This is evident when the reconstructed model will only be used to predict the outcome of the system for a fixed interval, which is the same at which samples have been collected. There would be then no differences between the two descriptions.
Furthermore, there may be less pragmatic and more philosophical reasons to always consider discrete-time dynamical systems, especially in fields related with computer science, where nothing can physically go under the machine's clock.
That is why, from this moment onward, only the discrete-time formulation of the Koopman operator will be considered. Although the sampled version can be fairly regarded as less general, as it does not describe the evolution for the whole real parameter t, this is not true for arbitrary discrete-time systems. They indeed allow for a different kind of generality, as not all discrete-time system can be obtained by integration of a continuous-time system over the time jump. The restriction to discrete-time system can therefore be seen also as a deliberate choice of design.

4.2 Discrete-time Koopman operator

Consider then a discrete-time, autonomous and time-invariant dynamical system defined by the generic nonlinear map $f : X \to X$, where X is a finite-dimensional metric space. The dynamics are described through orbits of the state, $\Phi^{k-k_0}(x_0)$, where $x_{k_0} = x_0$ is

the initial condition. The same customary notation as in Chapter 2 will be adopted also for the discrete-time setting, i.e., if not otherwise specified, k_0. Moreover, in this case if the superscript is omitted, it is assumed that $k = 1$ so that the flow represents the evolution over 1 sampling time.
The Koopman operator framework, as in the continuous-time case, provides an alternative description of the latter system based on the propagation of observables (see Definition 2.2.1), whose evolution is indeed given by the action of the composition operator associated with the system.

Definition 4.2.1. Let Φ be the flow induced by a discrete-time dynamical system. Consider a Banach space Ψ of observables $\psi : X \to \mathbb{R}$ closed under composition, i.e. $\psi \circ \Phi(\cdot) \in \Psi$. The Koopman operator associated with the discrete-time flow $\Phi(\cdot) : X \to X$ is defined as:

$$\mathcal{U}[\psi](x) := \psi \circ \Phi(x) = \psi(\Phi(x)) \tag{4.3}$$

$\forall \psi \in \Psi$, $\forall x \in X$.

The above definition for the Koopman operator associated with a discrete-time system is essentially equivalent to the definition for the continuous-time operator, in which the parameter t generating the family of operators is fixed to the constant sampling interval Δt.
The latter makes it clear why the discrete-time version is less informative than the continuous one, since the former case can describe evolution only at predefined time instants, while the latter does not have any restrictions
The discrete-time version is clearly still a linear operator, being a special case of the continuous-time version. However the semigroup property can no longer hold, since it is no longer a parametric family of operators, which also means that there is no generator of the discrete-time Koopman operator. The latter would indeed correspond to the description obtained through the vector field, and this is not possible for discrete-time system, as already underlined.
The definition for the spectrum of the discrete-time operator has the same properties as in the continuous-time case. Also the characterisation of the Koopman modes remains unchanged. Instead, for the the eigendecomposition, there are slight differences, in the form of the eigenvalues.

Definition 4.2.2 (Discrete-time eigenpair)**.** The pair $(\varphi_\lambda, \lambda)$ is composed of an eigenfunction φ_λ and the corresponding eigenvalue λ of the Koopman operator associated

with the discrete-time map Φ if they satisfy

$$\mathcal{U}\left[\varphi_\lambda\left(\cdot\right)\right] = \varphi_\lambda\left(\Phi\left(\cdot\right)\right) = \lambda\varphi_\lambda\left(\cdot\right) \tag{4.4}$$

$\varphi_\lambda \in \Psi \setminus 0$, $\lambda \in \mathbb{C}$.

The relationship between the eigenvalues of the discrete-time Koopman operator and its continuous-time version is the same as between the eigenvalues of a linear discrete-time system and a continuous one.
The duality characterisation also holds without particular modifications with respect to the continuous-time case. In particular the discrete time Perron-Frobenius operator will propagate measures through the considered discrete-time dynamical system, and they will be available only for the chosen time intervals.
The definition of the stochastic Koopman operator for discrete-time system is analogous to the continuous one, since the former is again just a particular case of the latter. By considering indeed a discrete-time Markov chain evolving according to a discrete-time stochastic dynamical system, which is in this case explicitly denoted by $\underline{f}$, it holds:

$$x_{k+1} = \underline{f}\left(x_k\right), \tag{4.5}$$

so that the action of the Koopman operator remains equal to the continuous case, i.e.:

$$\mathcal{U}\left[\psi\left(x_k\right)\right] = \mathbb{E}\left[\psi\left(x_{k+1}\right)\right]. \tag{4.6}$$

However, some care needs to be taken in the definition of the stochastic dynamics. In particular, discrete-time Markov chain are considered, which are stochastic processes for which the following Markov property holds (Pavliotis, 2014).

Definition 4.2.3 (Markov property)**.** A stochastic process x is a Markov chain if

$$\mathbb{P}\left(x_{k+\tilde{k}} \mid \{x_l,\, l \leq k\}\right) = \mathbb{P}\left(x_{k+\tilde{k}} \mid x_k\right) \tag{4.7}$$

$\forall \tilde{k} \geq 0$.

The latter definition enables to obtain an equation for the transition probability of a discrete-time Markov chain. Its evolution is completely characterised by the initial distribution and the transition function.

4.3 Finite-dimensional approximations

As long as numerical methods are concerned, there is no hope to learn an actual infinite-dimensional operator, even if its simpler description in discrete-time is considered. Trough data coming from the system, it is indeed possible to collect only a finite amount of information, which then can be used for the purpose of system identification. However the latter will never be enough, and only in the limit for the data budget going to infinity there would be the possibility of recovering an exact description of the system.
However, the data budget is always assumed to be finite. Moreover, observations from dynamical systems are often corrupted by noise, so that even in the case of an infinite data budget the infinite-dimensional problem would be unfeasible, without resorting to prior information for the estimation.
In this section are then discussed properties of a generic finite-dimensional approximation. As already underlined, this is a crucial step in order to understand benefits and limitations of data-driven methods which allows to estimate only a finite-dimensional version of the Koopman operator and make predictions through it.
Consider a N-dimensional subspace $\Psi_D \subset \Psi$, spanned by the basis functions in a dictionary of observables $D = \{\psi_j\}_{j=1}^N$, along with a projection operator $\mathfrak{P}_{\Psi_D} : \Psi \to \Psi_D$, mapping from the general Banach space of observables onto the finite-dimensional subspace Ψ_D. A finite-dimensional approximation of the Koopman operator U_D is then given by:

$$U_D = \mathfrak{P}_{\Psi_D} \mathcal{U}|_{\Psi_D} : \Psi_D \to \Psi_D \tag{4.8}$$

where $\mathcal{U}|_{\Psi_D}$ is the restriction of the Koopman operator to the space Ψ_D.
Clearly, even if the infinite-dimensional space of observable Ψ is invariant under the action of the Koopman operator, the subspace given by Ψ_D in general is not, therefore the projection is needed in order to cast the output in Ψ_D.
Define the vector of basis functions which make up the dictionary D as

$$D_\psi(\cdot) := \begin{bmatrix} \psi_1(\cdot) & \dots & \psi_N(\cdot) \end{bmatrix}, \tag{4.9}$$

stacking in a row the N observables, in which usually the argument is dropped to ease the notation if it is not worth specifying.
The function resulting from the action of the finite-dimensional operator in (4.8) will surely belong to the linear span of the functions contained in the dictionary, so that the projection operator can be decomposed as

$$\mathfrak{P}_{\Psi_D} = D_\psi \mathcal{E}, \tag{4.10}$$

in which the map $\mathcal{E} : \Psi \to \mathbb{C}^N$ is a bounded linear operator which yields the coordinates of the projection of ψ in the basis $\{\psi_j\}_{j=1}^N$, and $D_\psi : \mathbb{C}^N \to \Psi_D$ provide the basis for the latter coefficients. In particular, if Ψ is a Hilbert space and $\mathfrak{P}_{\Psi_D}$ is an orthogonal projection, then $\mathcal{E}[\psi] = 0$ for all $\psi \perp \Psi_D$.
Since the projection $\mathfrak{P}_{\Psi_D}$ is just an identity operator on Ψ_D, it holds that:

$$U_D[\psi] = \mathfrak{P}_{\Psi_D}\mathcal{U}|_{\Psi_D}[\psi] = \mathfrak{P}_{\Psi_D}\mathcal{U}\mathfrak{P}_{\Psi_D}[\psi] = D_\psi\mathcal{E}\mathcal{U}D_\psi\mathcal{E}[\psi] = D_\psi U\mathcal{E}[\psi] \tag{4.11}$$

$\forall\psi \in \Psi_D$, where

$$U := \mathcal{E}\mathcal{U}D_\psi \tag{4.12}$$

is still a finite-dimensional approximation of the Koopman operator $U : \mathbb{C}^N \to \mathbb{C}^N$, defined on another space, the space of coefficients. The latter indeed maps the coefficients for the original observable ψ into coefficients of the projection of the composition $\psi \circ f$ in the same subspace given by the linear span of the basis functions given by D. Therefore it actions does not differ from the one provided by U_D. They perfectly match the behaviour of the original Koopman operator in the case that both the input and the output functions belong to Ψ_D. If instead the composition does not belong to Ψ_D, its projection in the latter space is then returned.
A direct connection between the finite-dimensional operator acting on coefficients U and the original operator $\mathcal{U}$ can be given through the map $\mathcal{E}$, as:

$$\mathcal{E}\mathcal{U}[\psi] = U\mathcal{E}[\psi]. \tag{4.13}$$

For $\psi = \psi_i$ it holds that $\mathcal{E}[\psi_i] = e_i$ - where e_i is the i-th unit vector - so that the i-th column of U contains the coordinates of $\mathfrak{P}_{\Psi_D}\mathcal{U}[\psi_i] = U_D[\psi_i]$ in the basis function given by D.
The finite-dimensional operator U is exactly the maps that *Extended Dynamic Mode Decomposition* aims to estimate, as reviewed in Subsection 4.4.2.

4.3.1 Eigendecomposition of finite-dimensional approximations

The eigendecomposition of a finite-dimensional operator is simpler with respect to the infinite-dimensional case, because its spectrum is only composed of eigenvalues. However it is not obvious to state that there is a clear relationship between the eigendecomposition of the Koopman operator and the eigenpairs of its matrix approximation given by U_D. Clearly it would be desirable that the eigenvalues and the eigenvectors of U_D somehow yielded an approximation of the eigenvalues and the eigenfunction of $\mathcal{U}$.

Unfortunately it turns out that the eigenfunctions $\tilde{\varphi}_{\lambda_i}$ of the finite-dimensional operator U_D are not necessarily a projection of the corresponding eigenfunctions φ_{λ_i} of the original operator $\mathcal{U}$ onto the subspace Ψ_D in which the approximation is defined. The latter is indeed true only when the Koopman operator commutes with the projection operator $\mathfrak{P}_{\Psi_D}$, i.e., when the subspace given by the linear span of basis function is invariant under the action of the whole operator $\mathcal{U}$. In that case it holds:

$$\tilde{\varphi}_{\lambda_i} = \mathfrak{P}_{\Psi_D}\left[\varphi_{\lambda_i}\right] = \varphi_{\lambda_i}. \tag{4.14}$$

However, note that the equality defining the eigenpair in the finite-dimensional space, i.e.,

$$U_D\left[\tilde{\varphi}_{\lambda_i}\right] = \tilde{\lambda}_i \tilde{\varphi}_{\lambda_i} \tag{4.15}$$

along with (4.10) and (4.11), implies that

$$D_\psi U \mathcal{E}\left[\tilde{\varphi}_{\lambda_i}\right] = \tilde{\lambda}_i D_\psi \mathcal{E}\left[\tilde{\varphi}_{\lambda_i}\right] \tag{4.16}$$

which in turn is equivalent to

$$U\mathcal{E}\left[\tilde{\varphi}_{\lambda_i}\right] = \tilde{\lambda}_i \mathcal{E}\left[\tilde{\varphi}_{\lambda_i}\right]. \tag{4.17}$$

The latter equation shows that the eigenvalues of the finite-dimensional approximation of the Koopman operator U_D are in fact also the eigenvalues of the coefficient matrix U; and that the corresponding eigenfunctions of U_D are realised through the coefficients given by the right eigenvectors in the basis provided by the dictionary D.
The same relationship holds between Koopman modes and left eigenvectors.
A more precise discussion on the convergence on the eigenvalues of the finite-dimensional approximation to those related to the original operator, can be found in a recent work by Korda and Mezić, 2018.

4.4 Data-driven approximations

The approach described in the previous section allows to retrieve a finite-dimensional approximation of the Koopman operator, so that predictions for the evolution of any observable are quite straightforward.
There is another line of methods which directly seek instead an approximation of eigenvalues and then obtain eigenfunctions and Koopman modes through projection theorems (Arbabi and Mezić, 2017a; Budišić, Mohr, and Mezić, 2012; Mezić, 2005; Mezić, 2013;

Mezić and Banaszuk, 2004; Mohr and Mezi'c, 2014). The latter make use of *generalised Laplace Averages* and its framework is more suitable for the purpose of analysis. Therefore such methods are not addressed in this Dissertation.

The first solution to the problem of approximating the Koopman operator from data was given by *Arnoldi*-type - or *Krylov–Rayleigh–Ritz* - algorithms, which utilises *Krylov* subspaces of the state space, and construct a companion matrix mimicking the behaviour of the Koopman operator. The latter method relies on data organised in time series, i.e., a long trajectory needs to be collected in order for this approach to work. In particular, if a stream of M datapoints are gathered from the system, the last state x_M is approximated with a linear combination of the previous $M-1$ observations.

The latter method was then later developed into the *Dynamic Mode Decomposition* by Tu et al., 2014, which is arguably the most widespread algorithm to approximate the action of the Koopman operator in a finite-dimensional subspace. The original formulation of DMD envisaged the exploitation of data coming from a trajectory, but it was later generalised to address the more general case in which snapshots of the system are available. Dynamic Mode Decomposition can be considered as a finite-dimensional spectral approximation of the Koopman operator associated with the dynamics under study (Arbabi and Mezić, 2017a), which allowed it to play an important role in the study of complex phenomena, especially in fluid dynamics (Rowley et al., 2009; Lele and Nichols, 2014; Le Ngo, See, and Phan, 2017; Berger et al., 2015; Mann and Kutz, 2016). Williams, Kevrekidis, and Rowley, 2015 proposed an extension of the latter method, called Extended Dynamic Mode Decomposition, which aimed to extend the reconstruction of the dynamics to a more general function space. If with DMD the approximation takes place on the space spanned by linear functions of the state, EDMD method frame the problem as a regression over coefficients for a user-defined dictionary of function, as seen in Section 4.3.

Both perspectives have their merits and defects. DMD approach relies on a simple subspace, which can be further reduced to consider just the most important components, through *Singular Values Decomposition*, so that it yields fast computations. This is the reason behind its widespread use for the linear analysis of a nonlinear systems. However, for many system, the subspace of linear functions of the state is not enough to capture a meaningful description of the nonlinear behaviour (Tu et al., 2014; Williams, Kevrekidis, and Rowley, 2015). The latter problem is what EDMD method seeks to overcome by relying on a larger dictionary of functions, including also explicitly nonlinear observables, so that it is easier to reconstruct the generic system. However this makes the method less favourable from a computational point of view, since in order to approximate reasonably

well the evolution of the observable, a huge number of functions is required. This leads to a regression problem which becomes computationally challenging.
A unified description of these two approaches is given in Chapter 6, Section 6.3, through the exploitation of kernel methods (see Chapter 5). This perspective will allow to mitigate the problem of selecting a the dictionary of function, which will be reduced to the selection of the kernel function.

4.4.1 DMD

Originally, Dynamic Mode Decomposition was formulated in terms of a companion matrix, underlying its connection to Arnoldi-type algorithms. However the same algorithm can be retrieved in a simpler formulation, making it easier to understand its connections with Extended Dynamic Mode Decomposition. The latter allows also to frame both methods in a dual way as done by Mauroy and Gonçalves, 2020, and further explored in Chapter 6, Section 6.3. The more recent formulation relies on M snapshots of the system given by $\{\overline{x}_i, \overline{y}_i\}_{i=1}^{M}$, where $\overline{y}_i = f(\overline{x}_i)$, so that the following matrices can be defined,

$$\overline{x} = \begin{bmatrix} \overline{x}_1 & \cdots & \overline{x}_M \end{bmatrix}, \qquad \overline{y} = \begin{bmatrix} \overline{y}_1 & \cdots & \overline{y}_M \end{bmatrix}. \tag{4.18}$$

The latter datapoints are thought to be propagated through an unknown linear operator $U : \mathbb{R}^d \to \mathbb{R}^d$, through the relation:

$$\overline{y}_k = U_{\text{DMD}} \overline{x}_k, \tag{4.19}$$

$\forall k \in [1, M]$.
The estimate of the Koopman operator is given by the least squares solution, i.e., the one which minimises the Frobenius norm $\|U_{\text{DMD}}\overline{x} - \overline{y}\|_F$:

$$U_{\text{DMD}} = \overline{y}\,(\overline{x})^{\dagger}, \tag{4.20}$$

where $(\overline{x})^{\dagger}$ denotes the Moore-Penrose pseudoinverse of $\overline{x}$.
The more recent definition of DMD procedure includes also the computation of the *reduced* Singular Value Decomposition for the matrix $\overline{x}$, in order to deal with the problem in lower dimensional subspace. The dimension of the problem is then the rank at which the reduced SVD is truncated, which is an hyper-parameter of the algorithm, playing a role similar to regularization.
The matrix $\overline{x}$ is indeed by designed ill-conditioned, i.e., it is nearly rank deficient. The introduction of the SVD allows indeed to overcome the latter issue, which is particularly

troublesome in presence of noise, by reducing sensitivity on variations of the input.
The reduced version of the Koopman matrix U_{DMD} allows to quickly compute eigenvalues and eigenvectors, which then can be projected back to the original space, as explained by Tu et al., 2014.
Note that the DMD approach requires that the dimension of the state space is much greater than the available datapoints, i.e. $M \leq d$, that the action of the linear operator is empirically known only in a small-dimensional subspace.
If through the presented formulation the DMD algorithm approximate a mapping from values of input locations $\overline{x}$ to values of output locations $\overline{y}$, the one given by Mauroy and Gonçalves, 2020 seeks for a mapping between values of observables evaluated in input locations $\psi_i(\overline{x})$ to values of observables evaluated in output locations $\psi_i(\overline{x})$, for some predetermined observables $[\psi_i\}_{i=1}^N$.

4.4.2 EDMD

Extended Dynamic Mode Decomposition is another popular method which generated from the previous algorithm. It approximates the behaviour of the Koopman operator over a predetermined function space given by the linear span of the elements in a dictionary D.
Consider in this regard again a set of snapshot pairs $\{\overline{x}_i, \overline{y}_i\}_{i=1}^M$, which are the only data coming from the system, on which the estimate should rely. The method relies on a predefined dictionary of function, as done in (4.3), which is composed of user-chosen observables $\psi_i(\cdot)$ that should be able to describe the propagation of the generic observable $\psi(\cdot)$ through a linear combination.
Given then a dictionary of functions $D = \{\psi_j\}_{j=1}^N$, it is possible to define the matrices of evaluations on the available datapoints for all the function in the dictionary, which are given by:

$$D_{\overline{x}} = \begin{bmatrix} \psi_1(\overline{x}_1) & \psi_2(\overline{x}_1) & \dots & \psi_N(\overline{x}_1) \\ \psi_1(\overline{x}_2) & \psi_2(\overline{x}_2) & \dots & \psi_N(\overline{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\overline{x}_M) & \psi_2(\overline{x}_M) & \dots & \psi_N(\overline{x}_M) \end{bmatrix},$$
$$D_{\overline{y}} = \begin{bmatrix} \psi_1(\overline{y}_1) & \psi_2(\overline{y}_1) & \dots & \psi_N(\overline{y}_1) \\ \psi_1(\overline{y}_2) & \psi_2(\overline{y}_2) & \dots & \psi_N(\overline{y}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\overline{y}_M) & \psi_2(\overline{y}_M) & \dots & \psi_N(\overline{y}_M) \end{bmatrix}. \tag{4.21}$$

The approximation of the Koopman operator given by EDMD is a finite-dimensional operator acting in the space provided by the linear span of the observables in the dictionary. The rationale behind the latter algorithm is again the minimisation of the square loss, so that the solution takes then the form:

$$U_{\text{EDMD}} = D_{\overline{x}}^{\dagger} D_{\overline{y}}. \tag{4.22}$$

As already stated, the latter represent the mapping from the coefficients realising the considered observable in the prescribed basis function to the coefficients which best approximate the composition of the observable and the discrete-time dynamics, in the same basis.
The approximation of the Koopman eigenvalues, eigenfunctions and modes has already been discussed in Section 4.3, and in particular they are given by eigenvalues, right eigenvectors and left eigenvectors of U_{EDMD}, respectively.
It can be noted that by selecting a particular dictionary of functions, the two methods described in this section can be seen under the same framework. In this regard consider the case in which the N basis functions are chosen as the M scalar projections that pick the i-th entry of the vector, for $i = 1, \dots, N$, with $N = d$. The matrices characterising the EDMD then become:

$$D_{\overline{x}} = \overline{x}^{\top}, \qquad D_{\overline{y}} = \overline{y}^{\top}. \tag{4.23}$$

Hence it becomes clear that DMD is trying to approximate the Koopman operator with linear functions of the state, while EDMD approach allows to introduce a more general behaviour for the basis functions defining the space in which the regression takes place. Note also that for the EDMD to work properly, the number of basis functions should be smaller than the available datapoints i.e., $M \geq N$, so that the least squares procedure makes sense, and yield the solution of minimum norm. Otherwise there would be too many degrees of freedom for the regression problem. The formulation given in Chapter 6, Section 6.5 provides a way to overcome this limitation as the dictionary of functions will always be implicitly chosen with the same dimension as the number of datapoints.
By considering every dimension of the state space as an observable, as in (4.23), it is easy to understand that the DMD assumption of $M \leq d$ can be interpreted as $M \leq N$. This means that DMD actually operates under a different regimes with respect to EDMD, which is in agreement with the dual perspective. If instead a generic dictionary of function is considered, as done for the Extended Dynamic Mode Decomposition, the DMD method can be generalised to yield:

$$U_{\text{DMD}} = D_{\overline{y}} D_{\overline{x}}^{\dagger} \tag{4.24}$$

as given by Mauroy and Gonçalves, 2020.

5

Kernel methods and RKHS

This Chapter presents the other fundamental mathematical tool which is required to understand subsequent development of this Dissertation, i.e., Reproducing Kernel Hilbert Spaces and kernel methods. Their introduction is considered from the perspective of operator theory, in order to make the discussion coherent with the characterisation of the Koopman operator in Chapter 2. Although the following contents clearly do not qualify as an original contribution, the following presentation can perhaps be regarded as an accurate and compact review of the concepts related to RKHSs, which is hard to find in the literature.

Reproducing Kernel Hilbert Spaces theory provides a finite-dimensional solution to an infinite-dimensional problem, through the Representer theorem. The latter can be exploited to make the problem of regularised regression well-posed, while maintaining the hypothesis space infinite-dimensional. In order to better link RKHSs to the Koopman operator framework, a characterisation of these spaces through the spectral decomposition of the kernel integral operator is also discussed.

5.1 An infinite-dimensional problem

To motivate the introduction of *Reproducing Kernel Hilbert Spaces*, the starting point will be to consider the *inverse problem* of reconstructing a whole function from discrete measurements. This is clearly an *ill-posed* problem since the solution is not unique, and if the search space is not restricted to a particular set, there are always an infinite number of function fitting the data perfectly (Baumeister, 1987; Vogel, 1987; Sabatier, 1987).

There are different ways to handle this issue, depending on the final objective of the estimation. For instance, in Chapter 3 one of the simplest solution has been considered, namely the zero-order hold approximation, which prescribes to hold each sample value until the next one is provided (Oppenheim, Willsky, and Young, 1983). Even if the latter

estimate can be very far from the true function in between two datapoints, it provides a viable solution, which is however more often employed as a way of approximating a certain quantity - as done in Chapter 3 - rather than for reconstructing the true behaviour of the unknown function. The relation reconstructed in that way would indeed be surely highly discontinuous and also very sensitive to the *noise* affecting the measurements.
The RKHS framework allows to restrict the search space for the reconstruction in a meaningful manner, in such a way that certain specific properties can be enforced for the final estimate. For instance, by considering only continuous kernels, the result of the identification procedure will surely be continuous (Berlinet and Thomas-Agnan, 2004).
In order to frame the problem, assume that M measurements of the true function h are available as *input locations* $\{\overline{x}_i\}_{i=1}^M$ and *output locations* $\{\overline{y}_i\}_{i=1}^M$, which are bound by the following relationship:

$$\overline{y}_i = h\left(\overline{x}_i\right). \tag{5.1}$$

The problem to be solved is the reconstruction of the true function given the discrete snapshots above, which can be formalised as

$$\hat{h} := \underset{h \in H}{\arg\min} \left\{ \sum_{i=1}^{M} \left\| \overline{y}_i - h\left(\overline{x}_i\right) \right\| \right\}, \tag{5.2}$$

for a generic norm $\|\cdot\|$. In this case it is easy to see that the regression task is in fact ill-posed, as long as H denotes a generic function space.
The latter framework indeed fits into the generic problem of minimising a cost function over a function space, which without particular restriction is infinite-dimensional. This differs from the one in classical analysis because here the optimisation is considered with respect to a function and not to a point, what usually is called a *variational* problem in mathematics (Goldstine, 2012). Then, in general, the search space is an infinite-dimensional one, which makes things more complicated. Therefore something needs to be done in order to address this more general setting; possibly with the aim of adapting the known optimisation theory in finite-dimensional spaces (Du, Pardalos, and Wu, 2009).
Reproducing Kernel Hilbert Spaces are particular Hilbert spaces which allows to embed some prior knowledge into the problem, and to accordingly shrink the hypothesis space. This is accomplished by selecting a kernel function, which will shape the hypothesis space for the reconstruction. As already mentioned, one type of a priori knowledge that is usually enforced through this method is the assumption on the regularity of the function to be estimated, usually continuity. In order to favour functions which comply with the induced prior knowledge, a penalty on the norm of the regressor is introduced, leading to a regularised regression task. The *Representer* theorem by Schölkopf, Herbrich, and

Smola, 2001 guarantees that the solution to the latter minimisation - if exists - is unique, so that the problem becomes well-posed. The RKHS framework allows also to cope with the problem of noisy measurement. The measurement model in (5.1) can indeed be relaxed by the addition of a random variable, which takes into account disturbances. This results in a weighted norm for the fitting term in (5.2), adjusted through the estimated noise affecting data, so that the procedure can be made robust to perturbations in the measurements (Mendelson and Neeman, 2010).

5.2 A finite-dimensional solution

The first thing to consider to deal with infinite-dimensional problems is a vector space on the field $\mathfrak{F} = \mathbb{R}$. As was made clear in Chapter 2, the latter is in general given by a Banach space, or by a Hilbert space, if there is a requirement for an inner product to be defined.
Since the current framework develops under the ZFC set of axioms, the axiom of Choice (Axiom 2.3.4) is assumed to hold. As already state in Chapter 2, the latter implies that any normed space admits a basis. Although usually the *Zorn's lemma* (see Moore, 2012) is quoted to state that any Hilbert space has an orthonormal basis, it has been shown to be a consequence of the Axiom of choice (Dunford et al., 1958), therefore no other assumption is needed. The following clarifies the definition of such a basis.

Definition 5.2.1 (Basis of a normed space)**.** The $\{\psi_i\}_{i\in I}$ elements of Ψ, with I a general (possibly uncountable) set of indexes, are said to be a basis of a normed space Ψ, if $\forall\psi \in \Psi$ and $\forall\epsilon > 0$ there exists $\tilde{I} \subseteq I$ and $\{c_i\}_{i\in\tilde{I}}$ such that $\|\psi - \sum_{i\in\tilde{I}} c_i\psi_i\| \leq \epsilon$.

As the set of indexes I may be uncountable, the definition is quite general. There are indeed cases of normed spaces with an uncountable basis, e.g. the set of all $\psi : \mathbb{R} \to [0, 1]$ endowed with the uniform norm. However, the majority of the results in this chapter hold for Hilbert spaces which admits a countable basis. Many useful properties in fact descend from this assumption, therefore it is useful to formally characterise these spaces. The latter are called *separable* Hilbert spaces (Prugovečki, 1971).

Definition 5.2.2 (Separable Hilbert space)**.** A Hilbert space is said to be separable provided it contains a dense countable subset.

All separable infinite-dimensional Hilbert spaces are isometrically isomorphic to l^2, which is the space of sequences with finite norm. The proof can be found in the book by Rudin, 1987. This relationship will be important for what follows. To see why this hold, it is necessary to introduce the definition for *isometric operators*.

Definition 5.2.3 (Isometric operator)**.** Let H_1 and H_2 be two Hilbert spaces. The operator $\mathcal{O} : H_1 \to H_2$ is said to be isometric if it preserves inner products, i.e. $\forall h_1,\ h_2 \in H_1$, then $\mathcal{O}[h_1], \mathcal{O}[h_2] \in H_2$ are such that $\langle h_1, h_2 \rangle_{H_1} = \langle \mathcal{O}[h_1], \mathcal{O}[h_2] \rangle_{H_2}$.

The characterisation of separable Hilbert spaces through their relationship with the l^2 space gives an idea of the finite nature of these particular Hilbert spaces and how they are close to the discrete-time setting. Also, it allows to think about the l^2 space as an intuitive model for any separable Hilbert space.
In particular it can be shown that the inner product of a generic Hilbert space can be computed through the inner product in l^2, by means of the isometric map in Definition 5.2.3. Since the inner product is the most important feature and the defining characteristic of an Hilbert space, the latter is equivalent to say that every separable Hilbert space H has the same properties of l^2.
RKHSs have been introduced in order to solve a regression problem, which can be generalised as the following minimisation of a functional:

$$\hat{h} := \arg\min_{h \in H} \left\{ \widetilde{\mathrm{LP}}(h) \right\}, \tag{5.3}$$

with a generic $\widetilde{\mathrm{LP}} : H \to \mathbb{R}$.
Since the problem to be solved is a regularised regression, the functional $\widetilde{\mathrm{LP}}$ will be composed of a loss function and a penalty on the norm of the function induced by the Hilbert space. Therefore a more specific problem can be rewritten as:

$$\hat{h} := \arg\min_{h \in H} \{ \mathrm{LP}(S_1[h], \ldots, S_N[h], \|h\|_H) \} \tag{5.4}$$

where $\mathrm{LP} : \mathbb{R}^{N+1} \to \mathbb{R}$ is a finite-dimensional map, i.e., $N < \infty$, and the functionals are such that $S_i : H \to \mathbb{R}$, $\forall i \in [1, N]$.
It turns out that by restricting the class of functionals $S_i[\cdot]$ to linear ones, a convenient solution for the optimisation problem (5.4) is available. This is because the linearity of the operator is equivalent to other important properties: for instance it is strictly connected with *boundedness* - whose importance was already evident from Chapter 2 - and *continuity* (Narici and Beckenstein, 2010). The finite-dimensional solution for the latter infinite-dimensional problem will be given by the Representer theorem, which is the goal of the present section.
First, a characterisation of linear operators is introduced, which relates them to bounded operators.

Definition 5.2.4 (Continuous operator)**.** Let H be a Hilbert space, and $S : H \to \mathbb{R}$ a

functional defined on the same Hilbert space. S is said to be continuous if $\forall h \in H$ and $\epsilon > 0$ there exists $\delta > 0$ s.t. $\forall \tilde{h} \in H$ s.t. $\left\| h - \tilde{h} \right\|_H \leq \delta$, it holds that $\left| S[h] - S\left[\tilde{h}\right] \right| \leq \epsilon$.

The following theorem, which holds for operators defined in general Banach spaces, shows the equivalence between bounded and continuous functionals under the linearity assumption.

Theorem 5.2.5 (Equivalence between boundedness and continuity for linear operators)**.** *Let $S : \Psi \to \mathbb{R}$ be a linear functional defined over a generic Banach space Ψ. Then S is continuous if and only if it is bounded.*

If the functionals $S_i[\cdot]$ are assumed to be linear and are defined over a generic vector space G, then the properties of boundedness and continuity are equivalent, in the sense that once linearity or boundedness of the functional can be proved, the other property consequently holds.
It turns out that in the important case of finite-dimensional operators, from linearity it directly descend the boundedness of the latter operator, which is then also continuous thanks to Theorem 5.2.5. This statement is formalised in the next theorem.

Theorem 5.2.6 (Linear and finite-dimensional operators are bounded and continuous)**.** *Let Ψ and $\tilde{\Psi}$ be normed spaces, with norms $\|\cdot\|_\Psi$ and $\|\cdot\|_{\tilde{\Psi}}$. Let Ψ be finite dimensional, and $S : \Psi \to \tilde{\Psi}$ be linear. Then S is always bounded, and thus continuous.*

The proof is easy and can be found in many textbooks, as the one by Schechter, 1996. It requires the well-known argument that all norms are equivalent in a finite-dimensional space, and then it suffices to apply the Heine-Borel theorem (Williamson and Janos, 1987).
Without the finite-dimensional assumption, clearly there exist functionals which are linear but not continuous.
Note that the Riesz theorem (Theorem 2.4.5) requires the functional to be linear and bounded, in order to be expressed as an inner product of the representer. As explained in Chapter 2, the latter result is a fundamental tool in learning theory. The assumption of the theorem clearly holds true when considering linear and finite-dimensional functionals, as is clear from the previous characterisation these operators. Another theorem of absolute importance for the development of kernel methods, which characterises the solution of the regularised regression problem, is the *Representer* theorem by Schölkopf, Herbrich, and Smola, 2001, which conveniently makes use of the same assumptions.

Theorem 5.2.7 (Representer)**.** *Consider the functional $\widetilde{\mathrm{LP}} : H \to \mathbb{R}$ where H is a*

generic Hilbert space, defined as

$$\widetilde{\mathrm{LP}}\left[h\right] := \mathrm{LP}\left(S_1\left[h\right], \ldots, S_N\left[h\right], \left\|h\right\|_H\right) \tag{5.5}$$

where $\mathrm{LP} : \mathbb{R}^{n+1} \to \mathbb{R}$ *and where the* $S_i : H \to \mathbb{R}s$ *are linear and bounded functionals. Suppose also that* LP *is strictly monotonically increasing with respect to the last argument, i.e.,* $\left\|h\right\|_H$*. Define*

$$\hat{h} := \arg\min_{h \in H} \left\{\mathrm{LP}\left(h\right)\right\}. \tag{5.6}$$

Assume that there exists at least one solution of the previous problem (i.e. that the solution exists but may be not unique). Then the latter has the form

$$\hat{h} = \sum_{i=1}^{N} c_i s_i \tag{5.7}$$

where the $s_i s$ *are the representers of the functionals* $S_i s$*, i.e.* $S_i\left[h\right] = \left\langle h, s_i \right\rangle_H$*,* $\forall h \in H$*.*

The proof can be found in the work by Schölkopf, Herbrich, and Smola, 2001.
Theorem 5.2.7 state that the solution of the regularised regression problem in (5.3) is given by a finite linear combination of the representers associated with the functionals $S_i\left[\cdot\right]$ composing the objective to be minimised. Note that the search space is still infinite-dimensional, as it is given by the Hilbert space H, so that the only assumption is about the functionals. In what follows it will become clear that those functionals are related to the fitting term in the regression, and in particular they will play the role of evaluation functionals. In what follows it will be clear that RKHS are exactly those spaces in which the pointwise evaluation is linear and bounded, thus respecting the conditions of the Representer theorem.

5.3 The regularised LS

In order to appreciate the impact of the Representer theorem for the identification task, in this section a regularised regression problem in a finite-dimensional space will be considered. The finite-dimensional setting will make it easier to understand the elements involved, while the infinite-dimensional framework is addressed in Section 5.6.
In this regard let Σ_M and Σ_N be positive definite matrices in $\mathbb{R}^{M \times M}$ and $\mathbb{R}^{N \times N}$ respectively. Denote as $q \in \mathbb{R}^M$ the fixed target quantity. Then the regularised regression problem over the vectors $h \in \mathbb{R}^N$, with $A \in \mathbb{R}^{M \times N}$ as regressor, is given by:

$$\hat{h} = \arg\min_{h \in \mathbb{R}^n} \left\{\left(q - Ah\right)^\top \Sigma_M^{-1} \left(q - Ah\right) + \eta h^\top \Sigma_N^{-1} h\right\}. \tag{5.8}$$

where η is a regularisation parameter.
The latter is also called *Tikhonov* problem (Kress, 1989), and the particular kind of penalty added as a regulariser is called Tikhonov regularisation. If the solution to the unregularised problem is not unique, the simple least-squares estimation leads to an underdetermined or overdetermined solution. The Tikhonov penalty improves the conditioning of the problem, allowing for a direct numerical solution.
Note that the regression problem (5.8) satisfies the conditions of Theorem 5.2.7. In particular, the hypothesis space is the Hilbert space $\mathbb{R}^N$ endowed with the inner product $\langle h, \tilde{h} \rangle_H = h^\top \Sigma_N^{-1} \tilde{h}$, and the functionals $S_i[h]$ correspond to the multiplications $A_{[*i]}h$ (where $A_{[*i]}$ denotes the i-th column of the matrix A), which are indeed linear and bounded. The Representer theorem does not provide an explicit solution, but guarantees that the minimiser lies in the linear span of the representers. The latter can be computed from the definition of the inner product:

$$S_i[h] = \langle s_i, h \rangle_H = s_i^\top \Sigma_N^{-1} h = A_{[*i]}h, \tag{5.9}$$

so that

$$s_i = \Sigma_N A_{[i*]}, \tag{5.10}$$

and therefore the solution will be given by

$$\hat{h} = \Sigma_N A^\top c, \tag{5.11}$$

for some coefficients $c \in \mathbb{R}^M$.
In this particular regularised regression problem it is still possible to find a closed-form solution, which also helps drawing connections with the Bayesian interpretation. The regression in (5.8) can indeed be rewritten as

$$\hat{h} = \underset{h \in \mathbb{R}^N}{\arg\min} \left\{ \begin{bmatrix} (q - Ah)^\top & h^\top \end{bmatrix} \begin{bmatrix} \Sigma_M^{-1} & 0 \\ 0 & \eta \Sigma_N^{-1} \end{bmatrix} \begin{bmatrix} (q - Ah) \\ h \end{bmatrix} \right\}, \tag{5.12}$$

so that, by defining

$$\tilde{q} = \begin{bmatrix} q \\ 0 \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A \\ -I\eta \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_M & 0 \\ 0 & \Sigma_N \end{bmatrix}, \tag{5.13}$$

the minimisation can be expressed as

$$\hat{h} = \underset{h \in \mathbb{R}^N}{\arg\min} \left\{ \left\| \tilde{q} - \tilde{A}h \right\|_\Sigma^2 \right\}. \tag{5.14}$$

The latter is an easy problem and can be solved through the *weighted least-squares* approach, yielding the following solution:

$$\hat{h} = \left(A^\top \Sigma_M^{-1} A + \eta \Sigma_N^{-1}\right)^{-1} A^\top \Sigma_M^{-1} q \tag{5.15}$$

$$= \Sigma_N A^\top \left(A \Sigma_N A^\top + \eta \Sigma_M\right)^{-1} q. \tag{5.16}$$

A proof of the equality above is given in Proposition 6.4.2.
Note that the solution in (5.16) is indeed of the form prescribed by the Representer theorem, matching the expression given in (5.11).
Although until now the problem in (5.8) has been regarded as a least squares problem, it is also possible to see it through the lens of *Bayesian estimation*. The regularisation term $h^\top \Sigma_N h$ can indeed be interpreted as deriving from a Gaussian prior on h, by assuming the following stochastic linear model:

$$q = Ah + w \tag{5.17}$$

where the noise $w \sim \mathcal{N}(0, \Sigma_M)$ is Gaussian distributed, and the regression variable $h \sim \mathcal{N}(0, \eta^{-1}\Sigma_N)$ has a prior Gaussian distribution. The *maximum a posteriori* estimator can be computed in closed form, and it is also a *Minimum-variance Unbiased Estimator* (MVUE), as explained in more detail in Chapter 6. Its formulation is of course identical to (5.16).
The Bayesian perspective will be considered in order to derive the formulas for the infinite-dimensional case in Section 5.6.

5.4 Definition of RKHS

The motivation behind the introduction of Reproducing Kernel Hilbert Spaces was to solve the estimation problem in (5.2), however, without any prior assumption on the true function generating the data, it is impossible to even pose the problem in a meaningful way. In fact there exists infinitely many functions which can perfectly fit the data provided, and they could be arbitrarily different from the true function anyway.
The idea of *stability* in machine learning proved itself to be a meaningful tool as a measure to evaluate the performances of learning algorithms and to address the reproducibility issue (Stodden, Leisch, and Peng, 2014). A learning algorithm is said to be stable if it produces consistent predictions with respect to small perturbation of training samples (Sun, 2015). It is worth mentioning that stability has indeed received much attention in statistics and related branches as, for instance, *high-dimensional regression* (Meinshausen

and Bühlmann, 2010; Shah and Samworth, 2013; Liu, Roeder, and Wasserman, 2010; Sun, Wang, and Fang, 2013; Breiman, 1996; Bousquet and Elisseeff, 2002; Elisseeff, Evgeniou, and Pontil, 2005).
In the problem of regression, stability can be regarded as *regularity* of the candidate function, because if the reconstruction consider a space of regular functions, the minimiset will be less affected by a change in the single datapoint. Often this is a good prior knowledge to be embedded in the regression problem in order to avoid fitting the noise, which indeed produces sudden and subtle variations, and to keep the hypothesis space free from overly complicated functions, according to the principle of *Occam's razor* (Jefferys and Berger, 1991).
Therefore, by leveraging the latter ideas, in what follows the discussion on Reproducing Kernel Hilbert Spaces is specialised for continuous functions. Continuity is in fact the main example of regularity of a function.
Although RKHS of non continuous functions can be defined as well, choosing a Kernel in such a way that the associated RKHS is composed of continuous functions yields some useful property and it is particularly suited for the regression task. Clearly, different objective - also among regressions problem - may require a different kind of prior knowledge for the hypothesis space, which often can nonetheless be characterised through a kernel function, which demonstrates the generality of this approach (Schölkopf et al., 1997; Veillard, Racoceanu, and Bressan, 2011).
From now on continuous functions $h : X \to \mathbb{R}$ will be considered, with domain X compact, and in particular $X \subset \mathbb{R}^d$.
In this section the relationship between RKHS and the associated kernel function is discussed. The first step is clearly to define the elements involved.

Definition 5.4.1 (Kernel)**.** A kernel is a function of two arguments $K(\cdot,\cdot) : X \times X \to \mathbb{R}$ which is symmetric and positive-definite, i.e.:

$$K(\tilde{x}, x) = K(x, \tilde{x}), \quad \forall x, \tilde{x} \in X \tag{5.18}$$

and

$$\sum_{i=1}^{M}\sum_{j=1}^{M} c_i c_j K(x_i, x_j) \geq 0, \quad \text{for any } x_1, \ldots, x_M \in X \text{ and } c_1, \ldots, c_M \in \mathbb{R}. \tag{5.19}$$

The specialisation for the continuous case is denoted *Mercer* kernel.

Definition 5.4.2 (Mercer kernel)**.** A Mercer kernel is a continuous kernel.

It turns out that through the definition of the kernel $K(\cdot,\cdot)$, it is possible to uniquely characterise the space of functions given by the corresponding RKHS. The latter is indeed the result of the *Moore-Aronszajn* theorem.

Theorem 5.4.3 (Moore-Aronszajn)**.** *Given a Mercer kernel $K(\cdot,\cdot)$, then there exists a unique Hilbert space H endowed with an inner product $\langle\cdot,\cdot\rangle_H$, called Reproducing Kernel Hilbert Space, such that:*

- $K(x,\cdot) \in H$, $\forall x \in X$;
- $\langle K(x,\cdot), h\rangle_H = h(x)$, $\forall x \in X$

In addition, H contains only continuous functions, i.e. $H \subset C(x)$.

The proof can be found in the work by Aronszajn, 1950.
The second point in the Moore-Aronszajn theorem, which expresses the evaluation of h in a point $x \in X$ through the kernel function, is called *reproducing property*. The latter makes it clear that the inner product with a kernel section $K(x,\cdot)$ corresponds to the evaluation functional of any function h on the point x. Therefore, from the Riesz theorem, it is trivial to understand that the kernel section is the representer associated with the pointwise evaluation functional.
The following characterisations of the RKHS will be useful in what follows.

Lemma 5.4.4 (Characterisation of RHKSs - 1)**.** *Let H be a Hilbert space which satisfies both points of the Moore-Aronszajn theorem, then $H = \overline{\text{span}\{K(x,\cdot);\ x \in X\}}$.*

Proposition 5.4.5 (Characterisation of RHKSs - 2)**.** *Let H be an Hilbert space, then H is a RKHS if and only if H is a Hilbert space where the pointwise evaluation is linear and bounded.*

The latter makes clear why RKHSs are so important: from the proposition above it stems that RKHSs contain only well-defined functions, which builds up a space in which the evaluation functional is linear and bounded. Recall that this was the assumption under which the Representer theorem held true. This is not the case for L^2, for example, because of the presence of classes of equivalence, which makes the evaluation functional not meaningful.
From the first characterisation of RKHS it becomes clear that the whole space is given by the span of kernel sections, which are the representers of the pointwise evaluation functionals. In order to understand why this is important, recall that if the assumption of the Representer theorem are satisfied, it has been proved that the representers of the linear and bounded functionals constitute the basis for the optimal estimate.

The structure of the representer and the kernel function must indeed be closely related, since the kernel shape the function space in which the optimisation problem takes place. In fact, by Riesz representation theorem, it must hold that:

$$S[h] = \langle h, s \rangle_H, \quad \forall h \in H. \tag{5.20}$$

This must be true even if the function $h(\cdot) = K(x, \cdot)$ is a kernel section, which acts as a point evaluation functional because of the reproducing property, giving:

- $S[K(x, \cdot)] = \langle K(x, \cdot), s \rangle_H, \quad \forall x \in X, \qquad$ (Representer theorem);
- $\langle K(x, \cdot), s \rangle = s(x), \quad \forall x \in X;$

which leads to

$$s(x) = S(K(x, \cdot)), \tag{5.21}$$

which highlights again the fact that kernel sections are the representers of the pointwise evaluation functionals. In light of the Representer theorem, the solution of the regression problem when the hypothesis space is a RKHS is then given by a linear combination of kernel sections.

5.5 Spectral characterisation of RKHS

In the following chapters, the RKHS framework will be used as the infinite-dimensional space on which the approximation of the Koopman operator will be learnt. In particular, the connection between these two worlds is given in Chapter 6.
In order to establish a common background, a spectral characterisation of RKHS is discussed in this section, as done in Chapter 2 for the Koopman operator.
The following definitions may be not standard, so they are recalled for completeness.

Definition 5.5.1 (Compact operator)**.** Let $\mathcal{O} : \Psi \to \Psi$ be an operator acting on Ψ, the latter being a generic metric space. $\mathcal{O}$ is said to be compact if it maps every limited subset of Ψ in a pre-compact subset of Ψ.

Definition 5.5.2 (Self-adjoint operator)**.** Let $\mathcal{O} : H \to H$ be a linear operator from H to H, with the domain given by a generic Hermitian space. The operator $\mathcal{O}$ is said to be self-adjoint if $\langle \mathcal{O}[h], \tilde{h} \rangle_H = \langle h, \mathcal{O}\left[\tilde{h}\right] \rangle_H$, $\forall h, \tilde{h} \in H$.

Definition 5.5.3 (Positive definite operator)**.** Let $\mathcal{O} : H \to H$ be a self-adjoint operator. $\mathcal{O}$ is said to be positive definite if $\langle \mathcal{O}[h], h \rangle_H \geq 0$, $\forall h \in H$. $\mathcal{O}$ is instead said to be strictly positive definite if $\langle \mathcal{O}[h], h \rangle_H > 0$, $\forall h \in H$.

The RKHS is a complete metric space, and not an operator, therefore it does not make sense to ask for its spectral decomposition. However it is possible to better understand some properties of this space by defining it through the eigenfunctions of a particular operator. The latter will indeed be the main goal of this section.
The following result is then useful, since it allows to characterise a generic Hilbert space H, which is invariant under the action of a compact operator $\mathcal{O}$, through the eigenfunctions of the same operator $\mathcal{O}$.

Theorem 5.5.4 (Spectral theorem for compact operators)**.** *Let $\mathcal{O} : H \to H$ be a linear and compact operator defined over a generic Hilbert space H. Then there exists in H a complete orthonormal basis $\phi_1, \phi_2, \dots$ given by eigenfunctions of $\mathcal{O}$.*

The theorem above holds for a generic Hilbert space and cannot be generalised further. In order to state the theorem above it is indeed required to deal with inner products, to define orthogonality, therefore it cannot apply to Banach spaces.
In particular, this result make it easy to deal with linear and compact operators because many properties can be rewritten in terms of the eigenvalues, as done in what follows. For instance, the Hilbert-Schmidt norm of a linear and compact operator is then trivially characterised by the sum of the eigenvalues. Indeed the Hilbert-Schmidt norm is the sum of the norms of the functions resulting from the application of the operator to each basis function, which trivially given by the sum of the eigenvalues, if the basis is the one formed by eigenfunctions. If the latter is finite, then the operator is a Hilbert-Schmidt operator (Axler et al., 1999). This does not hold for the kernel integral operator, which will be defined shortly.
The definition of the following operator will be of utmost importance for what follows, since the Reproducing Kernel Spaces will be characterised through its eigenfunctions.

Definition 5.5.5 (Kernel integral operator)**.** Let X be compact, μ be a non-degenerate Borel measure, and $K : X \times X \to \mathbb{R}$ a generic function. Define the kernel integral operator as:

$$\mathcal{S}_H : H \to H, \qquad \mathcal{S}\left[h\left(\cdot\right)\right] = \int_X K\left(x, \cdot\right) h\left(x\right) \mathrm{d}\mu\left(x\right) \tag{5.22}$$

$\forall h \in H$, where H is a generic Hilbert space.

The latter operator is defined through a generic function of two arguments, which might not even be a kernel. The properties of the operator clearly depend on the characteristics of the function that was used to define it. In the case of a continuous function, these are addressed by the following theorem (Einsiedler and Ward, 2017).

Theorem 5.5.6 (Kernel integral operator over continuous functions)**.** *Let $L^2_\mu\left(X\right)$ be a Hilbert space defined over the compact domain X, and let $K : X \times X \to \mathbb{R}$ be a general*

continuous function (not necessarily a kernel). The operator $\mathcal{S} : L^2_\mu(X) \to L^2_\mu(X)$ *defined in* (5.22) *is continuous, linear and compact.*

The next theorem instead governs the particular case in which K is actually a Kernel. Since the hypothesis of continuity is already there, the latter will be a Mercer kernel.

Theorem 5.5.7 (Kernel integral operator over Mercer kernel)**.** *Let* $L^2_\mu(X)$ *be a Hilbert space defined over the compact domain* X*, and let* $K : X \times X \to \mathbb{R}$ *be a Mercer kernel. The operator* $\mathcal{S} : L^2_\mu(X) \to L^2_\mu(X)$ *defined in* (5.22) *is continuous, linear and compact, also self-adjoint and positive definite.*

This reveals that the Mercer assumption is not restrictive, but the assumption of using a particular space such as the RKHS restrict also the operator $\mathcal{S}$ to be self-adjoint. Theorem 5.5.6 states that the operator $\mathcal{S}$ is compact, which has already been understood to be a remarkable property. In particular it is important in this case because it makes easier the spectral decomposition of the operator. In fact, the following theorem holds:

Theorem 5.5.8 (Spectrum of a compact operator)**.** *Let* Ψ *be a Banach space and* $\mathcal{O}$ *be a compact operator acting on* Ψ*. Then:*

- *Every nonzero* $\lambda \in \mathrm{Sp}(\mathcal{O})$ *is an eigenvalue of* $\mathcal{O}$*.*
- *For all nonzero* $\lambda \in \mathrm{Sp}(\mathcal{O})$*, there exist* n *such that* $\ker(\mathcal{O} - \lambda\mathcal{I})^n = \ker(\mathcal{O} - \lambda\mathcal{I})^{n+1}$*, and this subspace is finite-dimensional.*
- *The eigenvalues can only accumulate at* 0*. If the dimension of* Ψ *is not finite, then* $\mathrm{Sp}(\mathcal{O})$ *must contain* 0*.*
- $\mathrm{Sp}(\mathcal{O})$ *is at most countably infinite.*
- *Every nonzero* $\lambda \in \mathrm{Sp}(\mathcal{O})$ *is a pole of the resolvent function, which maps* $\lambda \mapsto (\mathcal{O} - \lambda\mathcal{I})^{-1}$*, and it is defined for* $\lambda \in \mathrm{Res}(\mathcal{O})$*.*

In particular the first proposition reveals that $\mathrm{Sp_c}(\mathcal{O}) = \mathrm{Sp_r}(\mathcal{O}) = \emptyset$, and the fourth one that $\mathrm{Sp}(\mathcal{O}) = \mathrm{Sp_p}(\mathcal{O})$ is at most countably infinite.
As a consequence, it turns out that the Kernel integral operator in Definition 5.5.5 can be diagonalised to yield a set of eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ so that $\lim_{i\to\infty} \lambda_i = 0$. Recall that if S is self-adjoint, then the eigenvalues are real, and if S is positive-definite, then $\lambda_i \geq 0,\ \forall i \in \mathbb{N}$.
From Theorem 5.5.4 it was already clear that eigenfunctions of compact operators form a basis for the Hilbert space on which the operator acts, but the latter result says also that

there are at most countably infinite non-zero eigenvalues, so that there is a subset of at most countably infinite eigenfunctions which are relevant to characterise the operator.
The previous section presented the Moore-Aronszajn theorem (Theorem 5.4.3), which formalises the correspondence between the kernel function $K(\cdot,\cdot)$ and the associated RKHS, which is then uniquely characterised. Also the kernel integral operator is defined on top of the kernel function, so that it too is uniquely characterised. Moreover, under the hypothesis of Theorem 5.5.7, the latter is guaranteed to be compact, meaning that it provides a basis for the Hilbert space, given by its eigenfunctions.
These eigenfunction can be used to characterise the kernel function itself, and therefore its associated Reproducing Kernel Hilbert Space. That is how a spectral characterisation of a complete vector space will be obtained.
The description of the Kernel through the eigenfunction of the kernel integral operator is given by the *Mercer* theorem.

Theorem 5.5.9 (Mercer)**.** *Let $K : X \times X \to \mathbb{R}$ be a Mercer kernel with X compact. Let $\{\lambda_i\}_{i=0}^{\infty}$, $\{\varphi_{\lambda_i}\}_{i=0}^{\infty}$ be the eigenvalues and eigenfunctions of the kernel integral operator defined in (5.22). Then:*

$$K(x,\tilde{x}) = \sum_{i=1}^{\infty} \lambda_i \varphi_{\lambda_i}(x)\, \varphi_{\lambda_i}(\tilde{x}) \tag{5.23}$$

where convergence is uniform in $X \times X$.

From Theorem 5.5.4 it descends that the set of eigenfunctions $\{\varphi_{\lambda_i}\}_{i=0}^{\infty}$ is an orthonormal basis for $L_\mu^2(X)$, given that $\mathcal{S}$ is compact. The following theorem characterises the RKHS $H \subset L_\mu^2(X)$ with the same eigenfunctions, proving that they are a basis also for H (Fukumizu, Song, and Gretton, 2013).

Theorem 5.5.10 (Spectral decomposition of RKHSs)**.** *If (5.23) holds, the associated RKHS can be expressed as:*

$$H = \left\{ h = \sum_{i=1}^{\infty} c_i \varphi_{\lambda_i} \mid \sum_{i=1}^{\infty} \frac{c_i^2}{\lambda_i} < \infty \right\} \tag{5.24}$$

and the inner product between $h = \sum_{i=1}^{\infty} \alpha_i \varphi_{\lambda_i}$ and $\tilde{h} = \sum_{i=1}^{\infty} \beta_i \varphi_{\lambda_i}$ is:

$$\langle h, \tilde{h} \rangle_H = \sum_{i=1}^{\infty} \frac{\alpha_i \beta_i}{\lambda_i} \tag{5.25}$$

The latter result makes it clear how the choice of the kernel function completely characterises the Reproducing Kernel Hilbert space through its spectral decomposition. Also, it is evident that $L^2_\mu(X)$ and H are basically the same space, though H has a constrained representation. The convergence requirement can indeed be seen as a constraint on the regularity of the function, imposed by the kernel structure. The coefficients c_i must indeed go to zero fast enough so that:

$$\|h\|_H = \sum_{i=1}^{\infty} \frac{c_i^2}{\lambda_i} < \infty, \tag{5.26}$$

for h to be in H.
Also, note that there is no dependency on the measure μ: even if the basis change - i.e., φ_{λ_i} change - their span and the induced norm remain the same, so that H is the same.
From Theorem 5.5.10 it is clear how the functions belonging to the Hilbert space H, are implicitly modelled using n basis function, with n possibly infinite. The great benefit of considering a RKHS as the hypothesis space, is that there is no need to specify the whole set of basis function - which would be infinite - but just the kernel which characterises the space. The latter can be chosen in such a way that some prior knowledge is embedded in the problem, such as smoothness of the regression function, and the hypothesis space is shaped accordingly. The explicit representation in terms of the basis function may also be unavailable, as it happens for some well-known kernels, e.g. *Matern* kernels, which are nonetheless largely employed.

5.6 RKHS and Gaussian Processes

With a deeper understanding of the RKHS framework, it is possible to derive a closed-form solution for the problem in (5.8) in the original case of an infinite-dimensional hypothesis space, which motivated the introduction of the whole framework.
Consider then the infinite-dimensional extension of problem (5.8), given by:

$$\hat{h} = \underset{h \in H}{\arg\min} \left\{ \overline{\sigma}^{-2} \sum_{i=1}^{M} \left(\overline{y}_i - h\left(\overline{x}_i\right) \right) + \eta \|h\|_H^2 \right\} \tag{5.27}$$

where $\Sigma_M = \overline{\sigma}^2 I$ for simplicity. Note that the regularised regression problem in (5.27) is a particular case of the one in (5.4) where $S_i[h] = h\left(\overline{x}_i\right)$, i.e., the S_is correspond to the evaluation functionals associated with the input location $\overline{x}_i$; while LP realises the fitting term through the corresponding $\overline{y}_i$ and weighs the penalty on the norm with η. From Proposition 5.4.5 it is clear that those functionals are linear and bounded, so the

problem above satisfies the conditions of the Representer theorem.
In order to explicitly compute the solution for the presented regression problem, the Bayesian perspective will be taken into account.
Recall that the penalty for the regularised regression is given by the squared norm of the candidate function in the Hilbert space H. In the case of RKHS, the latter can be computed as:

$$\langle h, h\rangle_H = \langle K(x,\cdot), K(x,\cdot)\rangle = K(x,x) \tag{5.28}$$

thanks to the characterisation in 5.4.4 and to the reproducing property in 5.4.3.
Recall that the latter, in the finite-dimensional setting, corresponded to the prior variance of the regressor.
Given that the regression variable is now a function h, a prior distribution over functions should be considered, which is taken as a Gaussian process, with zero mean for simplicity. The latter induces the following Gaussian distribution over the input points $\overline{x}$,

$$\hat{h}(\overline{x}) \sim \mathcal{N}(0, K(\overline{x},\overline{x})), \tag{5.29}$$

which is the prior distribution considered.
By generalising the measurement model in (5.17) for the infinite-dimensional case, the observations of the true function are given by noisy versions of samples from $\overline{x}$, as:

$$\overline{y} = h(\overline{x}) + w, \tag{5.30}$$

with $w \sim \mathcal{N}(0, \overline{\sigma}^2 I)$, i.e., the measurement noise which for simplicity is considered to be independently and equally distributed for every sample. Given that the noise is independent from the prior distribution of samples in (5.29), the induced prior distribution on the output locations $\overline{y}$ trivially results in:

$$\overline{y} \sim \mathcal{N}\left(0, K(\overline{x},\overline{x}) + \overline{\sigma}^2 I\right). \tag{5.31}$$

Consider then the joint distribution of output locations $\overline{y}$ and the output of the candidate function at a generic test location x; the latter is again a Gaussian distribution and in particular it is given by:

$$\begin{bmatrix} \overline{y} \\ \hat{h}(x) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\overline{x},\overline{x}) + \overline{\sigma}^2 I & K(\overline{x},x) \\ K(x,\overline{x}) & K(x,x) \end{bmatrix}\right). \tag{5.32}$$

Finally, the conditional distribution given the available data and the test location yields the predictive equations for Gaussian process regression, as:

$$\hat{h}(x) \mid \overline{x}, \overline{y}, x \sim \mathcal{N}\left(m_{\hat{h}}, \Sigma_{\hat{h}}\right), \tag{5.33}$$

where

$$m_{\hat{h}}(x) := K(x, \overline{x}) \left[K(\overline{x}, \overline{x}) + \overline{\sigma}^2 I\right]^{-1} \overline{y} \tag{5.34}$$

$$\Sigma_{\hat{h}}(x) := K(x, x) - K(x, \overline{x}) \left[K(\overline{x}, \overline{x}) + \overline{\sigma}^2 I\right]^{-1} K(\overline{x}, x). \tag{5.35}$$

The maximum a posteriori corresponds to the mean of the conditional distribution in the Gaussian case, and indeed it can be seen as a generalisation of the formula in (5.16). Note that, in accordance with the Representer theorem (Theorem 5.2.7), the estimate is given by a finite combination of the representers of the evaluation functional corresponding to the input locations, which have been understood to be the kernel sections $K(\cdot, \overline{x}_i)$, $i = 1, \ldots, M$.

6

Learning Koopman operators in RKHSs

This chapter lays the foundation for subsequent developments. In particular the Koopman operator framework and the estimation in Reproducing Kernel Hilbert Spaces are finally merged.
This is done by considering the two most popular methods for learning a finite-dimensional Koopman operator and analysing them under a Bayesian perspective. The latter process allows to understand the connections between the Koopman operator framework and kernel methods. Moreover, an interesting dual perspective between DMD and EDMD is formalised.
The proposed approach allows to smoothly introduce regularisation in the Koopman estimation and to meaningfully select the space on which the regression should be performed through the choice of the kernel.
The chapter is mainly based on the work by **Zanini, Francesco** and Chiuso, 2021b, and thus it represents an original contribution.

6.1 Kernel sections as dictionary of observables

As presented in Chapter 4, different approaches have been developed over the past years in order to approximate the operator from data. In particular, the estimation of Koopman modes have become an active area of research for the analysis of nonlinear flows, especially in the fluid dynamics community (Bagheri, 2013; Budišić, Mohr, and Mezić, 2012; Chen, Tu, and Rowley, 2012; Hemati, Williams, and Rowley, 2014; Mezić, 2013; Rowley et al., 2009; Taira et al., 2017). These advances all rely on the linearity of the Koopman description, which allows for a very simple characterisation of the evolution of the temporal dynamics, thanks to the eigendecomposition (see Chapter 2, Section 2.3). The Koopman modes are indeed invariant with respect to the propagation over time, which is then described by taking powers of the eigenvalues. As it now clear,

this description has the drawback of being infinite-dimensional: the Koopman operator propagates *observables*, which lie in a function space.
Therefore, any approach which aims to discover finite-dimensional approximation of the latter operator, deals with only a particular finite subset of the infinite-dimensional space of functions. This can be achieved implicitly, as done by Dynamic Mode Decomposition method (Tu et al., 2014), which performs its operation on the space spanned by linear functions of the state; or explicitly, following the Extended Dynamic Mode Decomposition procedure (Williams, Kevrekidis, and Rowley, 2015), which prescribes to select a dictionary of functions, in order to define in advance the invariant subspace for the Koopman operator. The two algorithms have been introduced in Chapter 4.
Also DMD can be generalised to a richer set of basis functions, as it is explicitly stated in Mauroy and Gonçalves, 2020, although loosing the computational advantages. The latter formulation though makes it clear that the two approaches have dual properties, which is in fact partially investigated in that work.
In particular, since the pseudo-inverse of a rectangular matrix is involved, they are well-defined for different regimes. The EDMD method requires the number of samples to be larger than the number of basis functions, in order for the procedure to have a unique solution; the DMD approach instead works under the assumption that the number of basis function is greater than the number of snapshots from the system. The latter issue is explained in Chapter 4, Section 4.4, while a step towards a better understanding of this duality is presented in Subsection 6.3.
This chapter discusses how the Koopman operator can be learned in Reproducing Kernel Hilbert Spaces, and what are the benefits of framing the problem in this setting. As explained in Chapter 5, RKHSs are Hilbert spaces which are uniquely characterised by a kernel function $K(\cdot, \cdot)$, whose basis of functions may not even yield an explicit formulation in the feature space. This is in contrast with the EDMD approach, which expressly requires to specify the observables generating the subspace.
In the proposed approach, it is the choice of the kernel function that determines the space, solving the problem of having to choose the dictionary of functions a priori. Of course it is still required to select the Kernel function, however nowadays there is a rather clear understanding about the prior information that many common kernels induce in the regression problem, so that a meaningful choice can be made, with respect to the system under study (Chen, 2018; Carli, Chen, and Ljung, 2017; Chen et al., 2014; Chen et al., 2016; Chiuso et al., 2014; Dinuzzo, 2015; Chen and Ljung, 2015; Marconato, Schoukens, and Schoukens, 2016; Pillonetto, Chiuso, and Nicolao, 2011). Moreover, the solution of the regression problem in RKHS lies in the span of linear combinations of kernel

sections, centred on available datapoints, as prescribed by the Representer theorem (see Chapter 5). This means that the number of kernel functions building the invariant space for the Koopman estimate is implicitly selected to be always equal to the number of snapshots from the system; making the computational complexity only dependent on the number of available observations, and not on the space of functions.

6.2 Koopman & Kernels

Given the two important reasons above, RKHSs seem a natural choice for the subspace of functions representing the invariant space for the approximation of the Koopman operator. However, so far, the literature has mainly focused on the analysis of the spectrum of the Koopman operator through kernel methods, without investigating an explicit formulation of the operator with respect to the Kernel sections, or bridging the Bayesian framework with the composition operator view.

In Williams, Rowley, and Kevrekidis, 2015 there is one of the first mentions of kernel methods applied to the Koopman operator framework. In the latter paper, the problem of computational complexity increasing with the number of basis functions is addressed. In particular, a matrix approximation of the Koopman operator is proposed, which is very similar to EDMD except that some algebraic manipulations are performed in order to "hide" within matrix products the dimension corresponding to the space of functions. This is indeed the principle behind the *kernel trick* (Theodoridis and Koutroumbas, 2009; Shalev-Shwartz and Ben-David, 2014), so that the space of functions can grow without burdening the computational complexity. The new matrix is shown to have the same eigenvalues as the EDMD approach, and the eigenvectors can be obtained by a linear combination of EDMD eigenvectors. This result in the same formulation as given later in (6.61), when the regularisation term is neglected.

A very similar approach is taken by Kawahara, 2016, resulting however in a variant of Dynamic Mode Decomposition. In this work it is explicitly stated that the search space is given by a RKHS, in order to approximate the Koopman eigenfunctions with a richer basis of functions. However, by relying on the Krylov subspace for regression, the method they propose only works under the assumption of sequential data, i.e., for a stream of observations coming from a long trajectory, which is a less general case than considering general snapshots of the system.

In Takeishi, 2019 and Kurebayashi, Shirasaka, and Nakao, 2016, the problem of learning kernel hyper-parameters is addressed, with the aim to obtain better prediction for the Koopman operator. The former method relies on the *Galerkin* approximation (Giannakis,

2017; Das and Giannakis, 2019), and thus again it works only for data collected along a single trajectory. In this paper a particular kernel is defined, whose implicit distance measure takes into account the evolution over time of observed points. The Koopman operator commutes with the integral operator defined upon that kernel function, when the number of "propagated points" goes to infinity (Das and Giannakis, 2019). By relying on the latter fact, the main idea of the paper is to minimise the norm of the commutator of the Koopman operator and the integral operator thus defined. The commutator however cannot be evaluated directly, so a bound on the norm is minimised instead. The latter paper instead relies on the aforementioned kernel version of the EDMD approach (Williams, Rowley, and Kevrekidis, 2015), but utilises a separate set of datapoints - which are then not used to approximate the Koopman operator - in order to minimise the Mean Squared Error over the kernel hyper-parameters, as it is usually done in cross-validation techniques.
Another related paper is the one by Das and Giannakis, 2020, whose aim is to identify Koopman eigenfrequencies and eigenfunctions, from data collected over a long trajectory. The main result of the paper are necessary and sufficient conditions for a Fourier function, defined on samples along an orbit of the dynamics, to be extensible to a Koopman eigenfunction over the whole state space, lying in a Reproducing Kernel Hilbert Space.
In Klus, Schuster, and Muandet, 2020 the eigendecomposition of transfer operator is analysed, with reference to Reproducing Kernel Hilbert Spaces. Similarly to what will be presented in later sections, an explicit formulation of the Koopman operator is derived, however they specifically link their approach with *conditional mean embeddings* and *covariance operators*, relying on a uniform sampling measure to match the framework in which only a finite amount of data are available. Also their approach seems less intuitive as, for instance, the introduction of the regularisation is there to overcome the otherwise necessary technical assumption that a version of the conditional expectation is included in the RKHS as a function of the input locations (Fukumizu, Song, and Gretton, 2013). The presentation in this chapter instead exploits the well-known Representer theorem (Schölkopf, Herbrich, and Smola, 2001) to address the regression in RKHS, and the regularisation parameter is introduced following the Bayesian interpretation (Rasmussen and Williams, 2006).

6.3 A dual view of EDMD and DMD

The two most popular approaches to estimate a finite-dimensional approximation of the Koopman operator from data, reviewed in Chapter 4, Section 4.4, can be given a

dual interpretation. The proposed unifying framework will reveal that in the case of an equal number of basis functions and snapshots of the system, the two approaches are substantially equivalent. Although some slight differences in the formulation will still be present, since EDMD acts in the space of coefficients for a fixed basis while DMD operates in the space of function values, the interpretation is actually the same. The latter is an important remark because in the RKHS framework that will be presented later, this is always the case. Indeed, with kernel methods, the solution of the regression problem is known to lie on a subspace of the associated Hilbert space, given by the finite-dimensional span of the kernel section which are centred in the available points. This means that the number of basis functions for the space is the same as the number of datapoints, and the estimate belongs to that space. The dual view for the kernel version of the Koopman operator is presented in Section 6.6.

As already explained in Chapter 4, both approaches rely on a fixed dictionary of observables, $\{\psi_j : X \to \mathbb{R}\}_{j=1}^{N}$, and on observations coming from the system, i.e., d-dimensional snapshots of the discrete-time flow describing the dynamics of interest, given as $\{\overline{x}_i, \overline{y}_i\}_{i=1}^{M}$, where $\overline{y}_i = f(\overline{x}_i)$. The vector containing the whole set of datapoints is denoted for the input and output locations respectively as $\overline{x} = \begin{bmatrix}\overline{x}_1 \dots \overline{x}_M\end{bmatrix}$ and $\overline{y} = \begin{bmatrix}\overline{y}_1 \dots \overline{y}_M\end{bmatrix}$.

Given then a dictionary of functions $D = \{\psi_j\}_{j=1}^{N}$, define as usual the space

$$\Psi_D = \operatorname{span}\{D\} = \operatorname{span}\{\psi_1, \dots, \psi_N\} \tag{6.1}$$

as the finite-dimensional space of linear combinations of the functions provided by the dictionary. The EDMD approach prescribes to minimise the residual $\mathfrak{r}(\cdot)$ between the actual Koopman operator and its projection on Ψ_D, for all the basis functions ψ_j. If the space Ψ_D was invariant with respect to the Koopman operator, then the residual would be zero. But since the fixed dictionary of function in general cannot describe exactly the behaviour of the Koopman operator (unlike in Example 2.1.1), the image of the operator is a superset of Ψ_D, i.e. $\exists j \in [1, N] \mid \mathcal{U}[\psi_j] \notin \Psi_D$ so that, with some abuse of notation, $\operatorname{Im}[\mathcal{U}[\Psi_D]] \supsetneq \Psi_D$.

The latter is analogously expressed as:

$$\mathcal{U}[\psi](\cdot) = \mathfrak{P}_{\Psi_D}[\psi](\cdot) + \mathfrak{r}(\cdot). \tag{6.2}$$

$\forall \psi \in \Psi_D$, where $\mathfrak{P}_{\Psi_D} : \Psi \to \Psi_D$ is the projection operator going from the whole space of observables onto the linear space spanned by the dictionary, and $\mathfrak{r}(\cdot)$ denotes the residual.

The objective is to minimise $\mathfrak{r}(\cdot)$ relying on the available observations $\{\overline{x}_i, \overline{y}_i\}_{i=1}^{M}$. There-

fore equation (6.2) is specialised for the observed snapshots pair as

$$\mathcal{U}\left[\psi\right]\left(\overline{x}\right) = \mathfrak{P}_{\Psi_D}\left[\psi\right]\left(\overline{x}\right) + \mathfrak{r}\left(\overline{x}\right), \tag{6.3}$$

so that

$$\begin{aligned}\mathfrak{r}\left(\overline{x}\right) &= \mathcal{U}\left[\psi\right]\left(\overline{x}\right) + \mathfrak{P}_{\Psi_D}\left[\psi\right]\left(\overline{x}\right) &(6.4)\\ &= \psi\left(f\left(\overline{x}\right)\right) + \mathfrak{P}_{\Psi_D}\left[\psi\right]\left(\overline{x}\right) &(6.5)\\ &= \psi\left(\overline{y}\right) + \mathfrak{P}_{\Psi_D}\left[\psi\right]\left(\overline{x}\right) &(6.6)\end{aligned}$$

Define the vector of basis functions composing the dictionary D as

$$D_\psi\left(\cdot\right) := \begin{bmatrix}\psi_1\left(\cdot\right) & \dots & \psi_N\left(\cdot\right)\end{bmatrix}, \tag{6.7}$$

with the N observables, as done in previous chapters. The matrix of evaluations on the available datapoints for all the function in the dictionary are again

$$\begin{aligned}D_{\overline{x}} &= \begin{bmatrix}\psi_1\left(\overline{x}_1\right) & \psi_2\left(\overline{x}_1\right) & \dots & \psi_N\left(\overline{x}_1\right)\\ \psi_1\left(\overline{x}_2\right) & \psi_2\left(\overline{x}_2\right) & \dots & \psi_N\left(\overline{x}_2\right)\\ \vdots & \vdots & \ddots & \vdots\\ \psi_1\left(\overline{x}_M\right) & \psi_2\left(\overline{x}_M\right) & \dots & \psi_N\left(\overline{x}_M\right)\end{bmatrix},\\ D_{\overline{y}} &= \begin{bmatrix}\psi_1\left(\overline{y}_1\right) & \psi_2\left(\overline{y}_1\right) & \dots & \psi_N\left(\overline{y}_1\right)\\ \psi_1\left(\overline{y}_2\right) & \psi_2\left(\overline{y}_2\right) & \dots & \psi_N\left(\overline{y}_2\right)\\ \vdots & \vdots & \ddots & \vdots\\ \psi_1\left(\overline{y}_M\right) & \psi_2\left(\overline{y}_M\right) & \dots & \psi_N\left(\overline{y}_M\right)\end{bmatrix},\end{aligned} \tag{6.8}$$

with $D_{\overline{x}}, D_{\overline{y}} \in \mathbb{R}^{M\times N}$, whose rows are in turn written as

$$D_{\overline{x}_i} = \begin{bmatrix}\psi_1\left(\overline{x}_i\right) & \dots & \psi_N\left(\overline{x}_i\right)\end{bmatrix}, \quad D_{\overline{y}_i} = \begin{bmatrix}\psi_1\left(\overline{y}_i\right) & \dots & \psi_N\left(\overline{y}_i\right)\end{bmatrix}. \tag{6.9}$$

The EDMD method requires to find N vectors of coefficients, grouped as columns of the matrix U, which give the best approximation of the composition $\psi_j \circ f$, using a linear combination of functions which are present in the dictionary, $\forall j \in [1, N]$. This means that the cumulative error between the observation and the approximation within Ψ_D is minimised, for every snapshot pair and every basis function. Denoting as $U^{(*j)}$ the j-th

column of the matrix U and as $U^{(i*)}$ the i-th row, the error is given by:

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\psi_j\left(\overline{y}_i\right)-\sum_{k=1}^{N}\psi_k\left(\overline{x}_i\right)U^{(jk)}\right]^2 \tag{6.10}$$

$$=\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\psi_j\left(\overline{y}_i\right)-D_{\overline{x}_i}U^{(*j)}\right]^2 \tag{6.11}$$

$$=\sum_{i=1}^{M}\left\|D_{\overline{y}_i}-D_{\overline{x}_i}U\right\|^2 \tag{6.12}$$

$$=\left\|D_{\overline{y}}-D_{\overline{x}}U\right\|_F^2. \tag{6.13}$$

The matrix U minimising (6.13) is the finite-dimensional approximation of the Koopman operator:

$$U_{\text{EDMD}}=\min_{U\in\mathbb{R}^{N\times N}}\left\|D_{\overline{y}}-D_{\overline{x}}U\right\|_F^2, \tag{6.14}$$

whose solution is

$$U_{\text{EDMD}}=D_{\overline{x}}^{\dagger}D_{\overline{y}}. \tag{6.15}$$

Recall that any observable $\psi\in\Psi_D$ is uniquely determined by a vector coefficient α multiplying the functions of the dictionary D. It holds:

$$\psi\left(\cdot\right)=\sum_{i=1}^{N}\alpha_i\psi_i\left(\cdot\right) \tag{6.16}$$

$$=D_{\psi}\alpha, \tag{6.17}$$

$\forall\psi\in\Psi_D$.
Then for any α characterising ψ the EDMD approach finds β describing another function in Ψ_D which best approximate the action of the infinite-dimensional Koopman operator on ψ. The action of the approximate operator is given by

$$\beta=U_{\text{EDMD}}\alpha, \tag{6.18}$$

which maps the coefficients of the observable ψ into the coefficients β representing $\psi\circ f$. The latter is exactly the operator acting on coefficients that has been derived in Chapter 4, Section 4.3.
Note that the problem is meaningful if $N\leq M$, i.e., when the number of functions in the dictionary are less than the training pairs, so that it is not possible to fit perfectly all the observations with the given functions. This implies that $D_{\overline{x}}$ is an injective operator, hence the left inverse exists, and it is given by $D_{\overline{x}}^{\dagger}$.

The DMD approach can be formulated in a dual fashion, where the sought matrix is a mapping from inputs to outputs, with respect to the values of the observables obtained by evaluating the functions of the dictionary at the available points. That is, the problem is to minimise the quantity:

$$\|\psi_i(\overline{y}) - U\psi_i(\overline{x})\|_F^2, \tag{6.19}$$

$\forall i \in [1, N]$, with respect to the matrix $U \in \mathbb{R}^{M\times M}$.
This yield the dual problem as follows:

$$U_{\text{DMD}} = \min_{U\in\mathbb{R}^{M\times M}} \|D_{\overline{y}} - UD_{\overline{x}}\|_F^2, \tag{6.20}$$

whose solution (Mauroy and Gonçalves, 2020) is

$$U_{\text{DMD}} = D_{\overline{y}} D_{\overline{x}}^{\dagger}; \tag{6.21}$$

which instead makes sense for $M \leq N$, so if the number of functions are higher than training points. The latter condition indeed makes $D_{\overline{x}}$ a surjective operator, and $D_{\overline{x}}^{\dagger}$ is its right inverse.
The duality of these two approaches can be appreciated by comparing (6.14), (6.15) and (6.20), (6.21). Moreover, in the particular case of $N = M$, the pseudo-inverse becomes the same, because of the uniqueness of the Moore-Penrose inverse.
So far the snapshots of the system are assumed to be noiseless pairs coming from the true transition function that should be estimated. In the next sections the system dynamics are instead assumed to be noisy.
This is considered in order to tackle a larger class of problems: the case of stochastic dynamics has already proved itself important in Chapter 3, and many system have inherent stochasticity. The latter can for instance be given by the measurement process. On the other hand, the noise could be an artifact to take into account undermodeling, when the postulated model class cannot describe the dynamics in full. Also, the estimated noise in the dynamics can be used as a regularisation parameter which is often employed to avoid fitting the disturbance and therefore to make the algorithm more stable.
In what follows, the measurements will be assumed to satisfy the following relation:

$$x_{k+1} = f(x_k) + w_k \tag{6.22}$$

where w_k is a general discrete-time noise process.
Note that for the task of one-step prediction, the one addressed in this chapter, the assumption of full measurability of the state is not needed. Indeed, as with kernel

methods, the same procedure is valid even if the observation do not correspond to the full state, but instead represent an output that could lie in a different space. However, as will be clear in the next chapter, the proposed procedure will be iterated to obtain subsequent predictions of the observable and therefore address the control problem. Predictions for steps arbitrarily far away in the future will be necessary, and in order to get this, the measurability assumption is required, to set the regression correctly. This is equivalent to impose that the state-transition function is an endomorphism.

6.4 From Koopman to kernels

The EDMD principle relies on a fixed dictionary of functions to capture the dynamics and understand the evolution of the observables of the system that are of interest. The approximation of the Koopman operator is then N-dimensional, with N the number of functions in the dictionary, and Ψ_D is the invariant subspace which keeps confined the action of the learned operator U_{EDMD}. As explained in the previous section and in Chapter 4, the Extended Dynamic Mode Decomposition learn an approximation of the Koopman operator acting on the space of coefficients, which uniquely characterise any function in the linear span of the given basis. This is accomplished as:

$$\beta = U_{\text{EDMD}}\alpha = D_{\overline{x}}^{\dagger} D_{\overline{y}}\alpha \tag{6.23}$$

where α are the coefficient of the generic desired function ψ, belonging to Ψ_D, and β are the coefficients which yield the best representation in Ψ_D for the composition $\psi \circ f$.
However, since the dynamics in (6.22) are affected by the noise w_k, there is no exact correspondence between the evaluations of the observable in the available snapshot pairs, i.e., $\psi(\overline{y}) \neq \psi(f(\overline{x}))$.
In what follows, this mismatch is modeled as a static perturbation ε, so that:

$$\psi(\overline{y}) = \psi(f(\overline{x})) + \varepsilon. \tag{6.24}$$

As prescribed by the EDMD approach, the objective is to find β so that the Koopman operator is well approximated in Ψ_D, i.e., $\psi(f(\overline{x})) = D_{\overline{x}}\beta$. Then the chosen β should satisfy

$$\psi(\overline{y}) = D_{\overline{x}}\beta + \varepsilon, \tag{6.25}$$

for $\beta \in \mathbb{R}^N$.
In order to make the problem easily tractable, the uncertainty ε is modeled as a zero-mean Gaussian random variable with variance $\mathbb{V}[\varepsilon] = \overline{\sigma}^2$. Also, by aiming for a Bayesian

perspective, a prior distribution is assumed for the regression coefficients as $\beta \sim \mathcal{N}(0, \Lambda)$, with $\Lambda \in \mathbb{R}^{N \times N}$.
Since the learned operator acts on Ψ_D, it only makes sense to choose ψ in that space, so that there exists α such that $\psi(\cdot) = D_\psi \alpha$. Specialising this relation for the observed outputs $\overline{y}$, it holds that $\psi(\overline{y}) = D_{\overline{y}}\alpha$. This gives rise to the following linear measurement model:

$$\psi(\overline{y}) = D_{\overline{y}}\alpha = D_{\overline{x}}\beta + \varepsilon. \tag{6.26}$$

For a linear model in the Bayesian setting, predictions are based on the posterior distribution of the coefficients β, computed using Bayes rule (Bayes, 1763).

Theorem 6.4.1 (Bayes)**.** *The conditional probability of the events E and $\tilde{E}$ belonging to a probability space can be expressed as:*

$$\mathbb{P}\Big(E \mid \tilde{E}\Big) = \frac{\mathbb{P}\Big(\tilde{E} \mid E\Big)\mathbb{P}(E)}{\mathbb{P}\Big(\tilde{E}\Big)} \tag{6.27}$$

which in the Bayesian setting can be interpreted as (Rasmussen and Williams, 2006)

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}. \tag{6.28}$$

In the considered case, the theorem is written as:

$$p(\beta \mid D_{\overline{y}}\alpha, D_{\overline{x}}) = \frac{p(D_{\overline{y}}\alpha \mid D_{\overline{x}}, \beta)\, p(\beta)}{p(D_{\overline{y}}\alpha \mid D_{\overline{x}})} \tag{6.29}$$

where the marginal likelihood is just a normalising constant, since it is independent of the regression coefficients. Indeed it is given by:

$$p(D_{\overline{y}}\alpha \mid D_{\overline{x}}) = \int p(D_{\overline{y}}\alpha \mid D_{\overline{x}}, \beta)\, p(\beta)\, \mathrm{d}\beta, \tag{6.30}$$

where it is clear that the regression coefficients are marginalised out.
By computing the terms in (6.29) for the Bayesian case, it is possible to retrieve the following expression for the probability of β, conditioned on the observations:

$$p(\beta \mid D_{\overline{y}}\alpha, D_{\overline{x}}) \propto \exp\left[-\frac{1}{2}\left(\beta - \hat{\beta}\right)^\top \left(\frac{D_{\overline{x}}^\top D_{\overline{x}}}{\sigma^2} + \Lambda^{-1}\right)\left(\beta - \hat{\beta}\right)\right], \tag{6.31}$$

where $\hat{\beta} = \left(D_{\overline{x}}^\top D_{\overline{x}} + \sigma^2\Lambda^{-1}\right)^{-1} D_{\overline{x}}^\top D_{\overline{y}}\alpha$. It is clear that the distribution in (6.31) is

Gaussian, so that the posterior has the following distribution:

$$p\left(\beta \mid D_{\overline{y}}\alpha, D_{\overline{x}}\right) = \mathcal{N}\left(\left(D_{\overline{x}}^{\top} D_{\overline{x}} + \sigma^2 \Lambda^{-1}\right)^{-1} D_{\overline{x}}^{\top} D_{\overline{y}}\alpha, \left(\frac{D_{\overline{x}}^{\top} D_{\overline{x}}}{\sigma^2} + \Lambda^{-1}\right)\right). \tag{6.32}$$

For any Gaussian posterior the mean coincides with the mode of the distribution (Rasmussen and Williams, 2006), so that $\hat{\beta}$ is called the *maximum a posteriori* (MAP) estimate of β.

Interestingly, the MAP estimate can be seen as a regularised version of EDMD. Indeed, by setting the regularisation parameter $\sigma^2 = 0$, the original formula from Williams, Kevrekidis, and Rowley, 2015 is recovered:

$$\hat{\beta} = \left(D_{\overline{x}}^{\top} D_{\overline{x}}\right)^{-1} D_{\overline{x}}^{\top} D_{\overline{y}}\alpha = D_{\overline{x}}^{\dagger} D_{\overline{y}}\alpha. \tag{6.33}$$

Therefore the *regularised EDMD* is defined to be the following:

$$U_{\mathrm{EDMD}}^{(\sigma^2)} = \left(D_{\overline{x}}^{\top} D_{\overline{x}} + \sigma^2 \Lambda^{-1}\right)^{-1} D_{\overline{x}}^{\top}. \tag{6.34}$$

The next proposition presents a different expression for the regularised EDMD operator in 6.34, which will be exploited in what follows.

Proposition 6.4.2. *The* regularised EDMD *can equivalently be rewritten as:*

$$U_{\mathrm{EDMD}}^{(\sigma^2)} = \Lambda D_{\overline{x}}^{\top} \left(D_{\overline{x}} \Lambda D_{\overline{x}}^{\top} + \sigma^2 I_N\right)^{-1} \tag{6.35}$$

Proof. By rearranging the terms in the following expression

$$\begin{aligned} D_{\overline{x}}^{\top} \left(D_{\overline{x}} \Lambda D_{\overline{x}}^{\top} + \sigma^2 I_N\right) &= D_{\overline{x}}^{\top} D_{\overline{x}} \Lambda D_{\overline{x}}^{\top} + D_{\overline{x}}^{\top} \sigma^2 && (6.36) \\ &= \left(D_{\overline{x}}^{\top} D_{\overline{x}} + \sigma^2 \Lambda^{-1}\right) \Lambda D_{\overline{x}}^{\top}, && (6.37) \end{aligned}$$

it is trivial to derive the identity

$$\left(D_{\overline{x}}^{\top} D_{\overline{x}} + \sigma^2 \Lambda^{-1}\right)^{-1} D_{\overline{x}}^{\top} \left(D_{\overline{x}} \Lambda D_{\overline{x}}^{\top} + \sigma^2 I_N\right) = \Lambda D_{\overline{x}}^{\top} \tag{6.38}$$

so that

$$\left(D_{\overline{x}}^{\top} D_{\overline{x}} + \sigma^2 \Lambda^{-1}\right)^{-1} D_{\overline{x}}^{\top} = \Lambda D_{\overline{x}}^{\top} \left(D_{\overline{x}} \Lambda D_{\overline{x}}^{\top} + \sigma^2 I_N\right)^{-1} \tag{6.39}$$

which proves the proposition. □

Given that the Extended Dynamic Mode Decomposition maps the coefficients for the observable in D into new coefficients approximating the composition in the same basis, the estimation for the action of the Koopman operator would be:

$$\widehat{\mathcal{U}[\psi]}(x) = D_\psi(x)\beta = D_\psi(x) U_{\text{EDMD}}^{(\sigma^2)}\alpha \tag{6.40}$$

$$= D_\psi(x)\Lambda D_{\overline{x}}^\top \left(D_{\overline{x}}\Lambda D_{\overline{x}}^\top + \sigma^2 I_N\right)^{-1} D_{\overline{y}}\alpha \tag{6.41}$$

$$= D_\psi(x)\Lambda D_{\overline{x}}^\top \left(D_{\overline{x}}\Lambda D_{\overline{x}}^\top + \sigma^2 I_N\right)^{-1} \psi(\overline{y}), \tag{6.42}$$

in which the last equation yields a clear connection with kernel methods. Indeed, by defining a suitable finite-dimensional kernel, it is possible to understand (6.42) as the estimate given by the Representer theorem. The latter kernel is the one associated with the dictionary of functions D. In order to see this, define the kernel as inner product of basis function, through:

$$K(x,y) := D_\psi(x)\Lambda D_\psi(y)^\top = \langle D_\psi(x), D_\psi(y)\rangle_\Lambda; \tag{6.43}$$

then, the estimate provided by the regularised EDMD can be rewritten in terms of kernel sections as:

$$\widehat{\mathcal{U}[\psi]}(x) = K(x,\overline{x})\left[K(\overline{x},\overline{x}) + \sigma^2 I_M\right]^{-1}\psi(\overline{y}). \tag{6.44}$$

It is possible to recognise in (6.44) the *a posteriori* Bayesian estimate of the composition function, given by $\psi_+(\cdot) := \psi(f(\cdot))$. Indeed, under the Gaussian prior

$$\psi_+(\cdot) \sim \mathcal{N}(0, K(\cdot,\cdot)), \tag{6.45}$$

and given the noisy measurements

$$\psi(\overline{y}) = \psi(f(\overline{x})) + \varepsilon, \tag{6.46}$$

the Gaussian posterior distribution is centred in (6.44), which is then the MAP estimate of $\psi_+(\cdot)$.

6.5 From kernels to Koopman

To ultimately draw connections with the RKHS framework, the reverse problem is addressed in this section, i.e., the reconstruction of the Koopman operator with a dictionary of functions composed of kernel sections. Exploiting again the Bayesian perspective, consider observables $\psi(\cdot)$ which are zero-mean Gaussian processes with

covariance function given by $K(x, \tilde{x}) = \mathbb{E}[\psi(x)\psi(\tilde{x})]$. This is equivalent to assume that Ψ_D is a Reproducing Kernel Hilbert Space characterised by the kernel $K(\cdot, \cdot)$.
The objective is the same, i.e., to reconstruct the composition function $\psi_+(\cdot)$, given the generic observable $\psi(\cdot) \in \Psi_D$, and noisy observations coming from the system, which are modeled through:

$$\psi(\overline{y}) = \psi(f(\overline{x})) + \varepsilon, \tag{6.47}$$

where $\varepsilon \sim \mathcal{N}(0, \overline{\sigma}^2)$ From these measurements, and by taking into account the space of functions given by the RKHS, an estimate of $\psi(f(\cdot)) = \psi_+(\cdot)$ should be provided.
Recall that under the Bayesian framework (see Chapter 5, Section 5.6) the MAP estimate is given by:

$$\hat{\psi}_+(\cdot) = K(\cdot, \overline{x})\left[K(\overline{x}, \overline{x}) + \sigma^2 I_M\right]^{-1}\psi(\overline{y}). \tag{6.48}$$

where σ^2 is a regularisation parameter corresponding to the estimated variance of the noise in the observations, which are the evaluation of the observables in the training points.
Under the assumption that $\psi \in \Psi_D$, it is clear that:

$$\psi(\cdot) = \sum_{i=1}^{M} K(\cdot, \overline{x}_i)\alpha_i, \tag{6.49}$$

so that

$$\psi(\overline{y}) = \sum_{i=1}^{M} K(\overline{y}, \overline{x}_i)\alpha_i. \tag{6.50}$$

This gives the coefficient of the considered observable ψ, with respect to the basis functions given by the kernel section centred in the input locations $\overline{x}$. The latter are needed to reconstruct the action of the EDMD approach. By rewriting equation (6.48) as

$$\hat{\psi}_+(\cdot) = K(\cdot, \overline{x})\left[K(\overline{x}, \overline{x}) + \sigma^2 I_M\right]^{-1}\sum_{i=1}^{M} K(\overline{y}, \overline{x}_i)\alpha_i \tag{6.51}$$

$$= K(\cdot, \overline{x})\left[K(\overline{x}, \overline{x}) + \sigma^2 I_M\right]^{-1} K(\overline{y}, \overline{x})\alpha \tag{6.52}$$

it is clear that the EDMD formulation with respect to the basis given by the kernel sections is given by:

$$U_{\text{EDMD}}^{(K)} = \left[K(\overline{x}, \overline{x}) + \sigma^2 I_M\right]^{-1} K(\overline{y}, \overline{x}). \tag{6.53}$$

Indeed by multiplying the above matrix with the coefficients α of the selected observable, the operator returns the coefficients $\beta = U_{\text{EDMD}}^{(K)}\alpha$ for the same kernel sections $K(\cdot, \overline{x})$,

which best approximate the composition function $\psi_+(\cdot)$.
This is formalised in the next proposition.

Proposition 6.5.1. *Given an observable $\psi \in \Psi_D$, the regularised estimate of the EDMD approach framed in a Reproducing Kernel Hilbert Space, defined as*

$$U_{\text{EDMD}}^{(K)} = \left[K(\overline{x},\overline{x}) + \sigma^2 I_M\right]^{-1} K(\overline{y},\overline{x}), \tag{6.54}$$

maps the coefficients α describing the observable in the space spanned by the kernel sections $K(\overline{x},\cdots)$, obtained as $\psi(\overline{y}) = K(\overline{y},\overline{x})\,\alpha$, to

$$\beta = U_{\text{EDMD}}^{(K)}\alpha, \tag{6.55}$$

which defines the Bayesian estimate of $\psi(f(\cdot))$ under the measurement model

$$\psi(\overline{y}) = \psi(f(\overline{x})) + \varepsilon. \tag{6.56}$$

Proof. Values at output locations $\psi(\overline{y})$ can be written as $K(\overline{y},\overline{x})\,\alpha$, with

$$\alpha = [K(\overline{y},\overline{x})]^{-1}\psi(\overline{y}). \tag{6.57}$$

Therefore, writing equation (6.48) with respect to the kernel section $K(\cdot,\overline{x})$, the Bayesian estimate of $\psi(f(\cdot))$ is given by:

$$\widehat{\psi(f(\cdot))} = K(\cdot,\overline{x})\left[K(\overline{x},\overline{x}) + \sigma^2 I_M\right]^{-1}\psi(\overline{y}) \tag{6.58}$$

$$= K(\cdot,\overline{x})\left[K(\overline{x},\overline{x}) + \sigma^2 I_M\right]^{-1} K(\overline{y},\overline{x})\,\alpha \tag{6.59}$$

$$= K(\cdot,\overline{x})\,U_{\text{EDMD}}^{(K)}\alpha, \tag{6.60}$$

where $U_{\text{EDMD}}^{(K)}$ is defined as:

$$U_{\text{EDMD}}^{(K)} = \left[K(\overline{x},\overline{x}) + \sigma^2 I_M\right]^{-1} K(\overline{y},\overline{x}). \tag{6.61}$$

By defining

$$\beta := U_{\text{EDMD}}^{(K)}\alpha, \tag{6.62}$$

then the sought estimate is given by:

$$\widehat{\psi(f(\cdot))} = K(\cdot,\overline{x})\,\beta \tag{6.63}$$

which indeed is the MAP estimate under the measurement model in 6.56. □

Remark 6.5.2. Technically, it should be assumed that the kernel matrix $K(\overline{y}, \overline{x})$ is invertible, to make (6.57) meaningful. However, from Definition 5.4.1, it follows that kernel sections are linearly independent, therefore it holds almost surely - with respect to the probability of realisations of the noise - that the matrix is invertible, hence the assumption is not needed.

The regularised version of the EDMD approximation for the Koopman operator simply differs from the kernel version in the fact that the former is expressed in the basis provided by the functions in the dictionary D, while the latter is written with respect to kernel sections centred at input locations $K(\cdot, \overline{x})$.
There emerges the greatest difference in the two approaches. By relying on the standard EDMD procedure it is necessary to specify the whole dictionary of function in order to perform the regression, and the latter will be fixed throughout the whole estimation process even though it may not be suitable for the problem at hand. In the kernel version it is instead required to specify only the kernel structure, which has the meaning of a similarity measure among functions. Through the kernel indeed it is possible to encode important prior information - such as *smoothness* - in the estimation, so that the search space of functions will automatically shape itself according to the observed data. However the solution will still lie in a finite-dimensional space so the estimation procedure is well-defined and has a closed-form solution in the case of a Gaussian prior, and Gaussian measurements.

6.6 Duality in RKHS

In the previous sections the connections between the EDMD approach and estimation in the Bayesian framework have been explored and analysed. The kernel version of the DMD formulation has not been characterised yet, though from the duality presented in Section 6.3 it is straightforward to understand that the latter approach should be similar to the EDMD formulation, given that, by framing the problem in a RKHS, it always holds that $N = M$; where N is the number of basis functions and M is the number of datapoints. The two operators however are defined in different ways: EDMD acts on coefficients for the basis functions, while DMD maps values of observables into the same values when the state is propagated through the dynamics. In what follows the connection between the two formulations in RKHS is discussed, finally accomplishing the objective of bridging the different frameworks.
Following the DMD perspective, given the values of an observable at the input locations

$\psi(\overline{x})$, the aim is to reconstruct the values of the observable at output locations $\psi(\overline{y})$, as introduced in Chapter 4. The latter problem is projected into the RKHS as follows.
An estimate of the observable function $\psi(\cdot)$ is obtained through the measurements $\psi(\overline{x})$, by performing the regression from the actual input locations $\overline{x}$ to the observable mappings $\psi(\overline{x})$. This is nothing but the projection of ψ on the finite-dimensional space spanned by the kernel sections centred on datapoints, given by:

$$\hat{\psi}(\cdot) = K(\cdot, \overline{x}) [K(\overline{x}, \overline{x})]^{-1} \psi(\overline{x}). \tag{6.64}$$

Then the values which the latter estimates takes on the output locations $\overline{y}$ can be seen as a noisy observation of the actual Koopman composition $\psi_+(\cdot) = \psi(f(\cdot))$, i.e.,

$$\hat{\psi}(\overline{y}) = K(\overline{y}, \overline{x}) [K(\overline{x}, \overline{x})]^{-1} \psi(\overline{x}), \tag{6.65}$$

which indeed highlights the mapping sought by the DMD approach. The kernel version of the Dynamic Mode Decomposition of the Koopman operator then takes the form of:

$$U_{\mathrm{DMD}}^{(K)} = K(\overline{y}, \overline{x}) [K(\overline{x}, \overline{x})]^{-1} \tag{6.66}$$

so that

$$\psi(\overline{y}) = U_{\mathrm{DMD}}^{(K)} \psi(\overline{x}). \tag{6.67}$$

It should be noted that in (6.65) a second regression problem is introduced. In particular, the values of the observable corresponding to the output locations are reconstructed through the knowledge of $\overline{x}$, $\overline{y}$ and $\psi(\overline{x})$. This is actually not necessary, as it is clear from Proposition 6.5.1, where the coefficients are taken as $\alpha = [K(\overline{y}, \overline{x})]^{-1} \psi(\overline{y})$, yielding the exact observed values. That formulation however can be employed only when the knowledge of the values of the propagated points are known. For instance, in the case of subsequent propagations - which is matter for Chapter 8 - this no longer true. Moreover, the second layer of regression allows to add regularisation also for the reconstruction of the observable function, which is neglected here, considering the evaluation of the observable as noiseless.
By recalling the formula of the final estimate for the composition of the observable and the transition function, it turns out to be trivial to link the EDMD and DMD approaches

in their kernel versions:

$$\widehat{\psi\left(f\left(\cdot\right)\right)} = K\left(\cdot,\overline{x}\right)\beta \tag{6.68}$$
$$= K\left(\cdot,\overline{x}\right)U_{\text{EDMD}}^{(K)}\alpha \tag{6.69}$$
$$= K\left(\cdot,\overline{x}\right)U_{\text{EDMD}}^{(K)}\left[K\left(\overline{x},\overline{x}\right)\right]^{-1}\psi\left(\overline{x}\right) \tag{6.70}$$
$$= K\left(\cdot,\overline{x}\right)\left[K\left(\overline{x},\overline{x}\right)+\sigma^2 I_M\right]^{-1}K\left(\overline{y},\overline{x}\right)\left[K\left(\overline{x},\overline{x}\right)\right]^{-1}\psi\left(\overline{x}\right) \tag{6.71}$$
$$= K\left(\cdot,\overline{x}\right)\left[K\left(\overline{x},\overline{x}\right)+\sigma^2 I_M\right]^{-1}U_{\text{DMD}}^{(K)}\psi\left(\overline{x}\right) \tag{6.72}$$
$$= K\left(\cdot,\overline{x}\right)\left[K\left(\overline{x},\overline{x}\right)+\sigma^2 I_M\right]^{-1}\psi\left(\overline{y}\right). \tag{6.73}$$

6.7 Illustrative example for estimation

The Koopman operator framework, and in particular the proposed approach making use of kernel methods, allows to estimate the evolution of a generic observable when propagated through the system. In the following chapters this capability will be exploited to tackle the Reinforcement Learning problem, leaning on the intuition that the reward (or cost) itself can be thought as an observable.

So far, the approach introduced in this chapter only deals with estimation. However, the following example already frames the problem in terms of a cost function, whose composition with the dynamics needs to be estimated. In other terms, the final task for an estimation algorithm in this framework would be the estimation of the cost of the next state, given the current one.

In order to illustrate the effective applicability of the proposed framework, a simple example is shown, in which the same task is solved both through the Extended Dynamic Mode Decomposition, which rely on a fixed dictionary of functions, and the proposed generalisation with kernels, which instead use a standard *Radial Basis Function* kernel. For the sake of clarity, a scalar system is considered so that transition maps and observables can be easily plotted, and the results visually inspected. In particular, the following discrete-time autonomous dynamics are taken into account:

$$x_{k+1} = f\left(x_k\right) = -x_k + \frac{3}{\left(1+x_k^2\right)} + \frac{1}{2}\sin\left(2x_k\right), \tag{6.74}$$

which are characterised by an oscillatory behaviour, and exhibit an equilibrium point in $x^{(e)} \simeq 0.988$.

This system is a modification from the one considered by Mauroy and Gonçalves, 2020,

in the example of the *genetic toggle switch* (Gardner, Cantor, and Collins, 2000). The latter system is derived from the statistical mechanics of *Transcription Factor Binding*, and in particular it is composed of Hill functions, which represents an approximation of how the production rate of a gene depends on the transcription factor concentration (Alon, 2006; Phillips, Kondev, and Theriot, 2008; Dill and Bromberg, 2011; Bintu et al., 2005; Lamprecht and Zotin, 2019).
The tasks to solve will be to reconstruct both the transition function given in (6.74) and its composition with the following cost function:

$$c\left(x_k\right) = \left\| x_k - x^{(r)} \right\|^2, \tag{6.75}$$

for an arbitrary reference value $x^{(r)}$, which would be the minimiser of the cost.
Both the 2 objectives can be easily embedded in the Koopman operator framework by considering the prediction of two different observables:

$$\psi^{(I)}\left(x\right) = x; \tag{6.76}$$

$$\psi^{(c)}\left(x\right) = c\left(x\right). \tag{6.77}$$

In particular, the observable in (6.76) corresponds to the identity function, so that learning its composition with the dynamics would mean to actually learn the transition function, which is the first task. The other observable instead is equal to the cost, which gives the sought composition. The latter framework then follows the same idea which will be enforced later, i.e., that the cost (or reward) should be thought as an observable. The aim of the experiment will be to prove that the infinite-dimensional implicit dictionary of functions provided by the Gaussian kernel can solve both problems, representing a meaningful approach when no explicit prior knowledge on the system is available. The reconstruction of the dynamics through EDMD method would indeed be easy if the three functions that linearly gives the state-transition function in (6.74) were all explicitly provided in the dictionary. However, this would not be true for the second task, in which the latter functions are no longer suitable to the regression, because they cannot account for the composition of the transition and the reward. This instead is well handled by the reconstruction with kernels, as proved by the following simple example.
In order to make the Extended Dynamic Mode Decomposition as competitive as possible, the user-defined dictionary of functions D is selected as:

$$D = \left\{1, x, x^2, \sin\left(2x\right)\right\} \cup \mathbb{H}^4, \tag{6.78}$$

where $\mathbb{H}^n$ is the set of the first n-th simple Hills functions with even powers, defined as:

$$\mathbb{H}^n = \bigcup_{i=1}^{n} \left\{ \frac{1}{1+x^{2i}} \right\}. \tag{6.79}$$

Therefore there are $N = 8$ functions in the dictionary $D = \{\psi_i(\cdot)\}_{i=1}^{N}$, with which the reconstruction can be performed. Moreover, the chosen function allows for a perfect reconstruction of the dynamics, as explained earlier. Note that also the cost function itself belongs to the dictionary, being equal to a linear combination of the functions $1, x, x^2$ in the scalar case. The latter is however not true for the composition of the dynamics and the cost function in (6.75), which in this would require to compute the squared norm of the transition and take every single function. In other words, it holds that:

$$\psi^{(I)}(f(\cdot)) \in \operatorname{span}\{D\}; \qquad \psi^{(c)}(\cdot) \in \operatorname{span}\{D\}; \qquad \psi^{(c)}(f(\cdot)) \notin \operatorname{span}\{D\}. \tag{6.80}$$

As explained in Section 6.4, the original EDMD algorithm can be expressed through kernel functions by defining a finite dimensional kernel which expresses the inner product between basis function, as given in (6.43). Therefore it is sufficient to compare the performances of the same algorithm when using the custom kernel given by:

$$K^{(D)}(x, \tilde{x}) = \sum_{i=1}^{N} \psi_i(x)\, \psi_i(\tilde{x}) \tag{6.81}$$

in which the prior variance of the coefficients $\Lambda = I_N$ is taken to be the identity matrix, without loss of generality. The regularisation will indeed be adjusted by the parameter σ^2, because, as can be appreciated from equation (6.34), what matters is only the ratio $\sigma^2\Lambda^{-1}$.

The reconstruction over the implicit infinite-dimensional space of function can be achieved by different choices of the kernel. In this case a standard Radial Basis Function (RBF) kernel is considered:

$$K^{(G)}(x, \tilde{x}) = \exp\left(-\varrho \left\| x - x^{(r)} \right\|^2\right). \tag{6.82}$$

In the actual simulations the reference state is taken to be $x^{(r)} = 0$ for the cost in (6.75). Data from the system are given as snapshots pair $\{x_k, x_{k+1}\}$, where the samples are collected as noisy input of the state-transition function in (6.74), according to the measurement model:

$$x_{k+1} = f(x_k) + w_k \tag{6.83}$$

in which w_k is a Gaussian zero-mean perturbation distributed as $w_k \sim \mathcal{N}(0, \overline{\sigma}^2)$.
Different scenarios are taken into account, with different noise levels, given by:

$$\overline{\sigma}^2_{(1)} = 0; \qquad \overline{\sigma}^2_{(2)} = 0.2; \qquad \overline{\sigma}^2_{(3)} = 0.5. \tag{6.84}$$

In any case, $M = 50$ available pairs are considered, consisting of 5 trajectories, each of 10 points. The initial point of every trajectory is sampled from a uniform distribution between 0 and 7.
By considering the variance of the output locations approximately equal to the variance of the input locations - which is reasonable given the behaviour of the system in (6.74) - the different scenarios can be understood in terms of the signal-to-noise ratio (SNR), given by:

$$\mathrm{SNR}_{(1)} = \infty; \qquad \mathrm{SNR}_{(2)} = 20; \qquad \mathrm{SNR}_{(1)} = 8. \tag{6.85}$$

In the actual experiments the regularisation parameter is set to be equal to the variance of the noise injected in the measurements, i.e., $\sigma^2 = \overline{\sigma}^2$.
The hyper-parameter ϱ for the Gaussian kernel has been optimised for each new reconstruction by minimising the negative marginal log-likelihood on a small grid. The latter is indeed an important step as it defines the shape of the kernel sections, which are linearly combined to form the estimate. While it's true that with the kernel approach only the kernel function must be chosen a priori, the hyper-parameters play a fundamental role in shaping the estimate. The advantage is that, unlike with EDMD approach in which the dictionary of function has to be selected a priori, the hyper-parameters can be adjusted based on available data. Therefore the goodness of the estimate also depends on a careful choice of the hyper-parameters.
As explained in the previous section, the kernel formulation of the Koopman operator yields an $M \times M$ matrix, where M is the number of training examples. A well-known difficulty for kernel learning is the computational complexity, which is bound by the necessity of handling the kernel matrix. Therefore the proposed approach suffer from the same issue, as the matrix inversion is a costly operation, which leads to a complexity of at least $O(M^3)$. However, given the benefits and the popularity of kernel methods, there is already an extensive literature on methods to reduce the computational cost of these approaches, e.g. by carefully selecting a subset of the kernel matrix entries, dealing with a low-rank approximation, or resorting to the random projection method (Cesa-Bianchi, Mansour, and Shamir, 2015).
In Figure 6.1 the reconstruction of the transition map with noiseless data is addressed. As expected, the true curves and the one obtained by EDMD are overlapping, since the

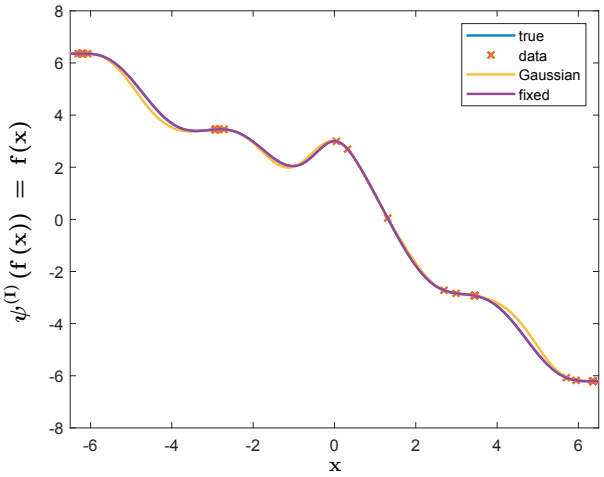


Figure 6.1: Example of reconstruction of the state-transition map, formally estimating the identity observable, obtained with the two kernels, where "fixed" corresponds to the estimation through EDMD technique, and "Gaussian" label is related to the RBF kernel. Data for the reconstruction are represented with a cross.

provided basis function allows for a perfect estimation of the dynamics.

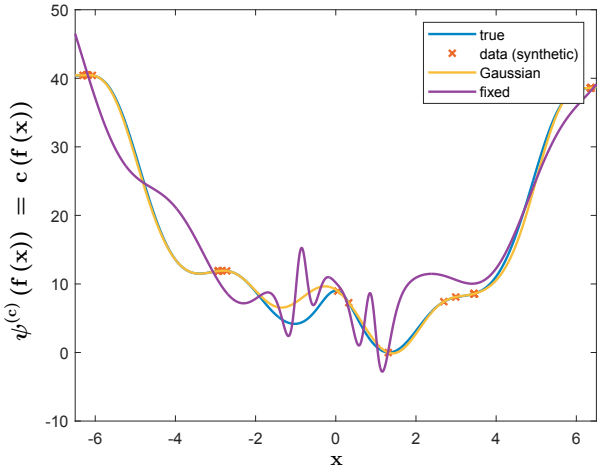


Figure 6.2: Reconstruction of the composition of the cost function and the state-transition map with the two different kernels related to EDMD and RBF regression. The label "fixed" corresponds to the estimation with the chosen dictionary of functions D, while "Gaussian" is the one coming from the kernel perspective. Data for the reconstruction are represented with a cross.

The drawbacks of dealing with a fixed dictionary of functions become evident in the second task, namely the reconstruction of the composition of the cost and the dynamics. The bad estimation is of course due to the fact that the basis functions in the dictionary

cannot linearly describe the objective function, therefore resulting in a shape which is very different from the original curve, no matter how many data may be provided. The latter is evident in Figure 6.2, in which the reconstruction through the RBF kernel can instead give a good representation of the sought composition. Furthermore, the Gaussian kernel estimator improves as more data are provided, contrary to EDMD technique.
In order to better understand the behaviour of the different perspectives, their performances have been analysed by performing the reconstruction under different noise levels, given in (6.84), and evaluating the normalised L^2 norm (NLN) of the difference between the estimate and the true function, given by:

$$\text{NLN} := \left\| \widehat{\psi^{(c)}}\left(x_{k+1}\right) - c\left(f\left(x_k\right)\right) \right\| / \left\| c\left(f\left(x_k\right)\right) \right\| . \tag{6.86}$$

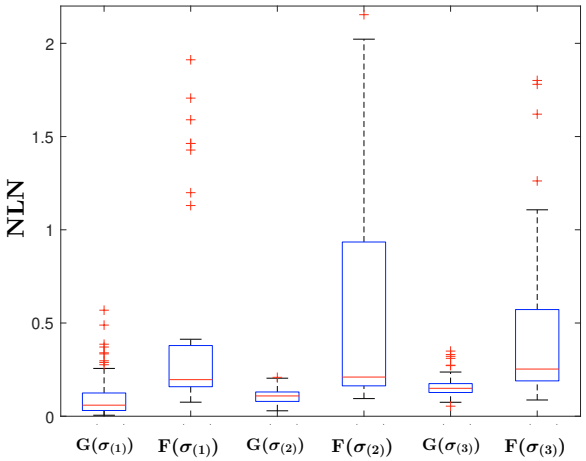


Figure 6.3: Evaluation over 100 trials of the NLN for the three different noise level considered. The label "G" corresponds to the estimation with the Gaussian kernel while "F" denotes the reconstruction with the fixed dictionary of functions. The noise level is specified within parenthesis. The boxplots represents the represent the 25-th and 75-th percentiles around the median, while the bar includes all data apart from the outliers, marked with a red plus sign.

Figure 6.3 shows the result of 100 Monte Carlo simulations, under the different level of noise. It can be seen that if the dictionary of function does not allow for a perfect reconstruction, then the approach with Gaussian kernel outperforms the EDMD technique. Also, the milder prior imposed by the Gaussian assumption helps to effectively deal with the noise, which could instead induce very wrong coefficients for the linear reconstruction with the fixed dictionary, due to misleading data.

7

Koopman for control

While in previous chapters the Koopman operator has been used to perform the reconstruction of autonomous systems, in what follows the control setting is addressed.
The control problem is formalised in the first section, which is followed by a thorough introduction of the Koopman operator framework for controlled systems, also by reviewing previous work on this topic which recently emerged from the literature. The linear setting provided by the Koopman operator proves to be advantageous as the linear control problem can be solved sharply. Different linear approximations for nonlinear controlled systems are analysed: although they succeed in approximating the dynamics, they fail to provide a simple form for the design of the controller, unlike the new approach proposed in the next Chapter.

7.1 The control framework

The classical definition of the Koopman operator (Definition 2.2.3) deals with autonomous dynamics, i.e., systems for which the time-invariant vector field $f : X \to X$ is a function only of the state $x \in X$. Considering such systems is surely useful from the point of view of the analysis: a large part of the literature on the Koopman operator is indeed concerned with understanding the properties and the behaviour of generic nonlinear flows (Mauroy and Sootla, 2017; Mezić, 2005; Mauroy and Mezić, 2012; Kühner, 2019; Bátkai, Fijavz, and Rhandi, 2017; Lasota and Mackey, 2013; Mezić and Wiggins, 1999; Mezić and Banaszuk, 2004; Lan and Mezić, 2013; Mezić, 2015; Mauroy, Mezić, and Moehlis, 2013). The Koopman operator indeed turned out to be a useful tool for this purpose, revealing geometric properties of the system and for example allowing the derivation of new conditions for stability (Mauroy and Mezić, 2016; Rantzer, 2001; Vaidya and Mehta, 2008; Susuki and Mezić, 2014; Sootla and Mauroy, 2017). The latter can give different insights with respect to the usual state-space view, as it is one of the most important

properties of a dynamical system and therefore something that should be thoroughly understood.
Nonetheless the final aim of this work is to investigate how the Koopman operator - and especially its embedding in the RKHS framework, as given in Chapter 6 - can be exploited for the *control task*.
When considering controlled systems, the dynamics are affected not only by the state x, but also on the control input a, which is in general a function depending on time, as stated by the following definition, which considers the continuous-time setting.

Definition 7.1.1 (Control input)**.** A function mapping time to inputs, $a : [0, \infty) \to A,\ A \subseteq \mathbb{R}^p$, is called a control. The set of control functions is denoted as Ψ_A.

The expression for an autonomous dynamical system, specialising the differential formulation in (2.1), is then given by:

$$\dot{x}(t) = f(x(t), a(t)), \tag{7.1}$$

where now the vector field maps state-action pairs into states, i.e., $f : X \times A \to X$.
The controlled dynamical system in (7.1) models the fact that part of the dynamics can be significantly changed through the exogenous input $a(t)$, so that the same system with two different control inputs can have different behaviours.
In control theory the controlled dynamics are analysed with different objectives. The degrees of freedom given by the choice of the control input allow for a whole spectrum of systems in the form of (7.1), hence different tasks can be pursued. For example, a common goal is to find the class of *stabilising controllers*, for a given set $\tilde{X} \subseteq X$, in which once again linearity proves to be a key property. Indeed for *Linear and Time Invariant* (LTI) systems the latter property holds globally, i.e., $\tilde{X} = X$ (Willems, 1970); however for nonlinear systems this is in general not possible, and it is sought a class of controllers that induce a stable dynamics in $\tilde{X} \subset X$ for the flow associated with the vector field (Lefschetz, 1963; Hirsch and Smale, 1974).
The general objective of the control problem is then to design a controller so that the performances of the system meet certain criteria. Usually the latter are defined through a functional which depends on the dynamics, so that the goal is to find the best control $a^*(t)$ which minimise or maximise this performance functional.
In order to make precise the control task, define the functional $J : \Psi_A \to \mathbb{R}$ as

$$J(a(t)) = \int_{t_0}^{t_f} c_r(x(t), a(t), t)\, dt + c_f(x(t_f), t_f), \tag{7.2}$$

which evaluates the performance of the controlled system under the action of the controller $a(t)$. The function c_{r} - often referred to as *running cost* - is the instantaneous measure of the system behaviour, while c_{f} denotes the *terminal cost*.
The general formulation of the control problem is to find the best control $a^*(t)$ so that the objective functional is minimised (or maximised):

$$a^*(t) = \underset{a \in \Psi_A}{\arg\min} \{J(a(t))\}, \tag{7.3}$$

$$\text{subject to:} \quad \dot{x}(t) = f(x(t), a(t)), \tag{7.4}$$

$$x(t_0) = x_0. \tag{7.5}$$

Since the objective functional depends on the evolution of the system under a particular control, the same problem could be rewritten in terms of the flow. As done for the autonomous case, define as

$$x(t) = \Phi_a^t(x_0) \tag{7.6}$$

the solution of the differential equation in (7.1), affected by the control $a(t)$, with initial condition given by $x(0) = x_0$. For a specific control action, the system evolution is fixed and the performance index depends only on the flow, which in turn depends on the control and on the initial state and time. However, usually the initial condition and the dynamics are not a decision variable for the control problem, so that the functional can be only minimised with respect to $a(t)$, given f and x_0.
Since the problem consists of minimising a functional with respect to a set of functions Ψ_A, it pertains to the field called *calculus of variations*. A way to find the best possible control minimising (7.3) is given by the *Pontryagin's maximum principle* (Pontryagin et al., 1962; McShane, 1989; Kirk, 2004; Lee et al., 1967), which states a necessary condition for any control to be optimal, along with the optimal state trajectory.
The derivation of the conditions is achieved by taking the Legendre transform of a modified cost function, to which Lagrange multipliers have been added. The resulting system is a *Hamiltonian* system, so that some necessary condition on optimality based on a first-order Taylor expansion can be derived. The latter allows to formulate the problem as a standard maximum condition on the control Hamiltonian, provided other technical conditions are satisfied. The variational problem is turned in this way into a static optimisation problem given by a two-point boundary value problem, which is definitely easier. The necessary conditions thus obtained becomes also sufficient under certain convexity conditions on the objective and constraint functions; which are related to the positive-definiteness of the Hessian of the Hamiltonian (Mangasarian, 1966; Kamien and

Schwartz, 1971).

The Maximum Principle stems from a variational approach, and requires the optimal curve to be at least at a local minimum, i.e., neighbouring curves should not yield smaller costs. The necessary conditions for optimality are indeed obtained by imposing that the candidate solution is optimal with respect to small variations of the latter curve. While this "spatial" view has been useful to derive the aforementioned necessary conditions, there exists another perspective, which considers optimality over time, that yield sufficient conditions for the control to minimise the objective functional. The "temporal" view is the one followed by Dynamic Programming, stating that the optimal curve remains optimal at intermediate points in time.

Dynamic Programming is a mathematical optimisation method developed by Bellman, 1954, whose idea is to break down a complicated problem into a sequence of simpler sub-problems, in a recursive manner. This is the foundation of Bellman's optimality principle, which says that once an optimal trajectory is found for the whole interval $[t_0, t_\mathrm{f}]$ by solving the Optimal Control problem on that interval, the resulting trajectory is also optimal on all sub-intervals of the form $[\bar{t}, t_\mathrm{f}]$, with $\bar{t} > t_0$, provided that the initial condition at time $\bar{t}$ was obtained from running the system forward along the optimal trajectory from time t_0 (Bellman, 1957). The term *Bellman* equation usually refers to the dynamic programming equation associated with discrete-time optimization problems (Kirk, 2004); while in continuous-time optimization problems, the analogous is given by a partial differential equation which is called the *Hamilton–Jacobi–Bellman* (HJB) equation.

The latter gives a necessary and sufficient condition for optimality of a control with respect to a loss function. It is, in general, a nonlinear partial differential equation in the value-function, which means its solution is the value-function itself. Once this solution is known, it can be used to obtain the Optimal Control by taking the maximiser (or minimiser) of the Hamiltonian involved in the HJB equation.

The Dynamic Programming approach has extensively been exploited in Optimal Control and later in the Reinforcement Learning field, so that the value-function actually became the key quantity to deal with. In particular, it represents the objective that the general Reinforcement Learning algorithm aims to minimise. Indeed many results in this field are derived from the Bellman equation, as well as the majority of the algorithms. The Reinforcement Learning framework is presented in the next chapter, as well as a way to tackle the related Optimal Control problem which exploits the Koopman operator. It is therefore of primary importance to understand how the control setting can be described by the Koopman framework.

7.2 Koopman operator for controlled systems

The extension of the Koopman operator which allows to take into account control inputs is a fairly new paradigm which can be regarded as the smooth consequence of the latest advancements brought by the Koopman perspective for the analysis of autonomous systems.
A natural way of defining this extended operator is to rely on a different definition of *observables*, so that the composition operator actually operates in the same manner. In this regard, it is necessary for the new observable to deal with the control input, which affects the dynamics of the system. The following definition of *control-dependent* observable, extrapolated from Proctor, Brunton, and Kutz, 2018, allows to extend the definition of the Koopman operator in a straightforward way.

Definition 7.2.1. A control-dependent observable is a scalar-valued function taking as argument both the state and the control, i.e., $\psi : X \times A \to \mathbb{R}$.

In order to stick to the idea of the Koopman operator as an operator that propagates the system forward in time according to the observable given in input, the composition operator for controlled system is defined by Mauroy, Mezić, and Yoshihiko, 2020 as follows.

Definition 7.2.2 (Koopman operator for controlled systems)**.** Let Φ_a^t be the flow induced by a controlled dynamical system as (7.1), under the control $a(\cdot) \in \Psi_A$. Consider a Banach space Ψ of control-dependent observables $\psi : X \times A \to \mathbb{R}$ closed under composition. The family of Koopman operators $\mathcal{U}^t$ associated with the flows $\Phi_a^t : X \times \Psi_A \to X$, $t \in \mathbb{R}^+$, is defined as:

$$\mathcal{U}^t[\psi](x, a(\cdot)) := \psi\left(\Phi_a^t(x), \mathcal{Z}^t[a(\cdot)]\right), \tag{7.7}$$

$\forall \psi \in \Psi$, $\forall x \in X$ and $\forall a(\cdot) \in \Psi_A$; where $\mathcal{Z}^t : \Psi_A \to \Psi_A$ indicates the left-shift semigroup of operators, defined by

$$\mathcal{Z}^t[a(t_0)] = a(t_0 + t) \tag{7.8}$$

which propagates forward in time the control input.

The definition above guarantees that the family of Koopman operators still enjoy the semigroup property, as shown in the next proposition. This is true under the same condition of the autonomous case, i.e., when the flow is itself a semigroup.

Proposition 7.2.3 (Semigroup property of Koopman operator for controlled systems)**.** *If the flow induced by a controlled dynamical system $\Phi_a^t(\cdot)$ on the set $X \times \Psi_A$ is a semigroup*

w.r.t. $t \in \mathbb{R}$*, then the family of Koopman operators associated with the same controlled dynamical system is a semigroup of operators.*

Proof. By relying on the semigroup property for the flow, it is possible to write:

$$\mathcal{U}^{t+s}[\psi](x, a(\cdot)) = \psi\left(\Phi_a^{t+s}(x), \mathcal{Z}^{t+s}[a(\cdot)]\right) \tag{7.9}$$
$$= \psi\left(\Phi_a^{t}(\Phi_a^{s}(x)), \mathcal{Z}^{t}[\mathcal{Z}^{s}[a(\cdot)]]\right) \tag{7.10}$$
$$= \mathcal{U}^{t}[\psi(\Phi_a^{s}(x), \mathcal{Z}^{s}[a(\cdot)])] \tag{7.11}$$
$$= \mathcal{U}^{t}[\mathcal{U}^{s}[\psi]](x, a(\cdot)) \tag{7.12}$$

which proves the semigroup property for the Koopman operator. □

Note that in Definition 7.2.2 the Koopman operator evolves also the control input acting on the system, through the left-shift operator. This is a natural choice, but there are other viable options, as underlined by Mauroy, Mezić, and Yoshihiko, 2020. Another meaningful definition is indeed given by:

$$\mathcal{U}^{t}[\psi](x, \overline{a}(\cdot)) := \psi\left(\Phi_a^{t}(x), \overline{a}(\cdot)\right), \tag{7.13}$$

obtained by considering an input signal $\overline{a}(\cdot)$ which is reset after every propagation. Evolution curves for the system described by (7.13) correspond to a re-initialisation of the state with the propagated value and the application of the same control input $\overline{a}(\cdot)$. In this case $\mathcal{U}^t$ does not have the semigroup property, which is broken by the re-initialisation of the controller.
Given that in the latter definition the control input always remains the same at each evolution, it is easy to see that the formulation in (7.13) corresponds to a family of operators parameterised by $\overline{a}$:

$$\mathcal{U}_{\overline{a}}^{t}[\psi](x) = \psi(\Phi_{\overline{a}}^{x}(t)) \tag{7.14}$$

$\forall \psi : X \to \mathbb{R}$, in which then the dependence of the observable on the control input is dropped. The notion of a Koopman operator parameterised by the control will be a key tool in the next chapter.
The same definition for the controlled dynamics can be given in the discrete-time domain. In this regard, the following definition settles the discrete-time control input.

Definition 7.2.4. A sequence $a : \mathbb{N} \to A,\ A \subseteq \mathbb{R}^p$, is called a discrete-time control. The set of control sequences is denoted as l_A.

The corresponding definition for a discrete-time Koopman operator is obtained from the continuous-time one by fixing a time-interval t, and it is given by Korda and Mezić, 2018 as follows.

Definition 7.2.5. Consider an extended space $X \times l_A$; the Koopman operator for the skew product system given by

$$\left(\Phi\left(x, a\left(0\right)\right), \mathcal{Z}\left[a\left(\cdot\right)\right]\right) : X \times l_A \to X \times l_A \tag{7.15}$$

is defined as

$$\mathcal{U}\left[\psi\right]\left(x, a\left(\cdot\right)\right) = \psi\left(\Phi\left(x, a\left(0\right)\right), \mathcal{Z}\left[a\left(\cdot\right)\right]\right), \tag{7.16}$$

where $\mathcal{Z}$ is the left-shift operator defined as:

$$\mathcal{Z}\left[a_k\right] = a_{k+1}, \tag{7.17}$$

$\forall k \in \mathbb{N}$.

Note that the discrete version of the Definition in (7.13) is inherently simpler, as there is no explicit dependence on time, and the Koopman operator evolves the system over a fixed time step. This makes it a convenient description with respect to the controller, which is reset to the same quantity at each propagation. Hence, given that the evolution on X depends on the input $a\left(\cdot\right)$ just through the first element $a\left(0\right)$, it is possible to consider the space given by the Cartesian product $X \times \mathbb{R}^p$, interpreting input variables as additional state variables, so that the Koopman operator is defined as (Proctor, Brunton, and Kutz, 2018):

$$\mathcal{U}\left[\psi\right]\left(x, a\right) = \psi\left(\Phi\left(x, a\right), a\right), \tag{7.18}$$

$\forall \psi : X \times \mathbb{R}^p \to \mathbb{R}$. In this formulation, the control acts as a parameter of the system, and its evolution over time is not considered. This is equivalent to defining a family of Koopman operators parameterised by a, as:

$$\mathcal{U}_a\left[\psi\right]\left(x\right) = \psi\left(\Phi_a\left(x\right)\right). \tag{7.19}$$

Each operator is associated with the map $\Phi_a\left(\cdot\right)$, so that the sequence of operators $\mathcal{U}_{a(0)}, \mathcal{U}_{a(1)}, \cdots$ corresponds to the evolution under the control input $a\left(\cdot\right)$.
Having understood how the Koopman operator can handle control inputs, and recalling that the Koopman operator yield a global linearisation of the dynamics, it would be interesting to understand if it is possible to exploit this property for the design of the controller. In this regard, the next section will highlight why the linearity property is so

important in the Optimal Control problem, while after that the Koopman setting will be considered.

7.3 Optimal controller

The following introduction of the Optimal Control problem under the linear assumption can be presented both in the continuous-time setting and in the discrete-time domain. Since in what follows only the latter perspective will be considered, as it has been explained in Chapters 3 and 4, also for the current section the discrete setting is assumed. In the case that the dynamics represented in (7.1) are linear in both the state and the control input, the system evolution is specified as:

$$x_{k+1} = A_k x_k + B_k a_k \tag{7.20}$$

where $A \in \mathbb{R}^{d\times d}$ and $A \in \mathbb{R}^{d\times p}$. Moreover if the matrices A and B are constant over time, the evolution is described by:

$$x_{k+1} = A x_k + B a_k \tag{7.21}$$

which characterises a LTI system. Its solutions are known to be exponential functions with respect to time, whose coefficients depend on the initial conditions. This greatly simplifies the complexity of the problem.
As explained in the previous section, in order to define a *control problem* it is required to specify a running cost and a terminal cost. If the former is given by a quadratic form of the state and input, and the latter by a quadratic form of the state, i.e.,

$$c_r\left(x_k, a_k, k\right) = x_k^\top Q x_k + u_k^\top R u_k, \tag{7.22}$$

$$c_f\left(x_{\overline{k}}, \overline{k}\right) = x_{\overline{k}}^\top Q_{\mathrm{f}} x_{\overline{k}}, \tag{7.23}$$

the associated Optimal Control problem is denoted as Linear Quadratic Regulator (LQR) (Kwakernaak and Sivan, 1972; Anderson and Moore, 1990). The matrices $Q, Q_{\mathrm{f}} \in \mathbb{R}^{d\times d}$ are positive semi-definite and are designed to adjust the weights affecting the state variables, while the matrix $R \in \mathbb{R}^{p\times p}$ is positive definite and is there to penalise the norm of the control input.
For the latter specific formulation, the Optimal Control problem has an explicit solution as a feedback law from the state, which can be written as:

$$a_k^* = C_k x_k. \tag{7.24}$$

This indeed results in a dynamic control gain for the finite-horizon scenario, in which the Optimal Control depend on the solution of the *Difference Riccati Equation* (DRE), as detailed by Reid, 1972 and Lancaster and Rodman, 1995. The Riccati equation exploits the same principle of the Hamilton–Jacobi–Bellman equation, and indeed can be seen as a special case of the latter for the LQR setting. In particular it is solved by constraining the value-function, evaluated at the last point of the interval, to be equal to the terminal cost, and then proceeding backwards in time, guaranteeing the optimality. Letting the final instant go to infinity, the DRE becomes stationary, because the transient has a negligible impact. Therefore in the infinite-horizon case, the optimal feedback is obtained as the solution of the *Algebraic Riccati Equation* (ARE) which does not depend on time, and it has a closed-form solution. The Optimal Control is a static feedback from the state, which can be written as (7.24) with a constant matrix $C^* \in \mathbb{R}^{p \times d}$.
For the infinite-horizon control problem, in which the dynamics are linear and time-invariant, the optimal controller is given by:

$$a_k = C^* x_k, \tag{7.25}$$

where

$$C^* = -\left(R + B^\top P B\right)^{-1} B^\top P A. \tag{7.26}$$

The matrix $P \in \mathbb{R}^{d \times d}$ is the matrix defining the value-function, and it can be computed by the following relation:

$$P = Q + A^\top P A - A^\top P B \left(R + B^\top P B\right)^{-1} B^\top P A, \tag{7.27}$$

which together with the conditions $P = P^\top \succeq 0$ characterizes P.
The fact that there is a well-defined and easily computable solution to the Optimal Control problem makes the linear framework a very appealing setting in which to formalise the minimisation of the objective functional. However, most of the meaningful problems involve nonlinear dynamics, as underlined in Chapter 2, with the trivial example of the pendulum.
As a matter of fact, the linear setting is so attractive that there has been a great deal of research effort into how to transform a nonlinear system into a linear one. An example is given by *Feedback Linearisation*. The latter is a technique to linearise a nonlinear system exploiting the expressiveness of the controller. A change of variables and a suitable control input is sought so that the control removes the nonlinearities in the new coordinates (Isidori, 1995; Khalil, 2002).
Nonetheless, a more natural approach would prescribe to approximate the dynamics

with a linearised version. Given that the dynamics are not far away from their linearised version, it is easy to understand that the corresponding solution of the Optimal Control problem would be a meaningful choice for the control input. This is even more true given the fact that the nonlinear control framework yields a hard problem and often do not provide explicit solutions.
However, the approximated system may be a good representation of the original one only in a small neighbourhood around the point which is taken into account for the linearisation. The Koopman operator instead provides a global linearisation of the dynamics, and therefore can be seen as the right perspective for this purpose. It indeed allows to retrieve a linear description, which can be used to design the controller, in which case it is an easy task. The finite-dimensional approximation however is in general not exact, as it has been explained in Chapter 4, therefore also the control solution will not be optimal for the original system.
In what follows the latest techniques for addressing the control problem through the Koopman operator are reviewed.

7.4 Koopman exact form with control

Under certain specific conditions on the dynamics, and by making use of the Koopman operator framework, it is possible to find an exact linearisation of the controlled nonlinear system in (7.1), which is however significantly different from the simple form in (7.21). In order to obtain an exact linearisation, it is required to keep the input matrix $B(x, a)$ a nonlinear function, dependent on both the state and the input.
The latter is the core result of a recent work by Iacob, Tóth, and Schoukens, 2022, which indeed discovered that it is always possible to represent exactly the Koopman operator in its control form through a *linear parameter-varying* model. Therefore the Koopman operator can be completely characterised by the matrices $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times p}$ corresponding to the free evolution and the input of the control, where A is a fixed matrix and $B(x, a)$ depends on both the state and the input. The following theorem is a rewriting of the main results presented by Iacob, Tóth, and Schoukens, 2022. The same result holds - with minor modifications - both for the continuous-time and the discrete-time setting. In order to refer only to one form of the dynamics, the dependence on time is dropped.

Theorem 7.4.1. *Given a nonlinear vector field as in* (7.1)*, write its decomposition in*

the autonomous part and input-related dynamics, as:

$$f(x,a) = f(x,0) + \tilde{f}(x,a) \tag{7.28}$$

with $\tilde{f}(x,0) = 0$. *Consider a lifting map* $\Psi : X \to \mathbb{R}^n$ *of class* $\mathcal{C}^1$ *so that* $\Phi(f(x,0)) \in \operatorname{span}\{\Phi\}$, *then there exists a finite-dimensional lifted form, of dimension* n, *with the continuous-time and discrete-time versions given below.*

- *Continuous-time.*

$$\dot{\Phi}(x(t)) = A\Phi(x(t)) + \tilde{B}(x(t), a(t)) \tag{7.29}$$

with $A \in \mathbb{R}^{n\times n}$ *and* $\tilde{B} : X \times \Psi_A \to \mathbb{R}^n$ *defined as*

$$\tilde{B}(x(t), a(t)) = \frac{\partial \Phi}{\partial x}(x(t))\, \tilde{f}(x(t), a(t)). \tag{7.30}$$

Furthermore assume that $\tilde{B} : X \times \Psi_A \to \mathbb{R}^n$ *is continuously differentiable in* $a(t)$, *continuous in* $x(t)$, *satisfying* $\tilde{B}(x(t), 0) = 0$ *and* Ψ_A *is a convex set containing the origin. Then by defining*

$$B(x(t), a(t)) = \int_0^1 \frac{\partial \tilde{B}}{\partial a}(x(t), \zeta a(t))\, \mathrm{d}\zeta \tag{7.31}$$

the controlled dynamics are completely characterised by:

$$\dot{\Phi}(x(t)) = A\Phi(x(t)) + B(x(t), a(t))\, a(t) \tag{7.32}$$

- *Discrete-time. With the additional assumption that* X *is convex:*

$$\Phi(x_{k+1}) = A\Phi(x_k) + \tilde{B}(x_k, a_k), \tag{7.33}$$

with $A \in \mathbb{R}^{n\times n}$ *and*

$$\tilde{B}(x_k, a_k) = \left(\int_0^1 \frac{\partial \Phi}{\partial x}\left(f(x_k, 0) + \zeta \tilde{f}(x_k, a_k) \right) \mathrm{d}\zeta \right) \tilde{f}(x_k, a_k). \tag{7.34}$$

Under the same condition as above, it is possible to define

$$B(x_k, a_k) = \int_0^1 \frac{\partial \tilde{B}}{\partial a}(x_k, \zeta a_k)\, \mathrm{d}\zeta \tag{7.35}$$

to obtain

$$\Phi(x_{k+1}) = A\Phi(x_k) + B(x_k, a_k)\, a_k \tag{7.36}$$

The integral in (7.31) provides a factorisation of $\tilde{B}$ such that the system can be expressed with a linear dependence on the input - however in this formulation also the input matrix is input-dependent.
This exact characterisation relies on the fact that exists a finite-dimensional space which is invariant under the action of the Koopman operator, i.e., that there is an invariant space in which the Koopman operator is finite-dimensional, as in Example 2.1.1, which is in general a rather strict assumption. As understood from Chapter 4, this is instead a necessary assumption when learning the Koopman from actual data. Also, the retrieved model is linear but the input matrix depends both on the state and the control input, making the derivation of the Optimal Control no longer straightforward. Therefore, the most common practical implementation nowadays rely on other methods, which provide a simpler description of the controlled system, so that the optimal input is readily available.
The main point of this chapter is to understand how to merge the control scenario with the identification of the Koopman operator. As has also been done for autonomous systems, the problem is formalised solely in the discrete-time perspective: this is because the estimation needs to rely on measurements of the systems, so that the discrete-time perspective is more natural.
Hence, from now on, only the discrete-time perspective is addressed.

7.5 Linear approximations

One of the earliest papers dealing with the control problem via the Koopman operator linearisation is given by Brunton et al., 2016. There it is explicitly stated that the Optimal Control law can be computed for the Koopman linearisation with minor modifications. In particular, assuming that the d components of the state are included in the basis functions for the observable space Ψ, and considering that they are in the first d position without loss of generality, the cost matrices just need to be modified as:

$$\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}, \qquad \tilde{Q}_{\mathrm{f}} = \begin{bmatrix} Q_{\mathrm{f}} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \tilde{R} = R. \tag{7.37}$$

so that the cost yields the same expression also in the lifted space.
The latter paper relies on an earlier work by the same authors (Brunton, Proctor, and Kutz, 2016) for the reconstruction the system. The algorithm presented there is very similar to the EDMD approach, although a regularisation penalty on the regression problem is added. This is done to bring out the most important features, so that the linear span of these features is a meaningful invariant space for the Koopman operator

to be learnt. In particular, the feature selection is performed through the Least Absolute Shrinkage and Selection Operator (LASSO) method (Tibshirani, 1996; Tibshirani, 1997; Shalev-Shwartz and Ben-David, 2014).
Then, through the closed-form expression for the optimal controller in the linear setting, when using the extended matrices in 7.37, the minimiser for the cost functional is obtained. Notice that the controller thus computed, is linear in the lifted space, but if there are nonlinear observables, it may include nonlinear terms in the original state. The latter approach is shown to produce a better controller than the classical linearisation method around a fixed point.
However, there are limitations to the application of this technique. In particular, the linearisation through Koopman operator may result in an unstable and uncontrollable system, which makes the design of the controller less straightforward or even impossible. Also, in the latter specific paper, only the control problem is addressed, and it is detached from the estimation procedure. Indeed the linearisation is assumed to be given, while solely the design of the input to control the model is discussed.
The first explicit integration of the control formulation into an identification algorithm exploiting the Koopman framework is given by Proctor, Brunton, and Kutz, 2016, where an extended linear model which takes into account also the input variable, as given in (7.21). In particular, the problem can be addressed by the original DMD algorithm if the input matrix B is known.
If the DMD procedure search for a matrix $A \in \mathbb{R}^{d\times d}$ so that:

$$\overline{y} = A\overline{x}, \tag{7.38}$$

given the observed control inputs as $\overline{a} = \begin{bmatrix}\overline{a}_1 & \dots & \overline{a}_m\end{bmatrix}$, the linear control model requires to find $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times p}$ such that:

$$\overline{y} = A\overline{x} + B\overline{a}. \tag{7.39}$$

If the matrix B is known, this problem can be addressed in the same way as DMD. Indeed by defining

$$\overline{\overline{y}} := \overline{y} - B\overline{a}, \tag{7.40}$$

the regression problem becomes

$$\overline{\overline{y}} = A\overline{x}, \tag{7.41}$$

which indeed is solved by Dynamic Mode Decomposition, as seen in Chapter 4, Section 4.4.
Instead when B is unknown, it is necessary to learn a rectangular matrix $U \in \mathbb{R}^{d\times(d+p)}$,

which corresponds to the finite-dimensional approximation of the Koopman operator with control, that should approximate the relation:

$$\overline{y} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \overline{x} \\ \overline{a} \end{bmatrix} = U \begin{bmatrix} \overline{x} \\ \overline{a} \end{bmatrix}, \tag{7.42}$$

which however can still be solved by Least Squares minimisation. As done for the original DMD algorithm, also in this framework an SVD decomposition can be exploited to deal with the problem in a lower-dimensional subspace, with some additional refinements resulting from the control formulation. The truncation of the SVD decomposition plays the same role of a regularisation penalty.

The latter approach refers to the definition of the Koopman operator as in (7.19), in which the control is not dynamically propagated, but only the next state is given as output, as a consequence of the chosen input. This is extended and generalised by Proctor, Brunton, and Kutz, 2018, where the prediction of the dynamics of the input alone is also considered, so that the matrix approximating the Koopman operator is always square, i.e., $U \in \mathbb{R}^{(d+p)\times(d+p)}$. The extended space of states and inputs yields the following dynamical model:

$$\begin{bmatrix} x_{k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} U_{11} & U_{21} \\ U_{12} & U_{22} \end{bmatrix} \begin{bmatrix} x_k \\ a_k \end{bmatrix}. \tag{7.43}$$

It is precisely in the work by Proctor, Brunton, and Kutz, 2018 that it is made clear that Definition 7.2.2 - which formalises the Koopman operator for controlled system - is a general case and includes different options, by characterising the input with some dynamics of its own. The latter view comprises the one given by 7.42, by considering the case in which the prediction for the input alone is always zero. This is equivalent to consider just the state space as output space for prediction, returning the method developed by Proctor, Brunton, and Kutz, 2016.

Therefore the *Koopman with Inputs and Control* (KIC) method proposed by Proctor, Brunton, and Kutz, 2018 is inherently more general. In the paper is also explained how the output space can be tuned to be any space spanned by a subset of eigenfunctions, that may depend only on the state, on the control, or both.

As it happened for the autonomous case, the results above have been extended in the same way EDMD extended DMD. Williams et al., 2016 considered again a dictionary of observables in order to approximate the autonomous dynamics, yielding the same matrices $D_{\overline{x}}$ and $D_{\overline{y}}$ already mentioned in previous chapters. In their framework, however, a parameter-dependent matrix $\tilde{U}(a) \in \mathbb{R}^{N\times N}$ is taken into account. Recalling the regression

problem of EDMD detailed in (6.14) (see Chapter 6, Section 6.3) and the definition of the Frobenius norm, the extended control version is given by:

$$\min_{\tilde{U}} \frac{1}{2} \sum_{m=1}^{M} \left\| D_{\overline{y}}^{(m*)} - D_{\overline{x}}^{(m*)} \tilde{U} \left(\overline{a}^{(m*)} \right) \right\|^2 \tag{7.44}$$

where $A^{(m*)}$ denotes the m-th row of the matrix A. It is clear from the form of the minimisation above that the inputs are treated as time-varying system parameters and not as control parameters. The parameter-dependent matrix $\tilde{U}(a)$ is further parameterised by a set of N_{A} basis functions $\tilde{\psi} : A \to \mathbb{R}$ as:

$$\tilde{U}(a) = \sum_{n=1}^{N_{\mathrm{A}}} \tilde{\psi}_n(a) \tilde{U}_n \tag{7.45}$$

where U_n are the coefficients of the expansion. This is basically the same technique exploited by EDMD in the autonomous case in order to turn the infinite-dimensional space of observables Ψ into a finite-dimensional one Ψ_D. The basis functions acting on inputs should be chosen in such a way that they are able to provide a good approximation of the control input behaviour. Both the basis function for the control and the ones approximating the autonomous dynamics are user-defined, as originally prescribed by the EDMD method.
Rewriting the minimisation in (7.46) with the decomposition of $\tilde{U}$ given in (7.45), it is possible to obtain:

$$\min_{\tilde{U}} \frac{1}{2} \sum_{m=1}^{M} \left\| D_{\overline{y}}^{(m*)} - \sum_{n=1}^{N_{\mathrm{A}}} \tilde{\psi}_n \left(\overline{a}^{(m*)} \right) D_{\overline{x}}^{(m*)} \tilde{U}_n \right\|^2 . \tag{7.46}$$

The latter can be better visualised by defining

$$D_{\overline{x}\overline{a}}^{(m*)} := \left[\tilde{\psi}_1 \left(\overline{a}^{(m*)} \right) D_{\overline{x}}^{(m*)} \quad \tilde{\psi}_2 \left(\overline{a}^{(m*)} \right) D_{\overline{x}}^{(m*)} \quad \cdots \quad \tilde{\psi}_{N_{\mathrm{A}}} \left(\overline{a}^{(m*)} \right) D_{\overline{x}}^{(m*)} \right] \tag{7.47}$$

and

$$U := \begin{bmatrix} \tilde{U}_1 \\ \tilde{U}_2 \\ \vdots \\ \tilde{U}_{N_{\mathrm{A}}} \end{bmatrix}, \tag{7.48}$$

so that the summation can be turned into an inner product.
The problem in (7.46) is then simplified into

$$\min_{U} \frac{1}{2} \|D_{\overline{y}} - D_{\overline{xa}} U\|_F^2 , \tag{7.49}$$

which makes clear the analogy with the EDMD approach. The solution is indeed given by:

$$U = D_{\overline{xa}}^{\dagger} D_{\overline{y}}. \tag{7.50}$$

Once U has been obtained, and consequently also every $\tilde{U}_i$, it is possible to compute $\tilde{U}$ for any given value of the control input a. This yields the sought finite-dimensional approximation of the Koopman operator, corresponding to that input value.
It should be noted that the approach by Williams et al., 2016 is considering a linear parameter-varying model, where the input representation as basis functions is detached from the state representation, unlike the method proposed by Iacob, Tóth, and Schoukens, 2022. However, also in this case, the obtained model is not truly LTI, therefore the design of the optimal controller is not straightforward, and in particular is not given by the well-known closed form.
In the next chapter, a new approach for approximating nonlinear system with control inputs is proposed, which results from the union of the Koopman operator framework and kernel methods as seen in Chapter 6. Although also the proposed method will yield a parameterised description of the system, based on the very common assumption of policy parameterisation, the derivation of the Optimal Control will be almost trivial, following an iterative method. The algorithm envisages indeed the use of a gradient descent procedure, in which the gradient of the value-function is directly provided. While the approach does not exploit the explicit formula for the optimal controller, it is guaranteed by Fazel et al., 2018; Sun and Fazel, 2021; Bhandari and Russo, 2021; Zhang et al., 2020 that, under technical conditions, policy gradient methods converges to the optimal solution. For example, in the case of linear systems, if the admissible class of policy which is chosen for the problem includes also the controller in (7.25), the latter papers guarantee that the solution of the proposed method converges to the Optimal Control.

8

Koopman Policy Gradient

This is the core chapter of this work, as it presents the main algorithm to tackle the Reinforcement Learning paradigm through the estimation of a parameter-dependent Koopman operator. This is achieved through a Policy Gradient algorithm in the space of policy parameters.
After the introduction of the Reinforcement Learning setting and a remark on the importance of the value-function, the first important result of the Chapter is presented, which is precisely the description of the value-function through the Koopman operator. From the latter characterisation, and exploiting the previously derived algorithm to learn Koopman operators through kernel methods, the main approach is presented, which consist on the extension for controlled dynamical systems. The final algorithm operates in an online fashion, iteratively updating the control parameter every time a new datum is available.
The contents of this chapter are based on the work by **Zanini, Francesco** and Chiuso, 2021a, and therefore they constitute original contributions.

8.1 Reinforcement Learning

The *Reinforcement Learning* framework basically pursues the same objective as the *Optimal Control* problem presented in Chapter 7, with slight changes in the setting and definitions (Recht, 2018). The usual scenario depicts an agent capable of performing actions within an unknown environment, whose goal is to figure out how to maximise a reward function. This is accomplished by trying different actions and logging the response and feedback given by the environment.
The latter framework of sequential decision making is formalised in Sutton and Barto, 2018 as a Markov Decision Process $\langle X, A, P, r, \gamma \rangle$, which consists of a state-space X, an action-space A, a transition probability $P(\cdot \mid \cdot)$, a reward function $r(\cdot)$ and a discount

factor γ (see also Kaelbling, Littman, and Moore, 1996; Szepesvári, 2010). The goal of the agent is to discover a control policy $\pi : X \to A$, which is a mapping from states to actions, so that the performances - as measured by the reward function - are maximised over the evolution of the system.
The dynamics are expressed through the transition probability model, which governs the behaviour of the system, and plays the same role as the vector field on a discrete-time system. In general the transition probability yields the conditional joint distribution of the reward and the next state, given the current state and action, as:

$$P\left(x', r \mid x, a\right) = \mathbb{P}\left(\underline{x}' = x', \underline{r} = r \mid \underline{x} = x, \underline{a} = a\right), \tag{8.1}$$

where in the equation above the underlined letters denote random variables and the standard lower case letters a value in their respective image. Oftentimes, when time does not play any role, it is customary to rely on the notation with the apex in order to indicate the next state x' with respect to x, and the state x'' following x'. The latter is used also here and in the next chapter.
The performance over the trajectory is given by the *return*, which is the sum, weighted by the discount factor, of the rewards over subsequent transitions. The latter is clearly a random variable, which also depends on the action taken at each step, and on the initial state. The standard setting in Reinforcement Learning is to consider its expectation as the ultimate measure of performance.
In this regard a key quantity in Reinforcement Learning is the *value-function*, which is the *expected return*, with respect to the transition probability, when starting in a certain state x and following a specific policy π thereafter. The latter is given by:

$$V_{\pi}\left(x\right) = \mathbb{E}_{\pi}\left[\sum_{\tau=\tau_0+1}^{\overline{\tau}} \gamma^{\tau-\tau_0-1} \underline{r}_{\tau}\right], \tag{8.2}$$

for a system which is in state x at time τ_0. The expectation is taken with respect to the transition probability induced by the policy π. The reward $\underline{r}_{\tau}\left(\cdot\right)$ with subscript τ is just the same reward function given by a transition τ steps in the future. In general the reward function may depend on the triplet given by the current state, the action, and the next state.

Note that for any state x and policy π the value-function in (8.2) can be rewritten as:

$$V_\pi(x) = \int_A \pi(a \mid x) \int_{X\times\mathbb{R}} P(x', r \mid x, a) \left[r + \gamma \mathbb{E}_\pi \left[\sum_{\tau=\tau_0+2}^{\bar{\tau}} \gamma^{\tau-\tau_0-2} \underline{r}_\tau \right] \right] \mathrm{d}(x' \times r)\, \mathrm{d}a \tag{8.3}$$

$$= \int_A \pi(a \mid x) \int_{X\times\mathbb{R}} P(x', r \mid x, a) \left[r + \gamma V_\pi(x') \right] \mathrm{d}(x' \times r)\, \mathrm{d}a, \tag{8.4}$$

which implies a recursion, being dependent on the value-function at the next state. The latter expression is the integral form of the Hamilton–Jacobi–Bellman equation.

The above formulations (8.2) and (8.4) can be related to a given initial state-distribution, instead of a specific initial state. This is a trivial generalisation which requires to consider a further integral over the density of the initial state. In particular, the exact expressions (8.2) and (8.4) are recovered for a δ-distribution centred in x. The latter is the simplest case where the initial state is assumed to be given.

The value-function is a measure of how good the feedback law is, in expectation, in terms of the generated return: this induces a partial ordering among policies, so that a better policy yields a higher value-function, for the same initial state-distribution.

The objective of the Reinforcement Learning algorithm is then ultimately to find the policy π^* attaining the highest score for the value-function, given an initial state-distribution $\rho(x)$, i.e.

$$\pi^*(\rho) \in \arg\max_{\pi\in\Pi} \int V_\pi(x)\rho(x)\, \mathrm{d}x, \tag{8.5}$$

where the optimisation is considered over a space of admissible policies Π, and the solution may not be unique. It can be noted also how in the Optimal Control framework the same task is being solved, where a minimisation over a functional is implied, subject to dynamics, as in (7.3) (see Chapter 7).

The latter equation makes it clear why the value-function is such an important quantity in Reinforcement Learning. Given the knowledge of the value-function for each state, or for the fixed initial state-distribution, the Reinforcement Learning problem reduces to the maximisation in (8.5). However this quantity is not readily available, but must be *computed* or *estimated*.

Conveniently, equation (8.4) provides a formal recursive method for computing the value-function, in which its value at the next state can be safely substituted with an estimate. As the equation keeps being iterated providing the new estimate, the reconstruction of the value function converges to its true value.

The latter approach works provided that the transition probabilities and the reward func-

tion are known, so that computations are feasible, which is a scenario usually pertaining to the field of Dynamic Programming. Within the literature related to the latter field, the process of computing the value-function for a certain policy is termed *policy evaluation*. This step is indeed required to understand the performances of the selected policy. When the number of states is *finite*, through the Bellman equation it is possible to compute the exact value-function. Since the true value-function V_π is the only fixed-point of (8.4), a standard argument by Bertsekas and Tsitsiklis, 1989, relying on the *Banach-Caccioppoli* theorem (Banach, 1922; Caccioppoli, 1932; Rudin, 1953), guarantees the convergence to the true function, for any initial estimate.

However, when the the state-space is *infinite*, approximate solution are necessary, if no prior knowledge on the value-function is available. Furthermore, in the usual Reinforcement Learning setting, the transition probabilities are not known, which makes it mandatory to resort to approximations. The same name of policy evaluation is used in the case that an approximation is sought, and not the exact computation.

Besides understanding the performances of a policy through the computation or the approximation of the value function, another essential task is to find a policy that performs better than the current one, or eventually to establish optimality. This phase takes the name of *policy improvement* (Bellman, 1957; Watkins, 1989).

Many successful algorithms perform sequentially a variable number of Bellman updates for policy evaluation and a some steps of policy improvement (Howard, 1960), which go under the name of *Generalised Policy Iteration* (Puterman and Shin, 1978; Kaelbling, Littman, and Moore, 1996). A notable example is given by the *value iteration* algorithm (Bertsekas, 1987), which alternates a single step for policy evaluation and one of policy improvement.

A simple but effective idea for managing the policy improvement task is given by *Policy Gradient* methods (Sutton and Barto, 2018). This class of techniques is usually employed when a continuous space for states and actions is considered, because many other approaches are found to be intractable (Williams, 1988; Williams, 1992). Under the assumption of a parameterised class of policies, the optimisation can be performed through standard gradient ascent with respect to the value-function. Although this framework is simple, the gradient of the value-function with respect to the parameter characterising the policy is not given, and must be estimated. The latter is in general a difficult task, however the Koopman formulation of the value-function given in Chapter 6 provides a simple way to retrieve the sought gradient direction, and consequently to iteratively find the optimal policy.

8.2 The reward as an observable

Discrete-time dynamical systems are defined through μ-measurable mappings acting on a measurable space X. In particular, X is a measurable space, μ is a measure, and $\mathcal{T}$ is a measurable map from X to itself which preserves the measure. The latter property means that the system is assumed to be measure-preserving.
Stochasticity is embedded into the dynamics through a transition density function $\mathcal{T} : X \times X \to \mathbb{R}^+$, so that the transition probability for every set $\tilde{X}$ in the σ-algebra $\mathfrak{S}$ is given by:

$$\mathbb{P}\left(f\left(x\right) \in \tilde{X}\right) = \int_{\tilde{X}} T\left(x, y\right) \mathrm{d}\mu\left(y\right), \tag{8.6}$$

$\forall \tilde{X} \in \mathfrak{S}$.
The Koopman operator for stochastic dynamical systems characterises the expectation of the observable given as input, with respect to the transition probability function, as described in Chapter 2, Section 2.5 and Chapter 4, Section 4.2. This means that the operator is given by:

$$\mathcal{U}\left[\psi\right]\left(x\right) = \mathbb{E}\left[\psi\left(f\left(x\right)\right)\right] = \int_{\tilde{X}} \mathcal{T}\left(x, y\right) \psi\left(y\right) \mathrm{d}\mu\left(y\right). \tag{8.7}$$

In what follows, the Reinforcement Learning problem is addressed through a new procedure for estimating the value function. In particular, both the value-function and its gradient are reconstructed from measurement of the environment dynamics, stored in transitions pairs, and knowledge of the reward function. The Koopman operator will prove itself a valuable tool for this objective, and even more so when formulated in its kernel version (as presented in Chapter 6).
The latter framework is particularly suited to this task because it allows to specify the dictionary of functions, required for approximating the Koopman operator in a finite-dimensional space, through the definition of the kernel function. This is helpful because it is possible to embed meaningful prior knowledge in the function hypothesis space, according to the system that should be characterised.
The main idea of this chapter is then to learn the Koopman operator associated with the underlying dynamics, in order to exploit the function propagation perspective, to estimate the reward for future states. Since the latter is the only quantity of interest in the Reinforcement Learning setting, modelling its behaviour over time is a crucial step. In order to do that, the reward function will be treated as an observable, by relying on the following assumption.

Assumption 8.2.1. The reward is a deterministic function of the next state only, i.e.,

when transitioning from state $x \in X$ with action $a \in A$, the reward will depend solely on the state that is reached x', although the transition may still be stochastic.

The latter is a simplifying assumption which makes the derivation of the next propositions simpler, as well as the exploitation of the Koopman framework into the Reinforcement Learning setting. In particular, the estimation of the value-function will follow naturally from a rewriting of the Bellman equation. The latter assumption is actually not strictly required, but the relaxation comes at the price of a more cumbersome description of the procedure, which essentially adds nothing of relevance.
It is also worth noticing that Assumption 8.2.1 is very mild, since the reward (or cost) function is usually a user-defined quantity which is there to specify the goal of the learning agent (Sutton and Barto, 2018). This is especially true in the control setting. As evidence, recently there has been an increase interest in the task of reward design, which would be meaningless unless the reward could be chosen (Eschmann, 2021; Devidze et al., 2021; Ryan and Deci, 2000; Barto, Singh, and Chentanez, 2004; Schmidhuber, 1991). And when it is not possible, and it is embedded in the system, in many cases it actually depends only on the arrival state of the transition.
Also, since the systems dynamics are allowed to be stochastic, there is no real need to consider a stochastic function as a reward: if the stochasticity is caused by the state or the action, then it can be thought as part of the transition, and it will be addressed through the Koopman operator framework. If it is independent from both the state and the action, and it is just given by randomness in the reward function, it is useless to the Reinforcement Learning objective. Indeed policies will be ordered through the value-function, which involves an expectation, so that the same ordering would result from considering directly the expectation of the reward, which is a deterministic function.
Finally, the requirement in Assumption 8.2.1 that the reward must be dependent only on the next state is only there to make the derivation in the next section easier. Indeed for the general case it would be sufficient to define an augmented dynamical system, whose dynamics are autonomous and they propagate the triplet given by the current state, the action, and the next state. In this scenario a reward which is only dependent on the next state in the enlarged system would correspond to a reward depending on the whole triplet in the original one. As a proof of this, the theoretical derivation is carried out by relying on Assumption 8.2.1, but in the illustrative example the latter is violated, and the reward, on the contrary, will depend on the current state and on the action, but not on the next state. A theoretical derivation for this case can be found in the paper by **Zanini, Francesco** and Chiuso, 2021a, from which this chapter has been drawn.

8.3 Koopman formulation of the value-function

The continuous-time setting for Reinforcement Learning is very challenging, since no algorithm can rely on the possibility of visiting the whole state-action space, even with infinite amount of data. Therefore approximation techniques are mandatory (Sutton and Barto, 2018; Hasselt, 2012; Hasselt and Wiering, 2007; Lazaric, Restelli, and Bonarini, 2007; Gaskett, Wettergreen, and Zelinsky, 1999; Montazeri, Moradi, and Safabakhsh, 2011). As a further difficulty, the space of admissible policies is infinite-dimensional, being a space of functions, so that the problem is ill-posed when relying only on sample data - as explained in Chapter 5 - although it is still well-defined if the knowledge of the transition function is fully available.
In the literature (Sutton et al., 1999; Jonathan and Peter, 1999; Greensmith, Bartlett, and Baxter, 2001; Ng and Jordan, 2000; Kakade, 2001; Silver et al., 2014) the latter problem is often bypassed through the restriction to a parameterised class of policies, so that the controller is uniquely determined by a parameter $\theta \in \Theta \subset \mathbb{R}^l$. The parameterisation and the space of parameters Θ determine the set of policies that will be considered for the optimisation problem. The latter is true both for the case of deterministic policies, and stochastic ones. In the more general hypothesis of a stochastic policy, the distribution over action can be written as:

$$a_k \sim \pi_\theta (x_k) \tag{8.8}$$

where $\theta \in \Theta$ is the parameter identifying the policy. Clearly the search space of the Reinforcement Learning framework is now constrained to be the subset of policies Π_Θ which can be described by an element in the parameter space.
Note that the assumption above allows to consider a controlled dynamical system as an autonomous parameterised system. This is reflected in the following equation:

$$\begin{aligned} x_{k+1} &= f (x_k, a_k) && (8.9)\\ &= f (x_k, \pi_\theta (x_k)) && (8.10)\\ &= f_\theta (x_k) && (8.11) \end{aligned}$$

which makes it easier to address the control problem.
By restricting the space of admissible policies to Π_Θ, also the value-function can be thought in terms of the policy parameter θ, given that there is a one-to-one mapping. This leads to a characterisation of the value-function geometry in the space of parameters, which allows to solve the problem in a simple way. Indeed if the value-function turns out to be continuous with respect to the policy parameter θ, the gradient of the latter

function gives a way to improve the same policy parameter, as the final objective is directly considered. A *gradient ascent* procedure it is then guaranteed to find at least a local maximum value for the value-function, and more importantly, a locally optimal parameter. Nonetheless, under some technical conditions given by Fazel et al., 2018; Sun and Fazel, 2021; Bhandari and Russo, 2021; Zhang et al., 2020, which are related to the gradient dominance property of the system under analysis, the convergence to the global minimiser is guaranteed. Another fundamental requirement for the optimality is of course that the optimal controller belongs to the class of admissible policies resulting from the parameterisation.

This is exactly what is prescribed by policy gradient methods. The objective is then to find the optimal parameter θ^* in the parameter space, associated with the policy in Π_θ for which the value-function attains the highest score, i.e.,

$$\theta^* = \arg\max_{\theta\in\Theta} \{V_\theta(x)\} = \arg\max_{\theta\in\Theta} \left\{ \mathbb{E}_\theta \left[\sum_{\tau=1}^{\overline{\tau}} \gamma^{\tau-1} r(x_\tau) \mid x_0 = x \right] \right\}, \tag{8.12}$$

in which the expectation is taken with respect to the transition probability of the dynamics induced by the parameter θ, the starting time is considered to be $\tau_0 = 0$, and the horizon $\overline{\tau}$ may as well be infinite.

Different control parameters induce different transitions probabilities, which in turn gives different values for the expectation. From (8.12) it is clear that the value-function can be interpreted as the static reward function applied to the state, which is evolving over time through the controlled dynamical system. This is exactly the action of the Koopman operator describing the controlled dynamical system induced by the parameter θ, when given as input the reward function $r(\cdot)$.

Note that by restricting the admissible policies to be of the form in equation (8.8), i.e., a feedback from the state parameterised by θ, the dynamical system can be thought as an autonomous system, whose trajectories are parameterised by the same control parameter θ, as in (8.11). Indeed the only dependence in the dynamics would be on the state alone. The latter is a crucial observation which will allow the extension of the estimation procedure presented in Chapter 6 to the control setting. In particular, a parameter-dependent Koopman operator will be estimated, in order to take into account the effect of the control and to find an local optimum in the space of parameters.

Also, given that the value-function is often the only quantity of interest for many algorithms, the fact that it can be exactly described through the action of the Koopman operator can open up several directions for new methods that may be characterised by different properties than those derived from the state-space view; as it has been done so

far with classical properties of the controlled systems (see Chapter 4).
The following proposition provides a rigorous formalisation of the idea that the value-function can be completely characterised by the Koopman operator applied to the reward function.

Proposition 8.3.1. *Let Assumption 8.2.1 hold and let $\theta \in \Theta$ be the parameter characterising the policy, then the value-function V_θ associated to a Markov Decision Process $\langle X, A, P, r, \gamma \rangle$ with horizon $\overline{\tau}$, can be expressed by iterated application of the Koopman operator $\mathcal{U}_\theta$ associated with a measure-preserving dynamical system to the reward function, as:*

$$V_\theta(x) = \sum_{\tau=1}^{\overline{\tau}} \gamma^{\tau-1} \mathcal{U}_\theta^\tau [r(x)] \tag{8.13}$$

$\forall x \in X$, $\forall \theta \in \Theta$.

Proof. The rationale behind the proof is to define a transition probability function $\mathcal{T}(\cdot, \cdot)$ to be associated with the stochastic Koopman operator, so that the latter description becomes equivalent to the expectation in the value-function. Once that is achieved, the claim trivially descend from the definition of the value-function and the Koopman operator.
Recall that the value-function is written as

$$V_\theta(x) = \mathbb{E}_{x' \sim \mathbb{P}(\cdot \mid x,\, a=\pi_\theta(x))} \left[r(x') + V_\theta(x') \right], \tag{8.14}$$

which gives rise to the following chain of expectations

$$V_\theta(x) = \mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x,\, a=\pi_\theta(x))} \left[r(x') + \mathbb{E}_{x'' \sim \mathbb{P}_\theta(\cdot \mid x',\, a=\pi_\theta(x'))} \left[r(x'') + V_\theta(x'',\, a = \pi_\theta(x'')) \right] \right] \tag{8.15}$$

$$= \mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x,\, a=\pi_\theta(x))} \left[r(x') + \mathbb{E}_{x'' \sim \mathbb{P}_\theta(\cdot \mid x',\, a=\pi_\theta(x'))} \left[r(x'') + \mathbb{E}_{x''' \sim \mathbb{P}_\theta(\cdot \mid x'',\, a=\pi_\theta(x''))} [\ldots] \right] \right]. \tag{8.16}$$

By focusing on the the very first step, it is easy to see that (8.14) corresponds to the Hamilton-Jacobi-Bellman equation in (8.4). It is of paramount importance to understand the first term, as the following ones can be managed in the same way through the unrolling of the value-function given in (8.16).
Since the case of deterministic policies is straightforward and easier to understand, the proof is split according to the nature of the policies considered.

- *Deterministic policies.*
  In this case the policy is a deterministic function of the state. Analogously the

policy can be thought as a δ-distribution centred in $\tilde{\pi}_\theta(x)$ which is a deterministic function of the state.
Define the distribution over the next states given the current state and the control parameter θ as

$$\mathbb{P}_\theta(\cdot \mid x) := \mathbb{P}(\cdot \mid x, a = \tilde{\pi}_\theta(x)); \tag{8.17}$$

so that the Bellman equation becomes

$$\begin{aligned} V_\theta(x) &= \mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x)} \left[r(x') + V_\theta(x') \right] && (8.18) \\ &= \mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x)} \left[r(x') \right] + \mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x)} \left[V_\theta(x') \right] && (8.19) \end{aligned}$$

The first term is given by the expectation of the reward function applied to the next state, which should be handled by the stochastic Koopman operator given in (8.7). This can be done by considering the transition density function that characterise a particular dynamical system, i.e., the one giving the same distribution over the next state as (8.17). Therefore, defining a parameterised transition as

$$\mathcal{T}_\theta(x, \cdot) = \mathbb{P}_\theta(\cdot \mid x), \tag{8.20}$$

it yields $f(x) = \mathbb{P}_\theta(\cdot \mid x)$, for the stochastic vector field of the underlying system. The latter represents the autonomous stochastic dynamics which are obtained for the specific parameter θ.
Hence the expectation of the current reward in (8.19) can be rewritten as

$$\begin{aligned} \mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x)} \left[r(x') \right] &= \mathbb{E}_{f(x) \sim \mathbb{P}_\theta(\cdot \mid x)} \left[r(f(x)) \right] && (8.21) \\ &= \int r(f(x)) \, \mathcal{T}_\theta(x, f(x)) \, \mathrm{d}\mu(f(x)) && (8.22) \\ &= \mathcal{U}_\theta \left[r(x) \right], && (8.23) \end{aligned}$$

which, as expected, only makes use of the Koopman operator, with additional dependence on the parameter θ.

- *Stochastic policies.*
  When dealing with stochastic policies, the transition should also take into account the uncertainty resulting from the randomness on the action selection, given the current state. In this regard, define the joint distribution of the action and the next state, conditioned on the current state, as

$$(a, x') \sim \tilde{\mathbb{P}}_\theta((\cdot, \cdot) \mid x) \tag{8.24}$$

for which it must hold that

$$\pi_\theta(x) = \int_X \tilde{\mathbb{P}}_\theta\left((a, x') \mid x\right) \mathrm{d}\mu(x'). \tag{8.25}$$

Then, marginalising over the action distribution for a specific value of the control parameter θ, it yields

$$\mathcal{T}_\theta(x, \cdot) = \int_A \tilde{\mathbb{P}}_\theta\left((a, x') \mid x\right) \mathrm{d}a \tag{8.26}$$

which is the sought distribution for matching the Koopman operator description and the expectation over the probabilities of the MDP. The same characterisation of the expectation of current reward as the Koopman operator applied to the reward function holds also for stochastic policies.

The construction above gives the transition probability for the first step in the chain of expectations of the Bellman equation given in (8.16). However, repeating the same reasoning with the transition density function for the second step it is easy to see that:

$$\begin{aligned}
&\mathbb{E}_{x' \sim \mathbb{P}_\theta(\cdot \mid x)}\left[\mathbb{E}_{x'' \sim \mathbb{P}_\theta(\cdot \mid x')}\left[r\left(x''\right)\right]\right] = \\
&= \int\int r\left(f\left(f\left(x\right)\right)\right) \mathcal{T}_\theta\left(f\left(x\right), f\left(f\left(x\right)\right)\right) \mathrm{d}\mu\left(f\left(f\left(x\right)\right)\right) \mathcal{T}_\theta\left(x, f\left(x\right)\right) \mathrm{d}\mu\left(f\left(x\right)\right) && (8.27)\\
&\qquad = \mathcal{U}_\theta\left[\int r\left(f\left(x\right)\right) \mathcal{T}_\theta\left(x, f\left(x\right)\right) \mathrm{d}\mu\left(f\left(x\right)\right)\right] && (8.28)\\
&\qquad = \mathcal{U}_\theta\left[\mathcal{U}_\theta\left[r\left(x\right)\right]\right], && (8.29)
\end{aligned}$$

which shows that the expected reward 2 steps ahead can be described by considering the Koopman operator twice. By iterating the same reasoning for all the steps $\overline{\tau}$ which affect the computation of the value-function, the following expression can be derived:

$$V_\theta(x) = \sum_{\tau=1}^{\overline{\tau}} \gamma^{\tau-1} \mathcal{U}_\theta^\tau\left[r\left(x\right)\right] \tag{8.30}$$

where $\mathcal{U}_\theta^\tau$ indicates τ subsequent applications of the same operator $\mathcal{U}_\theta$. □

Note that in the previous formulations the dependence of the Koopman operator on the control parameter has been made explicit with the subscript notation. If θ is kept fixed, the state evolves according to the dynamical system induced by the associated control policy, therefore it can be considered as an autonomous system, with transition density function given by (8.26), in the general case. By changing the control parameter, and hence the policy, the dynamical system varies as well. Therefore the Koopman

operator itself will depend on the control parameter.
As it has already made clear by Korda and Mezić, 2018, and in Chapter 7, the Koopman operator can be parameterised with actions, so that it returns the evolution of the state when a specific action is taken, regardless of the state. The latter case corresponds to a trivial parameterisation of control policies, in which the action - or the distribution of actions, if the policy is stochastic - does not depend on the state. The latter is actually an example of policy parameterisation in which $\Theta = A$, although often not a good one, since the space of admissible policies Π_Θ would be composed only of constant policies, which for many systems do not yield satisfactory performances for the controlled dynamics. Recall that the latter is an important issue in order to converge to the optimal controller, since a trivial necessary conditions to achieve optimality is that the optimal policy belongs to the search space Π_Θ. Moreover, by considering a continuous space of actions, it would be unfeasible to learn the transition for all actions, or all parameters.
Therefore, 2 important issues emerges which need to be addressed in order to successfully deal with parameterised control systems, and in particular to frame the setting in such a way that the Reinforcement Learning problem can be successfully solved: the *choice of the parameterisation*, and how to *generalise among parameters*. For both matters, kernel methods represent a viable solution, which makes the approach presented in Chapter 6 even more remarkable, as it is suitable for addressing the control problem with only a few modifications.
The choice of a suitable parameterisation for the control policy is actually part of the setting in which the objective is defined and does not pertain to the learning algorithm. The latter defines the problem to be solved, which is the maximisation of the value-function over the class of *admissible policies*. However, through the proposed approach, the explicit formulation in (8.8) is never exploited, and the regression is framed directly in the space of parameters. This gives complete freedom of choice for the parameterisation, so that there are no limitation on the class of policies which can be employed in order to tackle the Reinforcement Learning problem. This can be accomplished by formulating the problem with parameter-dependent observables in a Reproducing Kernel Hilbert Space. This will be explained in detail in Section 8.4.
As far as generalisation over parameters is concerned, a very simple idea leads again to the use of kernel methods, because of their interpretations as similarity measures, which can be exploited to generalise over parameters.
To understand this, consider the parameterisation of constant policies, given by the choice of an action that is always the same, without any dependency on the state, for which $\Theta = A$. Fixing a particular action $\overline{a} \in A$, the dynamical system can be seen as

autonomous, yielding different trajectories for different initial conditions. It is easy to realise that, starting from the same initial conditions, very similar values of the chosen action will result in very similar trajectories.
The latter turns out to be true for any parameterisation of the dynamics. In order to make more formal the latter intuition, consider a parameterised dynamical system in continuous time, given by:

$$\dot{x}(t) = f_\theta(x(t)). \tag{8.31}$$

Then its solutions starting from $x(t_0) = x_0$, given by $\Phi_\theta^{t-t_0}(x_0)$, should be similar for similar values of $\theta \in \Theta$.
The following theorem - which holds in the general case of time-varying ordinary differential equations - establishes with certainty that this is the case (Coddington and Levinson, 1955).

Theorem 8.3.2 (Continuity of solutions of ODEs on parameters)**.** *Suppose $f_\theta(x,t)$ is continuous and locally Lipschitz in x in an open set $E \subseteq \mathbb{R} \times X \times \Theta$. If $\Phi_{\bar{\theta}}^{t-t_0}(x_0)$ is a solution of the initial value problem given by*

$$\begin{cases} \dot{x}(t) = f_{\bar{\theta}}(x,t) \\ x(t_0) = x_0 \end{cases} \tag{8.32}$$

which is defined on the closed interval $[i_1, i_2]$ and $\left(t, \Phi_{\bar{\theta}}^{t-t_0}(x_0), \bar{\theta}\right) \in E$ for $t \in [i_1, i_2]$, then there is a neighbourhood N_0 of $\left[t_0, x_0, \bar{\theta}\right] \in \mathbb{R} \times X \times \Theta$ such that, for any triplet in N_0 the associated initial value problem has a solution defined on the interval $[t_0, i_2]$. Moreover, the function $\Phi_{\bar{\theta}}^{t-t_0}(x_0)$ is continuous on $[t_0, i_2] \times N_0$.

Therefore the map going from initial conditions and parameters of the initial value problem to the solution of the same problem is continuous with respect to parameters of the vector field, provided the latter is continuous and locally Lipschitz. Theorem 8.3.2 can also be generalised to stochastic differential equation, with appropriate adjustments, however this would require a more detailed elaboration on the subject, which is beyond the scope of this Dissertation. Clearly the continuity property would hold for sample paths of the solution (Mishura, Posashkova, and Posashkov, 2011; Yasinsky and Malyka, 2012; Schmelzer, 2010).
Recall that the main motivation for introducing *Mercer* kernels in Chapter 5 was to restrict the hypothesis space to continuous function, with the rationale of modelling the a priori belief that the true function has some regularity. Thanks to Theorem 8.3.2 the same idea can be safely applied also in this case, over the space of parameters. The use

of a continuous kernel function to generalise over parameters which are available as data, is a meaningful choice because the solutions are indeed known to be continuous with respect to the control parameter. As it will be further clarified in what follows, there is no significant distinction between states and parameters. Then the expectation of the solution of a stochastic differential equation will be reconstructed through kernel methods, learning both similarities among states - as done in Chapter 6 - and among control parameters, in order to address the control problem.

8.4 Koopman Policy Gradient

The main contribution of this Chapter is a way to tackle the Reinforcement Learning problem through the exploitation of the Koopman operator framework. In particular, the Koopman operator is learned as previously done in Chapter 6 and then applied to the reward function, in order to obtain an estimate of the value-function. However the setting of the control problem is different, as it is required to take into account changes in the behaviour of the system, caused by the policy. The estimation procedure must be then extended to handle controlled systems.
In lights of the reasoning above, this can be done with relative ease under the assumption that the policy is a parameterised function of the state, therefore changing the view from actions to policy parameters. Hence, instead of dealing with a regression for the new input, as it is proposed in the literature reviewed in Chapter 7, the method described below learns a Koopman operator which deals with the prediction of the next parameter. Conveniently, the extension turns out to be particularly simple, thanks to the insights provided by Theorem 8.3.2. The continuity of the solution of a differential equation allow state and control parameter to be treated equally. Indeed the procedure given in Chapter 6 exploited kernel methods to generalise over states, and the same generalisation has been verified to be also meaningful with respect to control parameters. In order to understand this better, consider a $(n-1)$-dimensional autonomous system with a scalar parameter. Through the estimation algorithm based on the Koopman operator, some kind of similarity is leveraged among states through a kernel function, in order to generalise the estimate - see Chapter 6. Following the result of Theorem 8.3.2, a similar kind of similarity measure could be enforced for the scalar parameter. By supposing that the same similarity measure is meaningful for the scalar parameter, there would be no difference between the depicted scenario and the estimation of an n-dimensional autonomous system, in terms of how to manage generalisation.
The kernel approach anyway is very general and it is capable of modelling different

similarity measures even for the same state space. Therefore, given that the kernel is chosen appropriately, the latter can induce very different generalisation over state and parameters, reflecting the user's prior belief. However, the procedure for the estimation remains unchanged.

Moreover, the prediction of parameters instead of actions is particularly suited for the problem, as in the propagations defining the value-function the control parameter is fixed, so there is no need to predict its evolution. The task of evaluating the performance of a specific policy $\overline{\pi} = \pi_{\overline{\theta}}(x_k)$ is indeed partially solved, as the control is characterised by a fixed parameter $\overline{\theta}$ which remains the same over the whole trajectory, so that the "estimation" of the parameter is trivial.

The above perspective yields the following definition for a parameter-dependent observable:

$$\mathcal{U}[\psi(x,\theta)] = \psi(f(x, a \sim \pi_\theta(x)), \theta), \tag{8.33}$$

which - as done for the case of a fixed action in (7.19) - allows the definition of a parameter dependent Koopman operator:

$$\mathcal{U}_\theta[\psi](x) = \psi(\Phi_\theta(x)). \tag{8.34}$$

In order to capture all the possible behaviours of the underlying dynamical system for the different control parameters, it would be necessary to consider a huge dictionary of functions $\{\psi_i(x,\theta)\}_{i=1}^N$, when following standard approaches to compute a finite-dimensional approximation of the Koopman operator (see Chapter 4). With the RKHS approach presented in Chapter 6 instead, it is only needed to specify the kernel function, which can be seen as a similarity measure in the state-parameter space, and the solution of the regression problem is known to lie in the linear space of kernel sections centred in the available data, thanks to the Representer theorem (Theorem 5.2.7). Therefore by considering the enlarged space given by the Cartesian product of the state space and the parameter space, $Z := X \times \Theta$, it is possible to define a kernel which helps with the estimation problem by leveraging similarity both in the state space and in the parameter space. Since the Kernel approach is very general and many different kind of kernels can be defined, this method is compatible with all kinds of prior information which may be available for the dynamics.

The proposed method gives a way to retrieve a non-parametric estimate of the value-function, in light of Proposition 8.3.1. By following the general scheme of *Policy Gradient* algorithm, a procedure to derive the optimal parameter is derived, which relies on the assumption that the kernel function is differentiable with respect to the control parameter

θ.
Consider the extended variable $z := \begin{bmatrix} x^\top & \theta^\top \end{bmatrix}^\top$ in the augmented state-parameter space Z. Snapshots of the system are iteratively collected while actively interacting with the environment, so that the setting is as close as possible to the Reinforcement Learning paradigm. Data are gathered in the form of state-parameter pairs, given by:

$$z_k = \begin{bmatrix} x_k \\ \theta_k \end{bmatrix}, \tag{8.35}$$

representing the current state and the control parameter actually used in the current transition.
The proposed method is indeed framed as an *online* procedure, as is customary in RL algorithms. In general, datapoints are organised as input-output pairs, so that if M interactions with the system has already been performed, the available data would be:

$$\overline{z} = \begin{bmatrix} z_0 & \dots & z_{M-1} \end{bmatrix}, \qquad \overline{z}' = \begin{bmatrix} z_1 & \dots & z_M \end{bmatrix} \tag{8.36}$$

in such a way that every point can be linked with its future value. This leads exactly back to the setting of Chapter 6.
Given the chosen kernel function, the reward can indeed be estimated in the same way as before, and the value-function will be simply given by the sum of the iterated propagations, as detailed below.
An estimation of the gradient of the value-function is computed starting from the M available datapoints, which is then used to update the control parameter, in order to obtain the new and improved value, θ_M. The latter is then used to derive the policy for the M-th step, which will yield a better estimation of the value function and a new gradient. Iterating this procedure, the algorithm will converge to at least a local maximum of the value function.
The policy gradient algorithm is presented in its two phases: *policy evaluation* and *policy improvement*.

8.4.1 Policy evaluation

The procedure is divided in three steps to make it easier to be followed. Note that the first step is actually not necessary (see Chapter 6, Section 6.6), however it allows to frame the procedure in a iterative fashion that is straightforward to be propagated for subsequent predictions.

- Compute α.

These are the coefficients for the kernel section centred in $\bar{z}$ which best represent the reward observable. The projection of the reward observable onto the space formed by the kernel sections is given by:

$$\hat{r}(\cdot) = K(\cdot, \bar{z})\,\alpha \tag{8.37}$$

where

$$\alpha = [K(\bar{z}, \bar{z})]^{-1} r(\bar{z}). \tag{8.38}$$

- Learn the approximated Koopman operator U.
  By exploiting again the same RKHS formulation, the latter is given by Proposition 6.5.1:
  $$U = \left[K(\bar{z}, \bar{z}) + \sigma^2\right]^{-1} K(\bar{z}', \bar{z}), \tag{8.39}$$
  and it maps coefficients as $\beta = U\alpha$, in order to approximate the reward of the next state:
  $$r_1(\cdot) := r(f(\cdot)) \simeq K(\cdot, \bar{z})\,\beta. \tag{8.40}$$

- Predict future rewards.
  The prediction of the reward τ step ahead is given by:
  $$\hat{r}_\tau(z) = K(z, \bar{z})\,\beta_\tau,$$
  where $\beta_\tau = U^\tau \alpha$, or equivalently, $\beta_\tau = U\beta_{\tau-1}$.

The procedure outlined above allows to compute an approximation of the value function as:

$$\hat{V}_\theta(x) = K(z, \bar{z}) \sum_{\tau=1}^{\overline{\tau}+1} \gamma^{\tau-1} \mathcal{U}^\tau \alpha = K(z, \bar{z}) \sum_{\tau=1}^{\overline{\tau}+1} \gamma^{\tau-1} \beta_\tau. \tag{8.41}$$

Note that the actual computation of the regression coefficients α for the first step is actually not needed, as already made clear in Chapter 6, Section 6.6. The introduction of a second regression problem is not mandatory, but it is useful to understand the meaning of the α coefficients. Indeed the coefficients for the first step β can be directly recovered as:

$$\beta = [K(\bar{z}, \bar{z}')]^{-1} r(\bar{z}'), \tag{8.42}$$

which yields the exact values for the reward function. As outlined in Remark 6.5.2, the invertibility of the kernel matrix does not require any additional assumption.
Note that by relying on Assumption 8.2.1, it is only needed one transition to have a well defined gradient. Considering indeed the starting state x_0 and the initialisation

of the control parameter θ_0, which is used to derive the policy that interacts with the environment and move the state into x_1. The estimation of the value-function and the associated gradient make use of the state-parameter pair $\bar{z}' = \begin{bmatrix} x_1 & \theta_1 \end{bmatrix}$, which however cannot be currently available, except by choosing an arbitrary value for θ_1. However, since the reward will depend only on the next state, note that in equation 8.42 the dependence on the whole augmented state is only fictitious. Therefore the algorithm can be initialised with just the first transition yielding x_0, θ_0 and x_1.
The next section will discuss the case in which Assumption 8.2.1 does not apply.

8.4.2 Policy improvement

Under the assumption that the adopted Kernel is differentiable with respect to the control parameter θ, it is trivial to notice that the gradient with respect to θ of the value-function estimated in the previous Section can be written as:

$$\nabla_\theta \hat{V}_\theta(x) = \nabla_\theta \left[K(z, \bar{z}) \right] \sum_{\tau=1}^{\bar{\tau}+1} \gamma^{\tau-1} \beta_\tau, \tag{8.43}$$

since the derived coefficients are fixed and computed through available data, therefore they do not depend on the chosen control parameter.
By relying on the general scheme of *policy gradient methods* (Sutton et al., 1999), the update for improving the policy parameter is given by:

$$\theta_{\tau+1} = \theta_\tau + \eta \nabla_\theta \hat{V}_\theta(x), \tag{8.44}$$

where η is the learning rate.
The latter procedure is in general guaranteed to converge to a local maximum for the value-function, with respect to the geometry induced by the policy parameterisation. As already discussed, under further technical assumption the latter gradient algorithm converges to the global optimum.

8.5 Illustrative example for control

In this section the applicability of the proposed approach as an actual algorithm is illustrated through some simple numerical simulations. In particular both policy evaluation and policy improvement task will be addressed, in order to tackle the general Reinforcement Learning problem. Therefore the following examples focus on showing the convergence of the whole procedure to the optimal parameter θ^* yielding the optimal

policy within the prescribed policy class.

A controlled discrete-time stochastic dynamical system is considered, whose evolution is given by:

$$x_{k+1} = f(x_k, \theta_k) + w_k, \tag{8.45}$$

in which the noise represents the model disturbance, and it is assumed to be Gaussian distributed, i.e. $w_k \sim \mathcal{N}(0, \overline{\sigma}^2)$.

The theoretical derivation of the approach covered the general case of an d-dimensional state and a l-dimensional parameter. However, for the purpose of facilitating the illustration of the results, scalar parameter will be considered in this section.

In the classical setting of LQR problems, the cost would be a strictly positive and quadratic function of the state. In order to match the latter setting through the definition of a reward function, this is defined as the inverse of the cost. In this way the reward function is still positive, and the best performing policy would be the one maximising the reward. The latter is also made explicitly dependent on the policy parameter, as

$$r(x, \theta) = \left(\left\| x - x^{(r)} \right\|^2 + \nu \|\theta\|^2 \right)^{-1}. \tag{8.46}$$

A regularisation term explicitly dependent on the policy parameter, given by $\nu \|\theta\|^2$, has been added, which can be interpreted as a control cost.

The environment is initialised in state x_0, with the control parameter θ_0, which can be selected as arbitrary values. At every time-step the agent experience a transition from the current state x_k to a new state x_{k+1}, according to the dynamics in (8.45) with the current parameter θ_k, and receives the corresponding reward.

At the current time k, and assuming that the starting time is $k_0 = 0$, k state-parameter pairs are available to the agent from past transitions, as well as the next state x_{k+1} and the k rewards $r_1, \ldots, r_{k+1}$.

The latter knowledge is exploited in order to compute the kernel matrices $K(\overline{z}, \overline{z})$ and $K(\overline{z}', \overline{z})$ which are necessary for the derivation of the approximation of the Koopman operator in (8.39). In particular, the experienced data are gathered as:

$$\overline{z} = \begin{bmatrix} \overline{x}_0 & \overline{x}_1 & \ldots & \overline{x}_{k-1} & \overline{x}_k \\ \overline{\theta}_0 & \overline{\theta}_1 & \ldots & \overline{\theta}_{k-1} & \overline{\theta}_k \end{bmatrix} \tag{8.47}$$

for the input locations, and

$$\overline{z}' = \begin{bmatrix} \overline{x}_1 & \overline{x}_2 & \ldots & \overline{x}_k & \overline{x}_{k+1} \\ \overline{\theta}_1 & \overline{\theta}_2 & \ldots & \overline{\theta}_k & \overline{\theta}_k \end{bmatrix} \tag{8.48}$$

for the output locations.

Note that in the last entry for $\overline{z}'$, the value of the parameter was kept fixed to θ_k. This is clearly due to the fact that the value θ_{k+1} is unknown, since it relies on the new value-function for which the kernel matrices are necessary. As discussed earlier, if the reward depends only on the next state, as in Assumption 8.2.1, the latter value of the parameter is irrelevant, since the prediction of the next parameter does not play any role, and only the mapping from Z to X is sought. If instead the reward is also dependent on the parameter θ, as in (8.46), the last entry actually plays a role, and there are different viable options for setting it. The one considered in (8.48) relies on the assumption that the proximity parameter η is small, so that θ_{k+1} is actually very close to θ_k. This is indeed a very common assumption in Reinforcement Learning (Bhatia, Altosaar, and Gu, 2017; Schulman et al., 2015; Schulman et al., 2017). Moreover, it makes sense to predict the dynamics under the fixed parameter θ_k as it is the one characterising the current behaviour of the state-transition function.

The computation of the kernel matrices and the subsequent derivation of the kernel form of the Koopman operator allows to recover the regression coefficients β_τ, for $\tau = 1, \dots, \overline{\tau}$. Then, as prescribed by the theory, the calculation of the gradient of the kernel function with respect to θ allows to determine the steepest descent direction of the value-function. Given that the current transition have already been performed, the objective of the agent is now to select the best parameter with respect to the new state x_{k+1}, which is known. By performing the gradient step, the new parameter θ_{k+1} becomes available, and it will replace the last entry in the matrix $\overline{z}'$ in computations for future steps. Indeed, the new transition will be stored and the current time moved, so that the new parameter for the current transition will appear as θ_k also in $\overline{z}$ for the next transition.

The issue of *exploration* is a key element in Reinforcement Learning (Stadie, Levine, and Abbeel, 2015; Tang et al., 2017; Houthooft et al., 2016; Burda et al., 2018; Osband et al., 2016). It can be regarded as a "curiosity bias" for the agent, which is allowed not to blindly trust the current estimate of the value function, but try different parameters instead, which may result in a better understanding of the dynamics.

This is true also for the proposed procedure, since the reconstruction through a non-parametric method could cause the emergence of local minima, which may significantly affect the gradient direction, slowing down the procedure. Moreover, if the kernel function $K(\cdot, \cdot)$ is an even function with respect to θ, then the initial gradient would be zero for symmetry reasons, and thus would remain so indefinitely. This is true e.g. for the RBF kernel, which has no has no preferential direction.

The latter can be visualised by thinking of a 2-dimensional Gaussian kernel in the plane

with axis x and θ. If there is no modification of the parameter on the first step, i.e. $\theta_0 = \theta_1$, it is easy to understand that the next point will always be taken in $\overline{\theta} = \theta_0 = \theta_1$, because a linear combination of the radial kernel sections does not yield any gradient in the θ direction.

To overcome this problem the gradient ascent in (8.44) has been modified with an additive white noise perturbation given by $\zeta \sim \mathcal{N}(0, \sigma_\theta^2)$.

In the following examples, the Gaussian kernel has been chosen as the basis for the kernel approximation of the Koopman operator, which jointly considers state and parameter, as:

$$K(z, \tilde{z}) = \exp\left(-\varrho \left\| z - \tilde{z} \right\|^2\right). \tag{8.49}$$

Both linear and nonlinear dynamics have been considered for the numerical evaluation of the proposed approach, nonetheless, for both experiments the following parameters are kept fixed:

$$\eta = 0.5; \quad \nu = 5 \times 10^{-4}; \quad \overline{\sigma} = 0.1; \quad \sigma_\theta = 0.1; \quad x_0 = 0; \quad \theta_0 = 0. \tag{8.50}$$

The kernel hyper-parameter ϱ and the regularisation parameter σ in the kernel formulation of the Koopman operator are optimised by maximising the log-marginal likelihood, i.e.

$$(\sigma, \varrho) = \arg\max \left\{ \log\left(\mathbb{P}\left(\hat{r}_1(\overline{z}) \mid \overline{z}, \overline{z}' \right) \right) \right\} \tag{8.51}$$

$$= \arg\min \left\{ \frac{1}{2} \hat{r}_1(\overline{z})^\top \kappa_{\sigma,\varrho} \hat{r}_1(\overline{z}) + \frac{1}{2} \log \left| \kappa_{\sigma,\varrho} \right| \right\} \tag{8.52}$$

where $\kappa_{\sigma,\varrho} = \left[K(\overline{z}, \overline{z}) + \sigma^2 \right]^{-1}$, and the minimisation refers to the prediction of the first step. For subsequent predictions the hyper-parameters are kept fixed. In particular, optimisation of hyper-parameters takes place at specific time-instants, i.e. at $k = 10, 50, 100, 200, 400$, because performing the computation of the optimal parameters at every time-step would become computationally cumbersome, while not yielding performance advantages, since it may take a significant amount of points to change the minimum of the optimisation.

Conveniently, the hyper-parameter tuning does not make the algorithm any slower, thanks to automatic differentiation. The *cheap gradient principle* (Kakade and Lee, 2018; Griewank and Walther, 2008) states that evaluating the gradient provably requires at most a small constant factor more arithmetic operations than the function itself. By searching for the maximum of the log-likelihood, it is necessary to evaluate the gradient of the reconstructed value-function with respect to the parameters, which nonetheless yield the same computational cost. This leads to an overall complexity of $O(\mathfrak{c} M^3)$, in

which the constant $\mathfrak{c}$ depends on how many times the hyper-parameters are updated, and on the required precision of the convergence of the gradient descent procedure at each optimisation subroutine. By keeping these numbers constant with respect to the number of training samples, the computational order of the procedure will remain the same. The latter means that the optimisation should not be performed at every step, because that would increase the computational complexity of a factor M. Note however that the optimal hyper-parameters will not change significantly when adding a single snapshot to a batch of many. Therefore for most tasks it is still meaningful to perform the optimisation in a fixed number of steps, which can be also adjusted while the algorithm is running.

8.5.1 Linear example

The Linear Quadratic Regulator (LQR), as underlined in Chapter 3, is an important benchmark for Reinforcement Learning, since it is a well-studied class of problems for which the exact solution is available, as explained in Chapter 7. Therefore the first simple numerical simulation is built upon this classical framework.
The dynamics are then linear in both the state and the control, as:

$$f(x, \theta) = Ax + B\pi_\theta(x) \tag{8.53}$$

in which the input is made explicitly dependent on the control parameter θ, and in particular is parameterised as an affine function of the state, i.e.,

$$\pi_\theta(x) = \theta\left(x - x^{(r)}\right), \tag{8.54}$$

which defines the class of admissible policies.
The objective of the algorithm is then to find the best parameter θ^*, for which the value function attain the highest value, when input functions as in (8.54) are considered.
It is worth noticing that in the case of a linear dynamics and reward in (8.46), the value-function is a quadratic function over the whole Z space, as for the Linear Quadratic Regulator. However, its maximum in Θ depends on the time horizon $\overline{\tau}$.
The following parameters are considered for the presented simulations:

$$A = 0.9; \quad B = 0.1; \quad x^{(r)} = 5. \tag{8.55}$$

Figure 8.1 and 8.2 show the behaviour of the control parameter when the proposed

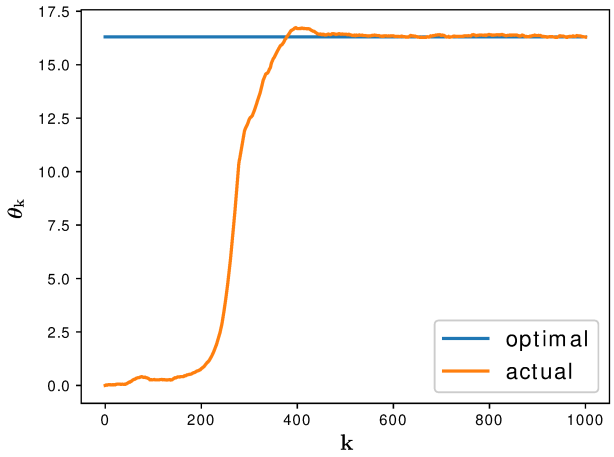


Figure 8.1: Behaviour of θ for the linear control problem over 1000 iterations of the proposed approach, with hyper-parameter optimisation taking place in the prescribed steps. The optimal parameter is depicted in blue, while the actual controller computed from the estimation over the different steps is given in orange. In this plot the horizon is set to $\overline{\tau} = 1$.

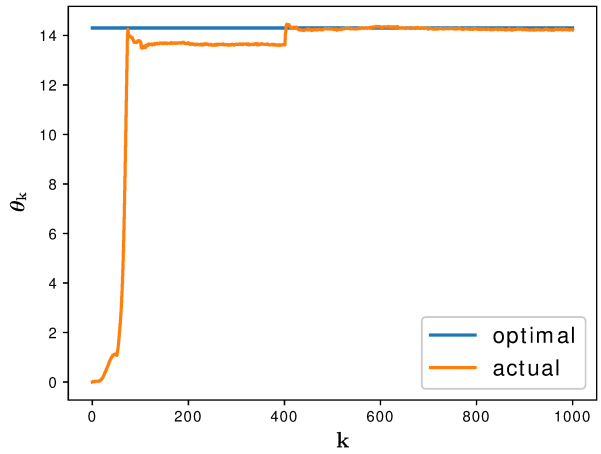


Figure 8.2: Control parameter as a function of time, when the kernel formulation of the Koopman operator is employed to solve the linear control problem. The estimation of the optimal parameter over 1000 iteration is drawn in orange, while the optimal value is shown in blue. The horizon here is set to $\overline{\tau} = 5$.

approach is applied to tackle the linear control problem, for different time horizon, respectively $\overline{\tau} = 1$ and $\overline{\tau} = 5$. Note that however the method presented in Section 8.4 make only use of samples which are 1-step ahead. Both the plots reveal that the algorithm is actually able to converge to the parameter of maximum reward and then stay there despite the exploration noise. The optimal controller is computed with the same technique, i.e. gradient ascent in the prescribed policy class, to which noise-free transition dynamics

was provided. The same procedure will be considered also in the nonlinear example, so that the setting is coherent.
In the second chart the algorithm moves faster because in the propagation over 5 steps the parameter θ has a greater impact, being the proximity parameter η kept fixed.
In order to better inspect the estimation of the value-function for the proposed approach, and then understand its capabilities in terms of policy evaluation, its estimates are plotted in Figure 8.3 along with their true values, as functions of θ. The plots shows the comparison for $k = 900$, therefore when the procedure has already reached the optimal value.

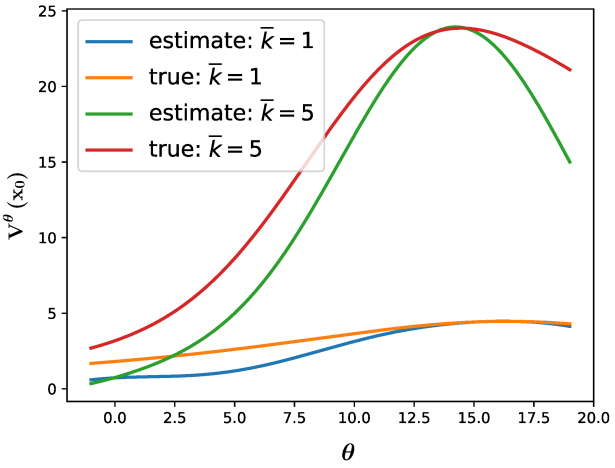


Figure 8.3: Estimation of the value-function in the linear case, for different horizons, $\overline{\tau} = 1$ and $\overline{\tau} = 5$, respectively in orange and in green. The figure shows the comparison with the true values, in blue and red, for $k = 900$, when the procedure has already reached the optimal value for the control parameter.

The accuracy of prediction is fairly kept when propagating the procedure for more steps. The latter will be further investigated in Chapter 9.

8.5.2 Nonlinear example

A slight modification of the linear dynamics in (8.53) is considered next, in order to address the case of nonlinear dynamics. In particular, in this case, the state-transition function is given by:

$$f(x, \theta) = Ax \sin(x) + B\pi_\theta(x), \tag{8.56}$$

with A and B as before, and the same proportional controller for the policy $\pi_\theta(x)$. In this example the reference state is taken to be $x^{(r)} = \pi/2$, while all other parameters are maintained unchanged.

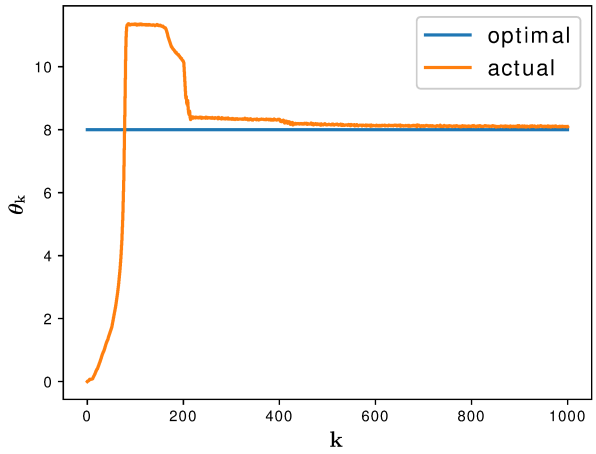


Figure 8.4: The behaviour of the control parameter as prescribed by the proposed approach, in the nonlinear example, over 1000 iterations. The θ parameter is shown in orange, while the optimal value is depicted in blue. The horizon is fixed to $\overline{\tau} = 1$.

In Figure 8.4 the results for the nonlinear dynamics are shown, in which it is considered the simplest case of horizon $\overline{\tau} = 1$.
Once again the proposed approach is capable of reaching the optimal value and identifying it as the optimal value, as it remains there afterwards.

9

Uncertainty propagation

This is the last chapter of this Dissertation and therefore it is rightly devoted to fine-tuning details and properties of the proposed approach, which has been introduced in the previous chapter. In particular, a gap between the presented theory and a standard assumption in simulation will be filled, through the characterisation of a link between the noise affecting the state dynamics, and the one affecting the observable dynamics. Moreover, an interesting property of the proposed algorithm is discussed in what follows, i.e., the possibility of propagating also the covariance description of the estimated observable, in such a way that confidence intervals around the estimate can easily be provided. The latter feature may be used to quantifying the uncertainty of the reconstruction and to mitigate the effect of model bias, supporting the idea that the proposed approach presents many differences with standard model-based algorithms.
This Chapter is drawn from the work by **Zanini, Francesco** and Chiuso, 2022, and therefore it presents original contributions.

9.1 Model-based or model-free?

The field of Reinforcement Learning is often divided into two different paradigms: *model-based* and *model-free* RL (Sutton and Barto, 2018).
The former is the closest to the Optimal Control problem (Recht, 2018), since it often assumes the knowledge of the reward function, which is regarded as a design variable. The outline of a model-based algorithm usually envisages the estimation of the environment, so that predictions of the transitions are available. This knowledge is then used to model the future behaviour of the system, making the problem of maximising the value-function or minimising the overall cost over a class of admissible policies well-defined. For instance, in order to tackle a linear time-invariant control problem as defined in (7.21) (see Chapter 7, Section 7.3), a model-based algorithm would first estimate the matrices A

and B characterising the system, and then compute the associated optimal controller with equation (7.24) based on the latter estimates (Dean et al., 2020; Li and Todorov, 2004; Tu and Recht, 2019a).
A Dynamic Programming solver is the prototypical example of a model-based algorithm. Actually most algorithms fall into this category, and among them also the procedure presented in Chapter 8, as well as *MuZero* algorithm by Schrittwieser et al., 2020, which indeed shares the same principle.
The model-free perspective instead is guided by a direct mapping from actions to rewards, so that the performances of a certain policy are considered to be the direct consequences of the actions, without modelling the propagation provided by the environment. For instance, in the same problem of a linear time-invariant controlled system, a model-free algorithm would estimate from data the gradient of the value-function, and then at every step, improve the parameter with gradient ascent (Tu and Recht, 2019a).
Examples of model-free algorithm are instead given by SARSA (Rummery and Niranjan, 1994; Sutton, 1995; Singh et al., 2000; John, 1994; Seijen et al., 2009; Hasselt, 2011), Q-learning (Watkins, 1989; Watkins and Dayan, 1992; Jaakkola, Jordan, and Singh, 1993; Tsitsiklis, 1994), and actor-critic methods (Witten, 1977; Barto, Sutton, and Anderson, 1983; Sutton, 1984; Degris, White, and Sutton, 2012; Williams, 1988; Williams, 1992).
The main distinction between the two paradigms refers strictly to whether, during learning or action selection, the agent uses predictions about the response of the environment. Regardless it being the pair of next state and reward, the expected next reward, or the full joint distribution of state and reward, either way a prediction is necessary, therefore the algorithm will be model-based.
These two sub-classes of Reinforcement Learning, model-free and model-based approaches, both yield strengths and weaknesses.
The model-free perspective may seem at first more tailored to the Reinforcement Learning problem, as it addresses the problem of maximising the return in a direct way, without having to learn a model of the environment. However algorithms following this perspective usually suffer from sample inefficiency. This means that learning is slow, so that it takes many samples from the environment to converge to a good solution. The asymptotic behaviour is instead usually considered to be better with respect to model-based algorithms, as they do not suffer from *model bias* (Tu and Recht, 2019b).
The latter issue stems from relying on the model that has been learnt, which - especially in the case of long term predictions - may result in a very inaccurate estimate. This is of course a feature of model-based methods only. Learning a model of the environment however allows for a compact representation of the dynamics and it is the reason behind

their sample efficiency. In fact, model-based methods have proven to be better under a low data regime and specific dynamic frameworks, as shown by Dean et al., 2020; Zheng et al., 2021.

The environment is modelled through its transition function, therefore the identification steps which is required to estimate its action on the agent is framed as the reconstruction of a function, with noisy data available. The latter is related with *supervised learning*, in which the *overfitting* issue is a pervasive and well-known problem (Everitt, 2002; Burnham and Anderson, 2002; Shalev-Shwartz and Ben-David, 2014). In the latter field, it has undoubtedly proved a good practice to restrain the representational capacity of the hypothesis class through a regulariser. The same difficulty translates to the Reinforcement Learning setting, in which the learned model should probably not be treated as correct, but its prediction capabilities should be mitigated, in order to improve performances on the environment (Arumugam et al., 2018). Indeed the model is represented through a function approximator, so the setting is actually the same. However predictions coming from the model are usually claimed with full confidence by the different model-based algorithms, leading to the aforementioned model bias issue.

By considering a Bayesian perspective, and tackling the regression problem through Gaussian Processes, this problem can be circumvented by considering the whole posterior distribution over functions, and not just the estimate. This would give an idea of the level of uncertainty in the model, and would allow to retrieve confidence intervals for the estimate. The same result is actually achieved by Deisenroth and Rasmussen, 2011, in which the Reinforcement Learning problem is framed through a Bayesian point of view. In particular, the same assumption on parameterised policies is considered, however kernel methods are there used to learn a true model of the environment, thus predicting state transitions, and not relying on the reward samples. In contrast to the latter work, the proposed procedure instead relies only on the reward, and thanks to this it can be extended to handle also the case in which its analytic expression is not provided to the learner. This extension will be presented in this chapter, which indeed is able to deal with the actual Reinforcement Learning setting. As already hinted in Chapter 1, the proposed approach can be regarded as a halfway procedure between the model-based and the model-free perspective. Undoubtedly, the Koopman operator does learn a model of the environment, however the targets of the regression are the value of the reward for the transition, and not the state. Moreover, the ability of propagating uncertainty for the prediction of future rewards, may help in mitigating the model bias issue, revealing the peculiar nature of the proposed "model-based" algorithm.

Note how the procedure highlighted in Chapter 8 is already framed in the correct way so

that the full distribution can be propagated. This is the main contribution of this last part of the Dissertation, in which a refined assumption on the measurement noise is also considered.

9.2 Posterior propagation through Koopman operators

In the illustrative examples for the proposed procedure, both for estimation (in Chapter 6, Section 6.7) and control (see Chapter 8, Section 8.5), the classic assumption of a Gaussian noise in the state dynamics was considered. However, for the theoretical derivation of the kernel formulation of the Koopman operator, and most importantly for Proposition 6.5.1 (see Chapter 4, Section 6.5), the noise model which has been taken into consideration was:

$$\psi\left(\overline{y}\right) = \psi\left(f\left(\overline{x}\right)\right) + \varepsilon, \tag{9.1}$$

where ε was assumed to be Gaussian, which is an additive noise in the observable dynamics, and not in the state-transition function.
The latter gap is filled in this chapter, since the case with the Gaussian additive noise in the state dynamics is addressed from the next section, and it is linked with the noise in the observable dynamics. Also, the problem of propagating the covariance of the estimated observable is considered, which makes it possible to obtain confidence bounds on the estimation of the value-function.
Hence, in what follows, the usual assumption of a Gaussian noise in the state dynamics is considered. From the usual disturbance model in the state, a characterisation for noise in the evolution of the observable is retrieved, so that the estimation technique provided in Chapter 6 can still be applied.
Consider the standard assumption on the noise affecting the flow of a generic discrete-time dynamical system, whose evolution is characterized by:

$$x_{k+1} = f\left(x_k\right) + w_k, \tag{9.2}$$

where the transition map goes from the state space X in itself and the additive noise process w_k, which is assumed to be i.i.d. zero-mean Gaussian $w_k \sim \mathcal{N}\left(0, \overline{\sigma}^2 I_d\right)$, models the stochasticity in the dynamics.
The main aim of the present chapter is to precisely characterise the estimator of $\psi_\tau\left(x_0\right) := \mathbb{E}\left[\psi\left(x_\tau\right)\right]$ so that it is possible to derive uncertainty bounds. This result can be obtained in a Bayesian estimation framework, so that the estimate will be characterized by the mean and variance of the posterior distribution, for a generic scalar-valued function

$\psi(\cdot)$. The connection with the kernel formulation of the Koopman operator introduced in Chapter 4 is therefore evident.
The estimator derived in this chapter will again rely on data coming from the true system, corresponding to snapshots of the evolution in (9.2), i.e. by pairs $\{\overline{x}_i, \overline{x}'_i\}_{i=1}^{M}$, so that

$$\overline{x}'_i = f(\overline{x}_i) + w,$$

in which the training points do not necessarily have to be taken consecutively along a single trajectory; each $\overline{x}_i$ can represent an arbitrary location, provided that the corresponding output location is also available.
In order to present the contents of this chapter in simpler way, the strategy for propagating the uncertainty in the estimation of the observable is divided into two subsection.
Firstly it is assumed that the observable $\psi(\cdot)$ to be propagated is known, i.e. its analytic expression is available. Note that this is the case for the first step of the propagation in the setting considered so far. Indeed in the previous chapters the observable is taken to be the reward of the Reinforcement Learning problem, which has been considered to be given. It should be noted that in the last subsection (Subsection 9.2.3) a method to deal with unknown rewards will be presented, which is able to achieve basically the same performances although exploiting only the observations of the reward $\psi(\overline{x}')$, without the knowledge of the analytic expression of the function. The latter is indeed the true setting of the Reinforcement Learning problem.
Secondly, the case of a stochastic observable is addressed, in which the combined uncertainty of the dynamics and the observable should be propagated. In particular the proposed approach models the iterative propagation of the observable as a Gaussian Process, so that the covariance it is easy to handle. This is the case for all steps after the first of the propagation in the setting considered in the previous chapters.

9.2.1 Known observable

Recall that the Koopman operator describes a generic system in terms of the evolution of a function $\psi(\cdot)$, so that the observable evolution can be considered, consisting of $\psi(\cdot)$ applied to (9.2), i.e.:

$$\psi(x_{k+1}) = \psi(f(x_k) + w) = \psi(f(x_k)) + \tilde{\varepsilon} \tag{9.3}$$

$$\simeq \psi(f(x_k)) + \varepsilon, \tag{9.4}$$

in which the last equality express the noise acting in the state dynamics w as an additive noise $\tilde{\varepsilon}$ in the observable dynamics which may have an arbitrary distribution. The

approximate inequality instead defines ϵ, which is a Gaussian approximation of the noise affecting the observable dynamics. The latter is a zero-mean Gaussian noise, with the variance depending on the derivative of the observable. The precise characterisation is given in the following proposition.

Proposition 9.2.1. *Considering the approximation in (9.4), the estimate of the Koopman Operator in a Reproducing Kernel Hilbert Space is given by:*

$$\widehat{\psi\left(f\left(\cdot\right)\right)} = K\left(\cdot,\overline{x}\right)\left[K\left(\overline{x},\overline{x}\right) + \mathbb{V}\left[\varepsilon\right]\right]^{-1}\psi\left(\overline{x}'\right), \tag{9.5}$$

where ε is zero-mean Gaussian distributed with variance

$$\mathbb{V}\left[\varepsilon\right] = \sigma^2 \left.\frac{\partial}{\partial a}\psi\left(a\right)\right|_{a=\overline{x}'} \left.\frac{\partial}{\partial a}\psi\left(a\right)\right|_{a=\overline{x}'}^{\top}. \tag{9.6}$$

Proof. The distribution of the noise ε is derived from a I order Taylor expansion, which corresponds to the linearisation of equation (9.3), defining the approximation in (9.4). The latter is computed as follows:

$$\begin{aligned}
\psi_1\left(\overline{x}\right) &= \psi\left(f\left(\overline{x}\right) + w\right) && (9.7)\\
&= \psi\left(f\left(\overline{x}\right)\right) + \left.\frac{\partial}{\partial a}\psi\left(a\right)\right|_{a=f(\overline{x})} w + O\left(w^2\right) && (9.8)\\
&\simeq \psi\left(\overline{x}'\right) + \left.\frac{\partial}{\partial a}\psi\left(a\right)\right|_{a=\overline{x}'} w, && (9.9)
\end{aligned}$$

in which the approximation comes from the fact that higher-order terms in the noise are neglected, and that the last expression is computed in $\overline{x}'$ instead of $f\left(\overline{x}\right)$, which are assumed to be close enough.
This allows to characterise the distribution of the noise in the observable dynamics as

$$\tilde{\varepsilon} \sim \mathcal{N}\left(0, \sigma^2 \left.\frac{\partial}{\partial a}\psi_n\left(a\right)\right|_{a=\overline{x}'} \left.\frac{\partial}{\partial a}\psi_n\left(a\right)\right|_{a=\overline{x}'}^{\top}\right). \tag{9.10}$$

As detailed in Chapter 6, and following the derivation by Rasmussen and Williams, 2006, the estimate of the Koopman operator in a Reproducing Kernel Hilbert Space is equivalent to the maximum a posteriori estimator for $\psi\left(f\left(\cdot\right)\right)$ under a Gaussian prior on $\psi\left(f\left(\cdot\right)\right) \sim \mathcal{N}\left(0, K\left(\cdot,\cdot\right)\right)$.
Since the observable observations $\psi\left(\overline{x}'\right)$ are considered as training outputs corresponding to the input locations provided by $\overline{x}$, the conditional expectation given the latter data

can be computed as:

$$\widehat{\psi\left(f\left(\cdot\right)\right)} = K\left(\cdot, \overline{x}\right)\left[K\left(\overline{x}, \overline{x}\right) + \mathbb{V}\left[\varepsilon\right]\right]^{-1}\left(\psi\left(\overline{x}'\right) - \mathbb{E}\left[\varepsilon\right]\right), \tag{9.11}$$

according to the approximate model which considers an additive Gaussian disturbance in the observable dynamics, given by ε. □

Note that the estimator obtained in (9.11) is precisely the sought function $\hat{\psi}_1\left(\cdot\right)$, whose posterior distribution is completely characterised by the Gaussian approximation given by ε. This step was simple, as the observable itself does not inject noise in the transition, being deterministic.
It turns out then that the resulting observable is modelled as a Gaussian Process, equipped with a covariance description, so that the derivation of confidence bound is straightforward. However, in order to iterate the process and to get an estimate of $\psi_2\left(\cdot\right)$, the required computations are now different, as the observable that needs to be propagated is intrinsically stochastic. The latter point would not be an issue if only the expectation of the estimator were to be considered, as done so far and explained in Chapter 8, Section 8.2. Nonetheless the primary aim of this chapter is to derive a way of propagating the posterior distribution, therefore also the covariance needs to be taken into account. The latter is addressed in the next subsection, where the generic n-th step would be considered, with $n > 1$. Since each subsequent estimator $\psi_n\left(\cdot\right)$ will be modelled as a Gaussian Process, so the propagation is the same for all step.

9.2.2 Estimated observable

The Bayesian reconstruction of $\psi_1\left(\cdot\right)$ resulted in a distribution over functions, whose uncertainty must be properly considered in the propagation over the evolution of the dynamics. As will be proven later, in Proposition 9.2.2, the iterated observables will all be described by Gaussian Processes: this makes it possible to handle all the steps above the first one equally, enclosing them together in a single discussion for a generic n-th step, with $n > 1$.
Note indeed that $\psi_n\left(\cdot\right)$ is still an observable, though a stochastic one, which describes the function of interest $\psi\left(\cdot\right)$ after n iterations of the state transition function.
Let $\hat{\psi}_n\left(\cdot\right)$ denote its conditional expectation given data $\{\overline{x}, \overline{x}'\}$, and denote by Σ_n its corresponding conditional variance, so that:

$$\begin{cases} \hat{\psi}_n\left(\cdot\right) = \mathbb{E}\left[\psi_n\left(\cdot\right) \mid \overline{x}, \overline{x}'\right] \\ \Sigma_n = \mathbb{V}\left[\psi_n\left(\cdot\right) - \hat{\psi}_n\left(\cdot\right)\right], \end{cases} \tag{9.12}$$

which characterises the full distribution under the Gaussian assumption.
Although the observable for the n-th step is stochastic, the same linearisation as in (9.4) is considered, in which however the Gaussian approximation of the additive noise acting on observables will take a different form. The observable dynamics for the n-th step is given by:

$$\psi_n\left(x_{k+1}\right) = \psi\left(f\left(x_k\right) + w\right) = \psi\left(f\left(x_k\right)\right) + \tilde{\varepsilon}_n \tag{9.13}$$

$$\simeq \psi\left(f\left(x_k\right)\right) + \varepsilon_n \tag{9.14}$$

which in fact holds for $n \geq 1$.
Note indeed that the generic noise which achieves the equality in (9.13) is different at each step n, because it depends on the particular observable $\psi_n\left(\cdot\right)$, which in general is different for each propagation index. The latter means that:

$$\tilde{\varepsilon}_n \neq \tilde{\varepsilon}_{\tilde{n}} \tag{9.15}$$

$\forall n \neq \tilde{n}$.
In turn, also the Gaussian approximation in (9.14) in general will change with the time-step n, i.e.

$$\varepsilon_n \neq \varepsilon_{\tilde{n}} \tag{9.16}$$

$\forall n \neq \tilde{n}$.
Since the true observable $\psi_n\left(\cdot\right)$ is not available for $n \geq 1$, but only its estimator given by the conditional expectation $\hat{\psi}_n\left(\cdot\right)$, the induced observable dynamics are modelled as follows:

$$\hat{\psi}_n\left(x_{k+1}\right) = \psi_n\left(f\left(x_k\right)\right) + \tilde{\psi}_n\left(f\left(x_k\right)\right) \tag{9.17}$$

$$\simeq \psi_n\left(f\left(x_k\right)\right) + \underbrace{\tilde{\psi}_n\left(x_{k+1}\right) + \varepsilon_n}_{E_n\left(x_{k+1}\right)} \tag{9.18}$$

where (9.17) defines $\tilde{\psi}_n\left(f\left(\cdot\right)\right)$, which is the difference between the true function and the conditional expectation given data; and the last approximate equality comes again from the Gaussian approximation of the actual noise, though this time it includes also the uncertainty provided by the stochasticity of the observable.
Note that the model for the observable dynamics thus obtained matches the original framework given in (9.4). Therefore the same approach as in the previous subsection can be applied, provided that the noise $E_n\left(\cdot\right)$ is adequately characterised. The latter is composed of the sum of the uncertainty coming from the observable, and the noise

characterising the dynamics. These are both modelled as Gaussian: the former because is the error of a Gaussian process estimator, hence it distribution is Gaussian; the latter because of the usual approximation that has been taken from the beginning. Moreover they are here assumed to be independent, so that their sum is still Gaussian distributed (Lemons, Langevin, and Gythiel, 2002), and in particular it holds that:

$$\mathbb{V}\left[E_n\left(x_{k+1}\right)\right] = \mathbb{V}\left[\tilde{\psi}_n\left(x_{k+1}\right)\right] + \mathbb{V}\left[\varepsilon_n\right] = \Sigma_n\left(x_{k+1}\right) + \varepsilon_n. \tag{9.19}$$

As explained before, the characterisation provided by equation (9.19) allows to iterate the same procedure outlined in Proposition 9.2.1 for steps greater than 1, resorting to the measurement model in (9.18).

Proposition 9.2.2. *Consider the measurement model in (9.18), which is satisfied by evaluations of the stochastic observable given by $\hat{\psi}_n\left(x_{k+1}\right)$, for all $n \geq 1$. Then the estimate of the next observable $\psi_{n+1}\left(\cdot\right) = \psi_n\left(f\left(\cdot\right)\right)$, based on data $\{\overline{x}, \overline{x}'\}$, obtained through the kernel formulation of the Koopman operator, is given by:*

$$\begin{cases} \hat{\psi}_{n+1}\left(\cdot\right) = K\left(\cdot, \overline{x}\right)\left[K\left(\overline{x}, \overline{x}\right) + \mathbb{V}\left[E_n\right]\right]^{-1}\hat{\psi}_n\left(\overline{x}'\right) \\ \mathbb{V}\left[\psi_{n+1}\left(\cdot\right)\right] = K\left(\cdot, \cdot\right) - K\left(\cdot, \overline{x}\right)\left[K\left(\overline{x}, \overline{x}\right) + \mathbb{V}\left[E_n\left(\overline{x}'\right)\right]\right]^{-1}K\left(\overline{x}, \cdot\right) \end{cases} \tag{9.20}$$

Proof. The proof follows the same lines as Proposition 9.2.1, however the measurement model in (9.18) must be considered.
By specialising the latter observable dynamics for the available data, it holds:

$$\hat{\psi}_{n+1}\left(\overline{x}\right) = \hat{\psi}_n\left(f\left(\overline{x}'\right)\right) + E_n\left(\overline{x}'\right), \tag{9.21}$$

in which all the uncertainty is captured by the additive noise $E_n\left(\overline{x}'\right)$. The proof is trivially concluded by writing down the MAP estimator for the next observable $\psi_{n+1}\left(\cdot\right)$, when the following Gaussian prior is considered:

$$\psi_{n+1}\left(\cdot\right) \sim \mathcal{N}\left(0, K\left(\cdot, \cdot\right)\right) \tag{9.22}$$

so that the Gaussian Process regression framework can be exploited. □

The last thing left to do is to characterise the overall noise approximated by the term $E_n\left(\cdot\right)$ in (9.18). In this regard it should be noted that the covariance of the posterior of the observable is given by the Gaussian Process regression framework, which is well-known, therefore the only term left out is the one coming from the stochasticity

of the state dynamics, i.e., ε_n. The latter is again approximated through a I order Taylor linearisation as:

$$\varepsilon_n = \frac{\partial}{\partial a}\hat{\psi}_n(a)\Big|_{a=\overline{x}'} w. \tag{9.23}$$

Its expectation is clearly 0, which also justifies the expression in (9.20). It variance, which is actually used in the propagation, is instead given by:

$$\mathbb{V}\left[\frac{\partial}{\partial a}\hat{\psi}_n(a)\Big|_{a=\overline{x}'} w\right] = \overline{\sigma}^2\mathbb{V}\left[\frac{\partial}{\partial a}\hat{\psi}_n(a)\Big|_{a=\overline{x}'}\right] + \\ + \overline{\sigma}^2\mathbb{E}\left[\frac{\partial}{\partial a}\hat{\psi}_n(a)\Big|_{a=\overline{x}'}\right]\mathbb{E}\left[\frac{\partial}{\partial a}\hat{\psi}_n(a)\Big|_{a=\overline{x}'}\right]^\top. \tag{9.24}$$

The latter Gaussian approximation of the noise affecting the state dynamics completes the description of the proposed approach for the propagation of any observable along with its II order characterisation as a Gaussian Process.
For a generic step, formulas in (9.20) gives the conditional expectation and the variance of the same observable when propagated through the state dynamics, taking into account all sources of uncertainty in the process. Clearly, by evaluating the very same expressions for the available data $\overline{x}'$, new synthetic observations becomes available, also with their covariance, so that the method can be iterated indefinitely in time.

9.2.3 Dealing with an unknown reward function

It is often the case, in Reinforcement Learning problems, that the agent directly collects evaluations of the reward while no explicit knowledge of the reward function is available. Nonetheless, in previous chapters and also in the method proposed in the earlier subsections, the explicit analytic expression of the reward function is required.
On closer inspection, however, it can be seen from the previous subsection that in order to propagate the observable for the next step, only the knowledge of the observable at the previous step is required, as given in (9.20), which is nonetheless estimated from observations. Moreover, in the first step, the explicit knowledge of the observable is only used to characterise the approximation of the noise ε in (9.10), through its derivative.
Therefore, if there was a way to approximate the latter derivative, then the procedure could run seamlessly only relying on samples of the reward function and not on its explicit formulation.
It turns out that this can actually be achieved by iteratively updating an estimate of the derivative, switching between two formulations of the derivative of the first propagation of the observable.

Recall indeed the derived formula for the first observable, which yields:

$$\hat{\psi}_1(\cdot) = K(\cdot, \overline{x}) \left[K(\overline{x}, \overline{x}) + \Sigma_0\right]^{-1} \psi(\overline{x}'), \tag{9.25}$$

in which only the derivative of $\psi(\cdot)$ is employed, and it appears only in Σ_0. The derivative of $\hat{\psi}_1(\cdot)$ can be computed from its kernel formulation, as:

$$\frac{\partial}{\partial a}\hat{\psi}_1(a)\Big|_{a=\overline{x}} = \frac{\partial}{\partial a}K(a, \overline{x})\Big|_{a=\overline{x}} \left[K(\overline{x}, \overline{x}) + \Sigma_0\right]^{-1} \psi(\overline{x}'), \tag{9.26}$$

and it is specialised for the available data $\overline{x}$. Note that this expression gives an indirect relation between the derivative of the original observable and the derivative of its first propagation.
The latter can be also obtained through the chain rule, which allows to obtain another relationship between the same quantities, as:

$$\frac{\partial}{\partial a}\psi_1(a)\Big|_{a=\overline{x}} = \frac{\partial}{\partial a}\psi(a)\Big|_{a=f(\overline{x})} \frac{\partial}{\partial b}f(b)\Big|_{b=\overline{x}} \tag{9.27}$$

$$\simeq \frac{\partial}{\partial a}\psi(a)\Big|_{a=\overline{x}'} \frac{\partial}{\partial b}f(b)\Big|_{b=\overline{x}}, \tag{9.28}$$

which is again specialised for the available data $\{\overline{x}, \overline{x}'\}$.
These two formulations can give rise to an iterative procedure that empirically appears to have as its only fixed point the actual derivative of the observable, in the following way. First, the sought derivative of the observable, computed in the output locations, should be initialise to an arbitrary value, e.g. $\dot{\psi}(\overline{x}') = 1$. With the latter initialisation, some (possibly wrong) values for the first propagation of the observable in the input locations can be computed through equation (9.25). These can be entered in the second formulation, equation (9.28), in order to obtain new value for the sought derivative of the observable. In fact under the assumption that the matrix $\frac{\partial}{\partial b}f(b)\Big|_{b=\overline{x}}$ is invertible the latter equation can be used to compute the new values as:

$$\frac{\partial}{\partial a}\psi(a)\Big|_{a=\overline{x}'} \simeq \frac{\partial}{\partial a}\psi_1(a)\Big|_{a=\overline{x}} \left[\frac{\partial}{\partial b}f(b)\Big|_{b=\overline{x}}\right]^{-1}, \tag{9.29}$$

so that a new matrix Σ_0 can be computed with the new estimate of the derivative of the observable, and the procedure iterated.
It might seem that the requirement of the knowledge of the analytic expression of the observable has simply been replaced with the constraint of having access to the state-transition function f - which would be a different, if not more restrictive, setting.

However, there is a simple way of estimating the state dynamics, by making use of the proposed procedure. It is sufficient to take as observable the identity function, so that the estimation of the composition will actually turn out to yield the state-transition function. Although the procedure has explicitly been derived to avoid this step, which would be equal to estimate a model of the environment, it has been experimentally proved that there is a low sensitivity for the reconstructed derivative with respect to the estimated state-transition function. This means that a rough estimate may be enough for the procedure to converge to good values, and the latter can be surely provided by the presented method, taking the identity function as observable, which can be computed as a sort of 0-th step.

9.3 Illustrative example for uncertainty propagation

In this section a simple example is set up, in order to empirically evaluate the proposed procedure.
The latter is given by some nonlinear dynamics, which are one-dimensional for the sake of illustration:

$$x_{k+1} = f(x_k) + w = -x_k^{3/2} + w; \tag{9.30}$$

where $w \sim \mathcal{N}(0, \overline{\sigma}^2)$ is taken as the standard additive Gaussian noise in the state dynamics, with $\overline{\sigma} = 0.1$.
The state-transition function is clearly stable in the interval $[-1, 1]$ and $x^{(e)} = 0$ is a trivial equilibrium point.
The reward function, which corresponds to the observable which should be propagated, has been considered to be:

$$r(x) = \exp\left(-\|x\|^2\right). \tag{9.31}$$

The function clearly penalises deviations from 0, therefore the system is naturally evolving towards its highest value.
In the following example only the estimation step for the value-function will be addressed, as it is clear from Chapter 8 that there is no difference in the actual algorithm for the procedure which takes care also of control. It is just required to extend the state-space with an augmented space of state and parameters, estimate the value function in that space, and take the gradient ascent with respect to that estimation.
The system considered in this simple example can then be thought as an already controlled dynamics, i.e. expressing the closed-loop under a particular policy, so that only the policy evaluation step will be performed. The policy selecting the actions is assumed to be stabilising, so that the state remains bounded.

The method outlined above has been implemented by means of a Radial Basis Function kernel, i.e.:

$$K\left(x, \tilde{x}\right)_{\xi,\varrho} = \xi \exp\left(-\varrho \left\|x - \tilde{x}\right\|^2\right), \tag{9.32}$$

for which the hyper-parameter has been optimised in the following fashion:

- The tuple $\text{hps} = (\xi, \varrho, \sigma^2)$ containing also the regularisation parameter σ^2 - which should be an estimate of $\overline{\sigma}^2$ - is optimised by minimising the log-marginal likelihood for the problem of estimating the state-transition function - the so-called 0-th step - as:

  $$\text{hps}^* = \arg\min_{\text{hps}} \left\{ \frac{1}{2} \overline{x}'^{\top} \kappa_{\text{hps}} \overline{x}' + \frac{1}{2} \log \left|\kappa_{\text{hps}}\right| \right\} \tag{9.33}$$

  with $\kappa_{\text{hps}} = \left[K\left(\overline{x}, \overline{x}\right)_{\xi,\varrho} + \hat{\sigma} I_M\right]^{-1}$.
  Recall that this step is not mandatory for the main procedure if the explicit knowledge of the reward is available; on the other hand, the latter is a necessary phase if the method relying only on samples from the observable dynamics must be exploited, as explained in Section 9.2. Note therefore that this does not burden the computational complexity of the algorithm.

- A new optimisation for each propagation step is performed for the main procedure, in which however the regularisation parameter is kept fixed at the value found at the previous point. The hyper-parameters of the kernel function are optimised again through log-marginal likelihood with a modified regulariser, which takes into account the actual noise affecting each step:

  $$\xi^*, \varrho^* = \arg\min_{\xi,\varrho} \left\{ \frac{1}{2} \overline{x}'^{\top} \kappa_{\xi,\varrho}^{(E_n)} \overline{x}' + \frac{1}{2} \log \left|\kappa_{\xi,\varrho}^{(E_n)}\right| \right\} \tag{9.34}$$

  for the n-th step, with $\kappa_{\xi,\varrho}^{(E_n)} = \left[K\left(\overline{x}, \overline{x}\right)_{\xi,\varrho} + \mathbb{V}\left[E_n\left(\overline{x}'\right)\right]\right]^{-1}$.

The regularisation parameter σ acting inside $\mathbb{V}\left[E_n\right]$ has been kept fixed because the joint optimisation with the other hyper-parameters did not lead to a significant increase of performances, and in this way the computational burden is lighter. Moreover, the numerical procedure adopted for jointly optimise the different parameter has been found to be less robust, leading to some numerical errors for some initialisations.
The implementation relies on the GPyTorch library (Gardner et al., 2018) in Python, which has been extended with an original feature to handle the heteroskedatic noise model when $E_n\left(\cdot\right)$ is involved. Indeed, since the noise depends on the observable, it takes a different variance at different location of the state space. Therefore it is necessary to

deal with a different level of noise in different region of the domain, giving rise precisely to a heteroskedatic model.
Several initial conditions are provided at each round of optimization to enhance the probability of convergence to the global maximum.
The hyper-parameter optimisation is performed for each step in the considered horizon for the value-function $\overline{\tau}$, nonetheless making use of the same data, so that no further interactions with the system and no simulations of the dynamics are required to carry out the optimization. This means that a different set of hyper-parameters is found at each considered step, in order to model a different scenario, although using the same data given data from the system.
When no explicit knowledge of the reward is provided to the learner, a fixed number of iterations $N = 10$ of the procedure presented in Subsection 9.2.3 is carried out, since convergence has been empirically proven to be very fast.

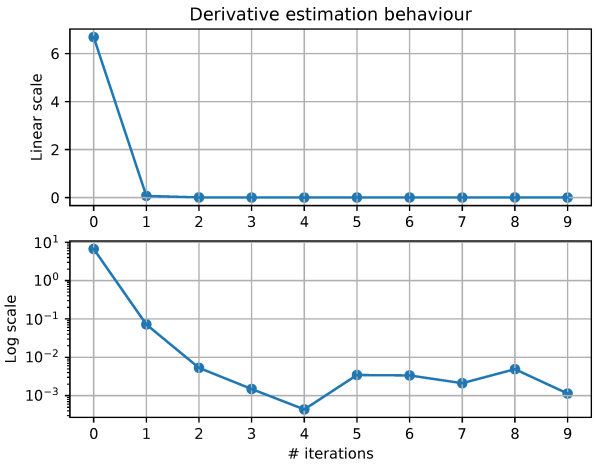


Figure 9.1: Behaviour of the derivative of the unknown observable which are used to estimate the first propagation of the reward. The two plots represents the same quantity with different axes, i.e., the difference estimation of the first propagation of the observable obtained with the two different formulations, (9.25) and (9.28).

In Figure 9.1 the behaviour of the estimate of the derivative is reported, for $N = 10$ updates of the derivative, which can be seen to quickly converge to a stationary value.
In order to check that the propagation has taken place correctly, in Figure 9.2 is reported the reconstruction of the observable which is evolved twice with respect to the dynamics, both through this technique and the one making use of the explicit knowledge of the reward.
In particular, in the latter figure the expectation and the 95% confidence intervals are plotted, which are built from the Gaussian distribution.

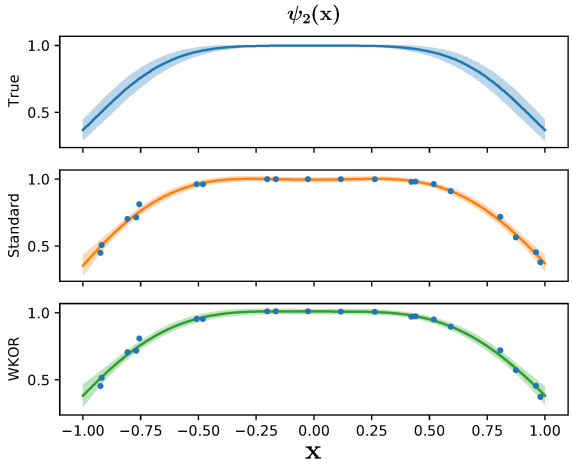


Figure 9.2: Reconstruction of $\psi_2(\cdot)$ both in the case of explicit knowledge of the reward and when only samples are provided. The blue dots represent the original samples after being propagated through the different iterations of the procedure. Estimates are equipped with their 95% confidence intervals given by 1.96 standard deviations above and under the mean.
Top: The ground truth given by Monte Carlo approximation of the noisy dynamics composed with the reward.
Middle: The reconstruction with the standard procedure, when the reward is provided to the learning algorithm.
Bottom: Result of the estimation procedure without explicit knowledge of the reward.

The true observable has been reconstructed by means of a Monte Carlo estimate of the true dynamics, which are stochastic, randomising over the reward of future states. For each test x on a fine grid, MC $= 1000$ samples of the iterated dynamics were collected, for each step considered in the computation of the value-function. The latter samples have been mapped through the reward function, which is deterministic, resulting in a sample distribution for the compositions $\psi_n(\cdot)$, with $\psi(\cdot) = r(\cdot)$ for each step. The reference curve for comparison with the proposed procedure is given by the sample mean of the latter distribution, and confidence intervals are computed relying on the sample estimate of the standard deviation.

Figure 9.3 shows that the proposed procedure successfully estimates the composition of the reward with the state-transition function at each time step, since the value-function is a cumulative description of the $\overline{\tau}$-th first steps. This is true even if the analytic expression of the reward is not provided to the learning agent, mimicking the actual Reinforcement Learning scenario. The two estimates are indeed very closed to each other, which also means that the procedure in 9.2.3 converges to good valued for the derivative of the reward.
In the last figure the estimation for the value-function $V_\pi(\cdot)$ is reported, for both of the

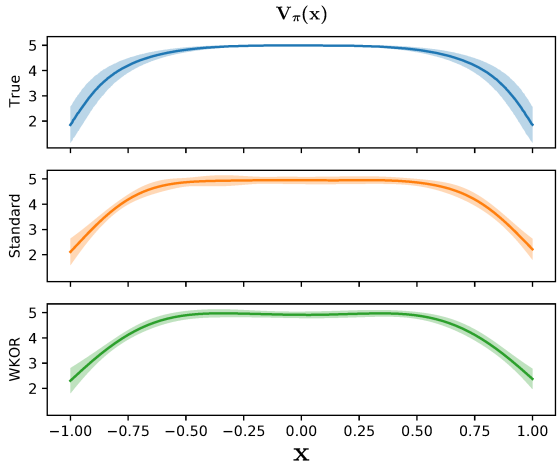


Figure 9.3: Reconstruction of the value-function $V_\pi(\cdot)$ again with both the presented procedures. The horizon is set to $\overline{\tau} = 6$. Estimates are equipped with their 95% confidence intervals given by 1.96 standard deviations above and under the mean.
Top: The ground truth given by Monte Carlo approximation of the noisy dynamics composed with the reward.
Middle: The reconstruction with the analytic expression of the reward.
Bottom: Estimate relying only on samples from the dynamics.

approaches, in which the ground truth is obtained as before. The horizon is set to $\overline{\tau} = 6$. The computation of the value-function just requires to sum the different estimates for the different step, as introduced in Chapter 8, Section 8.3. The similarity between the two approaches is then preserved.

10
Conclusion

In the present work, a new methodology is presented to address the Reinforcement Learning paradigm, which exploits the Koopman operator, and in particular its formulation in Reproducing Kernel Hilbert Space. The latter approach is built on discrete observations of the reward function of the system. Therefore, first of all, an analysis on the optimal step-size for the discretisation of a continuous-time dynamical system is carried out, with particular reference to the Riemann approximation for the value-function of a Langevin system with quadratic cost. Numerical simulations showed that the same trade-off applies to general nonlinear dynamics. Then, the theoretical connections between the Koopman operator framework and kernel methods are deeply explored, from a perspective which has not been considered in the current literature. The benefits of the formulation of the Koopman operator in Reproducing Kernel Hilbert Spaces are also made explicit by illustrative examples. This procedure is then applied to the Reinforcement Learning setting, which is the natural scenario given that the performance of the policy is described by the propagation of the reward alone. Explicit derivation of the value-function through the Koopman operator is addressed in detail, and the validity of the approach is assessed in numerical simulations. Confidence intervals are easily recovered provided the uncertainty about the reconstruction of the reward for future steps is correctly propagated.

The way the Reinforcement Learning problem is tackled in this Dissertation is through the estimation of the value-function. The latter is indeed a crucial quantity in Reinforcement Learning, through the knowledge of which it is possible to recover the optimal policy for the given setting. One of the key results of this work is a formal derivation of the value-function in terms of the Koopman operator. By learning how the reward function is propagated through time under the action of the environment and on the selected policy, the Koopman operator easily allows to directly characterise the value-function. In fact, even if the Koopman operator gives a comprehensive description of the dynamical

system under analysis, the latter framework allows the algorithm to be specialised for the evolution of a particular function: the reward. This deviates from the classical model-based approach, in which the state-transition function is estimated. The distinctive feature of the Koopman operator is that describes the dynamical system through the evolution of functions, allowing to focus on the evolution of a specific one. That is why this approach can be placed in between the model-based perspective and the model-free one. It certainly is a model-based approach, as it relies on predictions of the reward for future steps in order to recover the value-function. Nonetheless it directly addresses the estimation of the composition of the reward and the state-transition function, unlike most model-based algorithms. Clearly, if the dynamics of the system are extremely simple with respect to the composition of the reward and transition function, this approach may be at a disadvantage. It is therefore recommendable in cases where the transition function is hard to estimate and the reward is smooth, or can be designed as such. The proposed approach indeed does not need an estimate of the state-transition function, which can be arbitrarily complex. By directly addressing the evolution of the reward instead, this procedure is guaranteed to estimate what is really needed, avoiding reconstructing a simplistic model of a very complex system, thus not being affected by the model bias problem, which is certainly one of the greatest drawbacks for model-based methods. This difficulty is even more mitigated by the easily computable confidence intervals, which is a feature that only a few methods in model-based Reinforcement Learning can boast. Moreover, by selecting a reward function which is smooth enough, the composition will be a smooth function as well, so that the regularity imposed by kernel methods is actually suitable for the estimation problem.

Another major contribution of this Dissertation is indeed a formal derivation of the Koopman operator in RKHS, which removes the considerable problem of having to select a dictionary of function that should meaningfully approximate the evolution of the observable. By resorting to kernel methods, the latter step is substituted with the choice of the kernel, which is a well-known framework which has proved flexible enough to be used in many identification scenarios. If there is no prior information on the system, which can be for instance derived by physical principles, there is no way to select a suitable dictionary of function. There are instead many works on how to select a kernel function, based on the kind of prior knowledge it induces. The final estimate of the value-function is given by a linear combination of kernel sections centred on available datapoints, so that its generalisation capabilities and limitations are the same as for standard kernel methods. Moreover, oftentimes the reward function can be regarded as a

free variable, to be designed so that the agent will learn to perform a specific given task. This would allow to design a reward function that complies with the prior information induced by the kernel, or conversely to select a kernel which is suitable to identify the evolution of the chosen reward function. By contrast, this is impossible when dealing with the state transition function, which leaves no room for intervention.

The major drawback of this procedure is the computational complexity, which corresponds to the one of kernel methods. In light of this, there are already a number of methods in the vast literature on this topic which can be implemented to partially overcome this limitation. One possible extension of the contents in this work would be to design a dimensionality reduction method based on the properties of the Koopman operator. The peculiarity of dealing with a linear operator makes the interpretation of its spectrum immediate, so that a meaningful choice can be made of the subspace in which the dynamics are approximated.

The other important result is the characterisation of the trade-off in the choice of the step-size for the discretisation of a continuous-time system, along with the derivation of the optimal value in the case of a limited budget of datapoints. Although the analysis holds only for the case of linear system, it has been shown empirically that a similar trade-off is also exhibited by general nonlinear systems. The characterisation of the order in which the optimal step-size scales with the data budget is an important finding, especially if the latter holds approximately for general unknown nonlinear system. The optimal step-size can indeed be inferred from numerical simulations using a small data budget, in such a way that an approximately optimal step-size is available for the actual experiment with at full capacity. This fundamental trade-off concerning stochastic dynamical systems has been neglected from the literature, but it has great impact on the estimation of any quantity related with the dynamics. A meaningful follow up of this work would be to extend the analysis to standard system identification, in which the performance of the reconstruction of the actual dynamics should be investigated with respect to the sampling time. Another interesting line of research would be to understand if a similar reasoning applies also to different estimation techniques for the value-function, like the Temporal Difference method, which is one of the most successful in Reinforcement Learning.

A

Appendix

A.1 The Riemann Sum Approximation

The Riemann sum approximation is a standard argument that is reproduced here for completeness. Let $g : [0, T] \to \mathbb{R}$ be a continuously differentiable function. Assume that the task is to approximate the integral $\int_0^T g(t)\mathrm{d}t$ using the Riemann sum over $N = T/\Delta t$ elements, $\sum_{k=0}^{N-1} \Delta t\, g(k\Delta t)$.
The difference is readily computed up to first order as follows:

$$\mathbb{D} = \int_0^T g\left(t\right) \mathrm{d}t - \sum_{k=0}^{N-1} \Delta t\, g\left(k\Delta t\right) \tag{A.1}$$

$$= \sum_{k=0}^{N-1} \int_{k\Delta t}^{(k+1)\Delta t} g\left(t\right) - g\left(k\Delta t\right) \mathrm{d}t \tag{A.2}$$

$$\leq \sum_{k=0}^{N-1} \int_0^h g'\left(kh\right)\ t + O\left(t^2\right) \mathrm{d}t \tag{A.3}$$

$$= \frac{1}{2} \sum_{k=0}^{N-1} \left(g'\left(k\Delta t\right) \Delta t^2 + O\left(\Delta t^3\right)\right) \tag{A.4}$$

A naive bound is obtained as $\mathbb{D} \leq \frac{1}{2} N \Delta t^2 \left\|g'\right\|_\infty + O\left(N\Delta t^3\right)$. Translated to a squared error, this explains the dependency $O\left(N^2\Delta t^4\right) = O\left(T^2\Delta t^2\right)$.
In the case of discounting, let $g(t) = \gamma^t f(t)$ and $g'(t) = \gamma^t \left(f(t) + f'(t)\right)$. Hence, the previous display leads to the bound

$$\mathbb{D} \leq \frac{1}{2} \sum_{k=0}^{N-1} \gamma^{k\Delta t} \Big(\Delta t^2 \|f(t) + f'(t)\|_\infty + O(\Delta t^3)\Big) \tag{A.5}$$

$$= \frac{\Delta t^2 (1 - \gamma^{N\Delta t}) \left\|f(t) + f'(t)\right\|_\infty}{2(1 - \gamma^{\Delta t})} + O(\Delta t^3)\,. \tag{A.6}$$

Overall, the squared error is now $O\left(\Delta t^4 \left(1-\gamma^T\right)^2 / \left(1-\gamma^{\Delta t}\right)^2\right) = O\left(\Delta t^2 / \log\left(1/\gamma\right)\right)$. Note that this alone does not explain the improvement of the order from Δt^2 to Δt^4, which requires that also $f(t)$ is decaying fast enough.

A.2 Moment Calculations

Recall that the solution of the SDE in (3.9), with $x\left(0\right) = 0$, takes the following form:

$$x\left(t\right) = \overline{\sigma} \int_0^t e^{a(t-s)} \mathrm{d}W\left(s\right). \tag{A.7}$$

A significant part of finding the mean-squared error of the Monte-Carlo estimator is the computation of the moments $\mathbb{E}\left[x^2\left(t\right)\right], \mathbb{E}\left[x^4\left(t\right)\right]$ and $\mathbb{E}\left[x^2\left(s\right)x^2\left(t\right)\right]$ when $s \leq t$.

Lemma A.2.1. *Let $x(t)$ be the solution of (3.9). The second moment of the state variable is*

$$\mathbb{E}\left[x^2(t)\right] = \frac{\overline{\sigma}^2}{2u}\left(e^{2ut} - 1\right). \tag{A.8}$$

The forth moment yields:

$$\mathbb{E}\left[x^4\left(t\right)\right] = \frac{3\overline{\sigma}^4}{4u^2}\left(e^{2ut} - 1\right)^2 \tag{A.9}$$

Assuming that $s \leq t$, it holds:

$$\mathbb{E}\left[x^2(s)x^2(t)\right] = \frac{\overline{\sigma}^4}{4u^2}(e^{2us} - 1)e^{2ut}\left\{(e^{-2us} - e^{-2ut}) + 3(1 - e^{-2us})\right\}. \tag{A.10}$$

Proof. The proof is divided in three steps.

1. Starting with the second moment $\mathbb{E}\left[x^2\left(t\right)\right]$.

$$\mathbb{E}\left[x^2\left(t\right)\right] = \overline{\sigma}^2 e^{2ut}\,\mathbb{E}\left[\left(\int_0^t e^{-us}\mathrm{d}W\left(s\right)\right)^2\right] = \overline{\sigma}^2 e^{2ut}\int_0^t e^{-2us}\mathrm{d}s = \frac{\overline{\sigma}^2}{2u}(e^{2ut} - 1) \tag{A.11}$$

The calculation makes use of the Itô isometry, which can be stated as:

$$\mathbb{E}\left[\left(\int_0^t z(s)\mathrm{d}W\left(s\right)\right)^2\right] = \mathbb{E}\left[\int_0^t z(s)^2\mathrm{d}s\right], \tag{A.12}$$

for any stochastic process $z\left(\cdot\right)$ adapted to the filtration induced by the Wiener

process $W(\cdot)$.

2. Next, compute $\mathbb{E}\left[x^4(t)\right]$ through Itô's integral. Define $y(t) := \int_0^t e^{-ub}\mathrm{d}W(b)$, so that $\mathrm{d}y(t) = e^{-ut}\mathrm{d}W(t)$. Thus,

$$\mathrm{d}f(y(t)) = f'(y(t))\,\mathrm{d}y(t) + \frac{1}{2}f''(y(t))\,(\mathrm{d}y(t))^2 \tag{A.13}$$

$$= f'(y(t))\,e^{-ut}\mathrm{d}w(t) + \frac{1}{2}f''(y(t))\,e^{-2ut}\mathrm{d}t, \tag{A.14}$$

for any $f(\cdot)$.
By choosing $f(y) = y^4$:

$$f'(y) = 4y^3 \quad \text{and} \quad f''(y) = 12y^2. \tag{A.15}$$

Therefore, by integration and taking the expectation:

$$\mathbb{E}\left[f(y(t))\right] = \mathbb{E}\left[\int_0^t f'(y(b))\,e^{-ub}\mathrm{d}W(b)\right] + \frac{1}{2}\,\mathbb{E}\left[\int_0^t f''(y(b))\,e^{-2ub}\mathrm{d}b\right] \tag{A.16}$$

$$= \underbrace{\mathbb{E}\left[\int_0^t 4\left(\int_0^b e^{-uv}\mathrm{d}w(v)\right)^3 e^{-ub}\mathrm{d}W(b)\right]}_{=0} + \frac{1}{2}\,\mathbb{E}\left[\int_0^t 12\left(\int_0^b e^{-uv}\mathrm{d}W(v)\right)^2 e^{-2ub}\mathrm{d}b\right] \tag{A.17}$$

$$= 6\,\mathbb{E}\left[\int_0^t \left(\int_0^b e^{-uv}e^{-ub}\mathrm{d}W(v)\right)^2 \mathrm{d}b\right] \tag{A.18}$$

$$= 6\int_0^t \mathbb{E}\left[\left(\int_0^b e^{-uv}e^{-ub}\mathrm{d}W(v)\right)^2\right]\mathrm{d}b \qquad \text{(Itô isometry)} \tag{A.19}$$

$$= 6\int_0^t \int_0^b e^{-2uv}e^{-2ub}\mathrm{d}v\mathrm{d}b \tag{A.20}$$

$$= \int_0^t e^{-2ub}\frac{1}{2u}\left(1 - e^{-2ub}\right)\mathrm{d}b \tag{A.21}$$

$$= \frac{3}{4u^2}\left(1 - e^{-2ut}\right)^2 \tag{A.22}$$

From (A.7) it holds $x(t) = \overline{\sigma}e^{ut}y(t)$ so that the second part of the lemma follows.

3. Lastly, $\mathbb{E}\left[x^2(s)\,x^2(t)\right]$ is computed for $s \leq t$:

$$\mathbb{E}\left[x^2(s)\,x^2(t)\right] = \overline{\sigma}^4 e^{2u(s+t)}\,\mathbb{E}\left[\left(\int_0^s e^{-ub}\mathrm{d}W(b)\right)^2\left(\int_0^t e^{-ub}\mathrm{d}W(b)\right)^2\right] \tag{A.23}$$

$$= \overline{\sigma}^4 e^{2u(s+t)}\mathbb{E}\left[\left(\int_0^s e^{-ub}\mathrm{d}w(b)\right)^2\left(\int_0^s e^{-ub}\mathrm{d}w(b)+\int_s^t e^{-ub}\mathrm{d}w(b)\right)^2\right] \tag{A.24}$$

$$= \overline{\sigma}^4 e^{2u(s+t)}\Bigg\{\underbrace{\mathbb{E}\left[\left(\int_0^s e^{-ub}\mathrm{d}w(b)\right)^4\right]}_{\text{(i)}} + \underbrace{\mathbb{E}\left[\left(\int_0^s e^{-ub}\mathrm{d}w(b)\right)^2\right]\mathbb{E}\left[\left(\int_s^t e^{-ub}\mathrm{d}w(b)\right)^2\right]}_{\text{(ii)}}\Bigg\} \tag{A.25}$$

Note that (i) has been computed before. For (ii) it holds:

$$\mathbb{E}\left[\left(\int_0^s e^{-ub}\mathrm{d}W(b)\right)^2\right] = \int_0^s e^{-2ub}\mathrm{d}W(b) \tag{A.26}$$

$$= \frac{1}{2u}(1-e^{-2us}) \tag{A.27}$$

and

$$\mathbb{E}\left[\left(\int_s^t e^{-ub}\mathrm{d}w(b)\right)^2\right] = \int_s^t e^{-2ub}\mathrm{d}W(b)$$

$$= \frac{1}{2u}\left(e^{-2us}-e^{-2ut}\right).$$

Therefore, assuming $s \leq t$, it holds that (it is necessary to use Itô integration in this case):

$$\mathbb{E}\left[x^2(s)\,x^2(t)\right] = \overline{\sigma}^4 e^{2u(s+t)}\left\{\frac{1}{4u^2}\left(1-e^{-2us}\right)\left(e^{-2us}-e^{-2ut}\right)+\frac{3}{4u^2}\left(1-e^{-2us}\right)^2\right\} \tag{A.28}$$

$$= \frac{\overline{\sigma}^4}{4u^2}(e^{2ut}-1)e^{2us}\left\{(e^{-2ut}-e^{-2us})+3(1-e^{-2ut})\right\}, \tag{A.29}$$

which concludes the proof. □

A.3 Computation of the Mean-Squared Error

In the following the proofs of Theorems 3.3.1 and 3.3.6 are presented in detail, along with the precise characterisation of the mean-squared error for the finite-horizon and

discounted setting.

A.3.1 Finite-horizon, undiscounted

The proof of Theorem 3.3.1 is as follows.

Proof. First note that

$$\mathbb{E}\left[\hat{V}_M(\Delta t)\right] = \frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\mathbb{E}\left[x_i^2(k\Delta t)\right] = \Delta t\sum_{k=0}^{N-1}\mathbb{E}\left[x^2(k\Delta t)\right], \tag{A.30}$$

where, to ease the notation, $x(t) = x_1(t)$. Next, expand the mean-squared error as

$$\begin{aligned}
\mathbb{E}\left[(\hat{V}_M(\Delta t) - V_T)^2\right] &= \mathbb{E}\left[\hat{V}_M^2(\Delta t)\right] - 2V_T\mathbb{E}\left[\hat{V}_M(\Delta t)\right] + V_T^2 && \text{(A.31)}\\
&= \frac{\Delta t^2}{M^2}\,\mathbb{E}\left[\left(\sum_{i=1}^{M}\sum_{k=0}^{N-1}x_i^2(k\Delta t)\right)^2\right] - 2V_T\,\mathbb{E}\left[\hat{V}_M(\Delta t)\right] + V_T^2 && \text{(A.32)}\\
&= \frac{\Delta t^2}{M^2}\sum_{i,j=1}^{M}\sum_{k,l=0}^{N-1}\mathbb{E}\left[x_i^2(k\Delta t)x_j^2(l\Delta t)\right] - 2V_T\,\mathbb{E}\left[\hat{V}_M(\Delta t)\right] + V_T^2 && \text{(A.33)}\\
&= \frac{\Delta t^2}{M}\sum_{k,l=0}^{N-1}\mathbb{E}\left[x^2(k\Delta t)x^2(l\Delta t)\right] + \frac{M^2-M}{M^2}\,\mathbb{E}\left[\hat{V}_M(\Delta t)\right]^2 - 2V_T\mathbb{E}\left[\hat{V}_M(\Delta t)\right] + V_T^2\,. && \text{(A.34)}
\end{aligned}$$

For the last equality, note that $\mathbb{E}\left[\hat{V}_M(\Delta t)\right]^2 = \Delta t^2\sum_{k,l=0}^{N-1}\mathbb{E}\left[x^2(k\Delta t)\right]\mathbb{E}\left[x^2(l\Delta t)\right]$. It now remains to compute the expressions with the summation. By Lemma A.2.1 the second moment of the state variable is:

$$\mathbb{E}\left[x^2(t)\right] = \frac{\overline{\sigma}^2}{2u}\left(e^{2ut} - 1\right). \tag{A.35}$$

Assuming that $s \le t$, the forth moments can be derived from the same lemma as:

$$\mathbb{E}\left[x^2(s)x^2(t)\right] = \frac{\overline{\sigma}^4}{4u^2}(e^{2us} - 1)e^{2ut}\left\{(e^{-2us} - e^{-2ut}) + 3(1 - e^{-2us})\right\}. \tag{A.36}$$

Note that by symmetry, a similar expression follows for $s \ge t$.
Using these expressions, for the expected cost it holds

$$V_T = \int_0^T \mathbb{E}\left[x^2(t)\right]dt = \frac{\overline{\sigma}^2}{2u}\int_0^T\left(e^{2ut} - 1\right)dt = \frac{\overline{\sigma}^2}{2u}\left(\frac{e^{2uT} - 1}{2u} - T\right). \tag{A.37}$$

Note that a similar expression was previously obtained in (Bijl et al., 2016, Theorem 3).

Next, the expected estimated cost is given by:

$$\mathbb{E}\left[\hat{V}_M(\Delta t)\right] = \Delta t \sum_{k=0}^{N-1} \mathbb{E}\left[x^2(k\Delta t)\right] = \frac{\overline{\sigma}^2 \Delta t}{2u} \sum_{k=0}^{N-1} \left(e^{2uk\Delta t} - 1\right)$$
$$= \frac{\overline{\sigma}^2 \Delta t}{2u} \left[\frac{1 - e^{2uT}}{1 - e^{2u\Delta t}} - N\right] \tag{A.38}$$

Lastly, it remains to compute the summation:

$$\frac{\Delta t^2}{M} \sum_{k,l=0}^{N-1} \mathbb{E}\left[x^2(k\Delta t)x^2(l\Delta t)\right] = \frac{2\Delta t^2}{M} \sum_{k<l}^{N-1} \mathbb{E}\left[x^2(k\Delta t)x^2(l\Delta t)\right] + \frac{\Delta t^2}{M} \sum_{k=0}^{N-1} \mathbb{E}\left[x^4(k\Delta t)\right]$$
$$= \frac{\overline{\sigma}^4 T \left(\Delta t^2 \left(e^{2uT} - 1\right)\left(8e^{2u\Delta t} + 3e^{2uT} + 1\right) + T^2 \left(e^{2u\Delta t} - 1\right)^2 - 2\Delta t T \left(e^{2u\Delta t} - 1\right)\left(e^{2u\Delta t} + 5e^{2uT}\right)\right)}{4u^2 \mathbb{B} \Delta t \left(e^{2u\Delta t} - 1\right)^2} \tag{A.39}$$

The last equality is a cumbersome calculation that involves nested geometric sums. The result has been verified using symbolic computation. Finally it remains to collect all terms to get the final result. □

A.3.2 Finite-horizon, discounted

Lemma A.3.1 (Finite-horizon, discounted)**.** *In the finite-horizon with a discount factor* $\gamma \in (0, 1]$ *setting, the mean-squared error of the Monte-Carlo estimator is*

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}, \gamma) = E_1(\Delta t, T, u, \gamma) + \frac{E_2(\Delta t, T, u, \gamma)}{\mathbb{B}}, \tag{A.40}$$

where

$$E_1(\Delta t, T, u, \gamma) = C_1(T,\gamma,u)\overline{\sigma}^4\Delta t^2 + C_2(T,\gamma,u)\overline{\sigma}^4\Delta t^3 + \left(\frac{1}{144} + C_3(T,\gamma,u)\right)\overline{\sigma}^4\Delta t^4 + O(\Delta t^5)\,, \tag{A.41}$$

$$E_2(\Delta t, T, u, \gamma) = \frac{\overline{\sigma}^4 T + \gamma^T C_4(T,\gamma,u)}{\log(\gamma)(u+\log(\gamma))(2u+\log(\gamma))^2\Delta t} + \gamma^T O(1)\,, \tag{A.42}$$

$$C_1(T,\gamma,u) = \frac{\gamma^{2T}\left(e^{2uT}-1\right)^2}{16u^2}\,, \tag{A.43}$$

$$C_2(T,\gamma,u) = \frac{\gamma^T\left(e^{2uT}-1\right)\left(\gamma^T\left(e^{2uT}(2u+\log(\gamma))-\log(\gamma)\right)-2u\right)}{48u^2}\,, \tag{A.44}$$

$$C_3(T,\gamma,u) = \frac{\overline{\sigma}^4\gamma^T\left[\gamma^T\left(e^{2uT}(2u+\log(\gamma))-\log(\gamma)\right)^2 - 4u\left(e^{2uT}(2u+\log(\gamma))-\log(\gamma)\right)\right]}{576u^2}\,, \tag{A.45}$$

$C_4(T,\gamma,u)$ *is some finite constant of* (T,γ,a) *that includes a factor of* γ^T . (A.46)

Proof. The proof follows the similar computations as those in the previous proof with a new expected cost as follows. In particular, using Lemma A.2.1, it holds

$$V_T = \int_0^T \gamma^t \mathbb{E}\left[x^2(t)\right] dt = \frac{\overline{\sigma}^2}{2u}\left(\frac{\gamma^T e^{2uT}-1}{\log(\gamma)+2u} - \frac{\gamma^T-1}{\log(\gamma)}\right) \tag{A.47}$$

Furthermore, the expected estimated cost is given by:

$$\mathbb{E}\left[\hat{V}_M(\Delta t)\right] = \frac{\overline{\sigma}^2\Delta t}{2u}\sum_{k=0}^{N-1}\gamma^{k\Delta t}\left(e^{2uk\Delta t}-1\right) = \frac{\overline{\sigma}^2\Delta t}{2u}\left(\frac{1-\gamma^T e^{2uT}}{1-\gamma^{\Delta t}e^{2u\Delta t}} - \frac{1-\gamma^T}{1-\gamma^{\Delta t}}\right)\,. \tag{A.48}$$

Finally, the sum containing the forth order cross-moments is

$$\frac{\Delta t^2}{M}\sum_{k,l=0}^{N-1}\gamma^{\Delta t(l+k)}\mathbb{E}\left[x^2(k\Delta t)x^2(l\Delta t)\right] = \frac{2\Delta t^2}{M}\sum_{k<l}^{N-1}\gamma^{\Delta t(l+k)}\mathbb{E}\left[x^2(k\Delta t)x^2(l\Delta t)\right] + \\ + \frac{\Delta t^2}{M}\sum_{k=0}^{N-1}\gamma^{2k\Delta t}\mathbb{E}\left[x^4(k\Delta t)\right]\,.$$

While not impossible to calculate on paper, a written derivation is beyond the scope of this work. Instead, one can rely on symbolic computation to obtain the expression and corresponding Taylor approximations. □

A.3.3 Infinite-horizon

In what follows it is presented the proof of Theorem 3.3.6.

Proof. The proof relies on the decomposition provided in (3.31). It only remains to compute the following cross term.

$$\mathbb{E}\left[\hat{V}_M(\Delta t) - V_T\right] V_{T,\infty} \tag{A.49}$$

$$= \frac{\overline{\sigma}^4}{2u}\left(\frac{\gamma^T}{\log(\gamma)} - \frac{\gamma^T e^{2uT}}{\log(\gamma)+2u}\right)\left[\frac{\Delta t}{2u}\left(\frac{1-\gamma^T e^{2uT}}{1-\gamma^{\Delta t}e^{2u\Delta t}} - \frac{1-\gamma^T}{1-\gamma^{\Delta t}}\right) - \frac{1}{2u}\left(\frac{\gamma^T e^{2uT}-1}{\log(\gamma)+2u} - \frac{\gamma^T-1}{\log(\gamma)}\right)\right] \tag{A.50}$$

$$= \frac{\overline{\sigma}^4\gamma^{2T}\left(e^{2uT}-1\right)\left(\log(\gamma)\left(e^{2uT}-1\right)-2u\right)}{8u^2\log(\gamma)\left(2u+\log(\gamma)\right)}\Delta t + \\ + \frac{\overline{\sigma}^4\gamma^T\left(2u+\log(\gamma)-e^{2uT}\log(\gamma)\right)\left(2u\left(\gamma^T e^{2uT}-1\right)+\gamma^T\log(\gamma)\left(e^{2uT}-1\right)\right)}{48u^2\log(\gamma)\left(2u+\log(\gamma)\right)}\Delta t^2 + O(\Delta t^3)\,. \tag{A.51}$$

Thus, the mean-squared error $\mathrm{MSE}_\infty(\Delta t, \mathbb{B}, T, \gamma) = \mathbb{E}\left[(\hat{V}_M(\Delta t) - V_\infty)^2\right]$ is obtained by combining the above computation with (3.29) and Lemma A.3.1. □

A.4 Vector case analysis

In this section, the proofs for the analysis of the vector case are provided.

A.4.1 Finite-horizon, undiscounted

The proof of Theorem 3.3.4 is detailed below.

Proof. Consider the d-dimensional system that the solution of the trajectory of $x(t)$ is

$$x(t) = \overline{\sigma}\int_0^t e^{\tilde{A}(t-s)}\mathrm{d}W(t)\,. \tag{A.52}$$

Since $\tilde{A}$ is a diagonalisable matrix, it is possible to decompose $\tilde{A}$ as $\tilde{A} = P^{-1}\tilde{D}P$, where P is a invertible matrix (not necessarily orthogonal) and $\tilde{D}$ is a diagonal matrix whose diagonal entries $(\lambda_1, \cdots, \lambda_n)$ corresponds to the eigenvalues of the matrix $\tilde{A}$. Thus it is possible to decompose the matrix exponential of $\tilde{A}$ as:

$$e^{\tilde{A}t} = P^{-1}e^{\tilde{D}t}P\,. \tag{A.53}$$

Define the diagonalised process $\tilde{x}(\cdot)$ as:

$$Px(t) = P\overline{\sigma}\int_0^t e^{\tilde{A}(t-s)}\mathrm{d}W(s) \tag{A.54}$$

$$= \overline{\sigma}PP^{-1}\int_0^t e^{\tilde{D}(t-s)}P\mathrm{d}W(s) \tag{A.55}$$

$$= \overline{\sigma}\int_0^t e^{\tilde{D}(t-s)}\mathrm{d}\tilde{W}(s) =: \tilde{x}(t) \tag{A.56}$$

where $\tilde{W}(s)$ is a Wiener process (with dependent components when P is not orthogonal). This implies that $x(\cdot) = P^{-1}\tilde{x}(\cdot)$.
To see $\tilde{x}_i(t)$ clearly, denote $P = [p_{ij}]_{i,j=1}^n$, and $\tilde{x}_i(t) = (\phi_1^{(i)}(t), \cdots, \phi_n^{(i)}(t))^\top$, then $\phi_l^{(i)}(t) = \sum_{j=1}^n p_{lj}\overline{\sigma}\int_0^t e^{\lambda_l(t-s)}\mathrm{d}W_j^{(i)}(s)$ for each $l \in \{1, \cdots, n\}$. Specifically, in the latter expression, $W_j^{(i)}(s)$ are independent Wiener processes for different i or j. Correspondingly, $\tilde{x}(t) = (\phi_1(t), \cdots, \phi_n(t))^\top$, and $\phi_l(t) = \sum_{j=1}^n p_{lj}\overline{\sigma}\int_0^t e^{\lambda_l(t-s)}\mathrm{d}W_j(s)$ for each $l \in \{1, \cdots, n\}$, where $W_j(s)$ are independent Wiener processes for different j. By trace operation, we can rewrite $\hat{V}_M(\Delta t)$ as follows:

$$\hat{V}_M(\Delta t) = \frac{1}{M}\sum_{i=1}^M\sum_{k=0}^{N-1}\Delta t x(t_k)^\top \tilde{Q}x(t_k) \tag{A.57}$$

$$= \mathrm{Tr}\left(\frac{1}{M}\sum_{i=1}^M\sum_{k=0}^{N-1}\Delta t\tilde{x}(t_k)^\top P^{-\top}\tilde{Q}P^{-1}\tilde{x}(t_k)\right) \tag{A.58}$$

$$= \mathrm{Tr}\left(P^{-\top}\tilde{Q}P^{-1}\hat{\mathcal{V}}_M(\Delta t)\right), \tag{A.59}$$

where $\hat{\mathcal{V}}_M(\Delta t) = \frac{1}{M}\sum_{i=1}^M\sum_{k=0}^{N-1}\Delta t\tilde{x}(t_k)\tilde{x}(t_k)^\top \in \mathbb{R}^{n\times n}$.
Similarly, $V_T = \mathrm{Tr}\left(P^{-\top}\tilde{Q}P^{-1}\mathcal{V}_T\right)$, where $\mathcal{V}_T = \int_0^T \mathbb{E}\left[\tilde{x}(t)\tilde{x}(t)^\top\right]\mathrm{d}t$. Therefore, the $\mathrm{MSE}_T(\Delta t, \mathbb{B})$ can be written as

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}) = \mathbb{E}\left[\left(\hat{V}_M(\Delta t) - V_T\right)^2\right] \tag{A.60}$$

$$= \mathbb{E}\left[\mathrm{Tr}\left(P^{-\top}\tilde{Q}P^{-1}\left(\hat{\mathcal{V}}_M(\Delta t) - \mathcal{V}_T\right)\right)^2\right]. \tag{A.61}$$

For the sake of notation, denote matrix $P^{-\top}\tilde{Q}P^{-1} =: G = [g_{lj}]_{l,j=1}^n$ and $\hat{\mathcal{V}}_M(\Delta t) - \mathcal{V}_T =: C = [c_{lj}]_{l,j=1}^n$. Noting the fact that

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}) = \mathbb{E}\left[\left(\sum_{l,j} g_{jl}c_{lj}\right)^2\right] = \sum_{l_1,j_1,l_2,j_2} g_{j_1l_1}g_{j_2l_2}\mathbb{E}\left[c_{l_1j_1}c_{l_2j_2}\right], \tag{A.62}$$

it is sufficient to find MSE_T by only computing $\mathbb{E}\left[c_{l_1 j_1} c_{i_2 j_2}\right]$.
The following expectations are introduced, which will be used in further computations. For any $s \leq t$:

$$\mathbb{E}\left[\int_0^t e^{\lambda_1(t-b)}\mathrm{d}W(b)\int_0^s e^{\lambda_2(s-b)}\mathrm{d}W(b)\right] = \frac{e^{\lambda_1 t+\lambda_2 s}}{\lambda_1+\lambda_2}\left(1-e^{-(\lambda_1+\lambda_2)s}\right); \tag{A.63}$$

$$\begin{aligned}
&\mathbb{E}\left[\int_0^s e^{\lambda_1(s-b)}\mathrm{d}W(b)\int_0^s e^{\lambda_2(s-b)}\mathrm{d}W(b)\int_0^t e^{\lambda_3(t-b)}\mathrm{d}W(b)\int_0^t e^{\lambda_4(t-b)}\mathrm{d}W(b)\right] \\
&= e^{(\lambda_1+\lambda_2)s+(\lambda_3+\lambda_4)t}\left[\frac{1}{(\lambda_1+\lambda_2)(\lambda_3+\lambda_4)}\left(1-e^{-(\lambda_1+\lambda_2)s}\right)\left(1-e^{-(\lambda_3+\lambda_4)s}\right)+\right. \\
&\quad+\frac{1}{(\lambda_1+\lambda_3)(\lambda_2+\lambda_4)}\left(1-e^{-(\lambda_1+\lambda_3)s}\right)\left(1-e^{-(\lambda_2+\lambda_4)s}\right)+ \\
&\quad+\frac{1}{(\lambda_1+\lambda_4)(\lambda_2+\lambda_3)}\left(1-e^{-(\lambda_1+\lambda_4)s}\right)\left(1-e^{-(\lambda_2+\lambda_3)s}\right)+ \\
&\quad\left.+\frac{1}{(\lambda_1+\lambda_2)(\lambda_3+\lambda_4)}\left(1-e^{-(\lambda_1+\lambda_2)s}\right)\left(e^{-(\lambda_3+\lambda_4)s}-e^{-(\lambda_3+\lambda_4)t}\right)\right];
\end{aligned} \tag{A.64}$$

$$\int_0^T \mathbb{E}\left[\int_0^t e^{\lambda_1(t-b)}\mathrm{d}W(b)\int_0^s e^{\lambda_2(-b)}\mathrm{d}W(b)\right]\mathrm{d}t = \frac{e^{(\lambda_1+\lambda_2)T}-1-(\lambda_1+\lambda_2)T}{(\lambda_1+\lambda_2)^2}. \tag{A.65}$$

By using the definitions of $\hat{\mathcal{V}}_M(\Delta t)$ and $\mathcal{V}_T$, it is trivial to see that for any $l, j \in \{1, \cdots, n\}$ it holds:

$$c_{lj} = \frac{1}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\Delta t\phi_l^{(i)}(k\Delta t)\phi_j^{(i)}(k\Delta t) - \int_0^T \mathbb{E}\left[\phi_l(t)\phi_j(t)\right]\mathrm{d}t \tag{A.66}$$

$$\begin{aligned}
&= \frac{\Delta t\overline{\sigma}^2}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\sum_{\alpha=1}^{n}p_{l\alpha}\int_0^{k\Delta t}e^{\lambda_l(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\sum_{\alpha=1}^{n}p_{j\alpha}\int_0^{k\Delta t}e^{\lambda_j(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)- \\
&- \overline{\sigma}^2\int_0^T\mathbb{E}\left[\left(\sum_{\alpha=1}^{n}p_{l\alpha}\int_0^t e^{\lambda_l(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\sum_{\alpha=1}^{n}p_{j\alpha}\int_0^t e^{\lambda_j(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t
\end{aligned} \tag{A.67}$$

$$\begin{aligned}
&= \sum_{\alpha=1}^{n}p_{l\alpha}p_{j\alpha}\left[\frac{\Delta t\overline{\sigma}^2}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t}e^{\lambda_l(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t}e^{\lambda_j(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)-\right. \\
&\left.- \overline{\sigma}^2\int_0^T\mathbb{E}\left[\left(\int_0^t e^{\lambda_l(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_j(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t\right]+ \\
&+ \sum_{\alpha\neq\beta}p_{l\alpha}p_{j\beta}\left[\frac{h\overline{\sigma}^2}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t}e^{\lambda_l(k\Delta t-s)}\mathrm{d}w_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t}e^{\lambda_j(k\Delta t-s)}\mathrm{d}w_\beta^{(i)}(s)\right)\right],
\end{aligned} \tag{A.68}$$

where the last equation is due to the fact that for $\alpha \neq \beta$ it holds:

$$\mathbb{E}\left[\left(\int_0^t e^{\lambda_l(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_j(t-s)}\mathrm{d}W_\beta(s)\right)\right] = 0\,. \tag{A.69}$$

Thus, for any $l_1, l_2, j_1, j_2 \in \{1, \cdots, n\}$, the cross expectations are as follows:

$$\begin{aligned}\mathbb{E}\left[c_{l_1 j_1} c_{l_2, j_2}\right] = &\sum_{\alpha=1}^{n} p_{l_1\alpha} p_{j_1\alpha} p_{l_2\alpha} p_{j_2\alpha} \overline{\sigma}^4 \mathcal{I}_1\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha\right) + \\ &+ \sum_{\alpha\neq\beta}^{n} p_{l_1\alpha} p_{j_1\alpha} p_{l_2\beta} p_{j_2\beta} \overline{\sigma}^4 \mathcal{I}_2\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right) + \\ &+ \sum_{\alpha\neq\beta}^{n} p_{l_1\alpha} p_{j_1\beta} p_{l_2\alpha} p_{j_2\beta} \overline{\sigma}^4 \mathcal{I}_3\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right)\,,\end{aligned} \tag{A.70}$$

where

$$\mathcal{I}_1\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha\right) \tag{A.71}$$

$$\begin{aligned}&= \mathbb{E}\left\{\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t} e^{\lambda_{l_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t} e^{\lambda_{j_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right) - \right.\right. \\ &- \left.\int_0^T \mathbb{E}\left[\left(\int_0^t e^{\lambda_{l_1}(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_{j_1}(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t\right] \times \\ &\times \left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t} e^{\lambda_{l_2}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t} e^{\lambda_{j_2}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right) - \right. \\ &\left.\left. - \int_0^T \mathbb{E}\left[\left(\int_0^t e^{\lambda_{l_2}(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_{j_2}(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t\right]\right\}\,,\end{aligned} \tag{A.72}$$

and

$$\mathcal{I}_2\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right) \tag{A.73}$$

$$\begin{aligned}&= \mathbb{E}\left\{\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t} e^{\lambda_{l_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t} e^{\lambda_{j_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right) - \right.\right. \\ &- \left.\int_0^T \mathbb{E}\left[\left(\int_0^t e^{\lambda_{l_1}(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_{j_1}(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t\right] \times \\ &\times \left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t} e^{\lambda_{l_2}(k\Delta t-s)}\mathrm{d}W_\beta^{(i)}(s)\right)\left(\int_0^{k\Delta t} e^{\lambda_{j_2}(k\Delta t-s)}\mathrm{d}W_\beta^{(i)}(s)\right) - \right. \\ &\left.\left. - \int_0^T \mathbb{E}\left[\left(\int_0^t e^{\lambda_{l_2}(t-s)}\mathrm{d}W_\beta(s)\right)\left(\int_0^t e^{\lambda_{j_2}(t-s)}\mathrm{d}W_\beta(s)\right)\right]\mathrm{d}t\right]\right\}\end{aligned} \tag{A.74}$$

and

$$\mathcal{I}_3\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha,\beta\right) \tag{A.75}$$

$$= \mathbb{E}\left\{\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t} e^{\lambda_{l_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t} e^{\lambda_{j_1}(k\Delta t-s)}\mathrm{d}W_\beta^{(i)}(s)\right)\right]\times\right.$$

$$\left.\times\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\left(\int_0^{k\Delta t} e^{\lambda_{l_2}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t} e^{\lambda_{j_2}(k\Delta t-s)}\mathrm{d}W_\beta^{(i)}(s)\right)\right]\right\}. \tag{A.76}$$

Note that $W_\alpha^{(i)}$ and $W_\beta^{(i)}$ are independent for $\alpha\neq\beta$.
By using the expectations (A.63) and (A.65), the expression $\mathcal{I}_2\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha,\beta\right)$ can be written as

$$\mathcal{I}_2\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha,\beta\right) \tag{A.77}$$

$$= \left[\frac{\Delta t}{(\lambda_{l_1}+\lambda_{j_1})}\left(\frac{1-e^{(\lambda_{l_1}+\lambda_{j_1})T}}{1-e^{(\lambda_{l_1}+\lambda_{j_1})\Delta t}}-\frac{T}{\Delta t}\right)-\frac{1}{(\lambda_{l_1}+\lambda_{j_1})^2}\left(e^{(\lambda_{l_1}+\lambda_{j_1})T}-1-(\lambda_{l_1}+\lambda_{j_1})T\right)\right]\times$$

$$\times\left[\frac{\Delta t}{(\lambda_{l_2}+\lambda_{j_2})}\left(\frac{1-e^{(\lambda_{l_2}+\lambda_{j_2})T}}{1-e^{(\lambda_{l_2}+\lambda_{j_2})h}}-\frac{T}{\Delta t}\right)-\frac{1}{(\lambda_{l_2}+\lambda_{j_2})^2}\left(e^{(\lambda_{l_2}+\lambda_{j_2})T}-1-(\lambda_{l_2}+\lambda_{j_2})T\right)\right]. \tag{A.78}$$

In the following computations, constants that are not depending on Δt, T or $\mathbb{B}$ will be denoted as $\bar{C}$ and $C(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2})$.
The expectation $\mathcal{I}_1\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha\right)$ is computed exactly the same way as in the proof of Theorem 3.3.1 by using the expectation results (A.63) and (A.64). Notice that the expectation result (A.63) (when $s=t$) has the same order in t as the expectation (A.8). Moreover, the two expectations (A.64) and (A.10) have the same orders in s and t. Thus, $\mathcal{I}_1\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha\right)$ has the same orders in Δt, T and $\mathbb{B}$ as the scalar MSE, i.e.:

$$\mathcal{I}_1\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha\right) = \left(\bar{C}_1+C_1\left(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2}\right)O(T)\right)T^2\Delta t^2+$$

$$+O(\Delta t^3)+\left(\bar{C}_2+C_2\left(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2}\right)O(T)\right)\frac{T^5}{\Delta t\mathbb{B}}+O\left(\frac{1}{\mathbb{B}}\right) \tag{A.79}$$

The expectation $\mathcal{I}_2\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right)$ can be computed directly and yields:

$$\mathcal{I}_2\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right) = \frac{\left(e^{(\lambda_{l_1}+\lambda_{j_1})T}-1\right)\left(e^{(\lambda_{l_2}+\lambda_{j_2})T}-1\right)\Delta t^2}{4\left(\lambda_{l_1}+\lambda_{j_1}\right)\left(\lambda_{l_2}+\lambda_{j_2}\right)} + O(\Delta t^3) \tag{A.80}$$

$$= \left(\frac{1}{4}T^2 + C_3(\lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2})O(T^3))\right)\Delta t^2 + O(\Delta t^3)\,. \tag{A.81}$$

The expectation $\mathcal{I}_3\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right)$ can be computed as follows:

$$\mathcal{I}_3\left(M, \Delta t, T, \lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}, \alpha, \beta\right) \tag{A.82}$$

$$\begin{aligned}
&= \frac{\Delta t^2}{M}\sum_{k=0}^{n}\frac{\left(e^{(\lambda_{l_1}+\lambda_{l_2})k\Delta t}-1\right)\left(e^{(\lambda_{j_1}+\lambda_{j_2})k\Delta t}-1\right)\Delta t^2}{\left(\lambda_{l_1}+\lambda_{l_2}\right)\left(\lambda_{j_1}+\lambda_{j_2}\right)} + \\
&+ \frac{\Delta t^2}{M}\sum_{k<q}\frac{e^{\lambda_{l_1}k\Delta t+\lambda_{l_2}q\Delta t+\lambda_{j_1}k\Delta t+\lambda_{j_2}q\Delta t}}{\left(\lambda_{l_1}+\lambda_{l_2}\right)\left(\lambda_{j_1}+\lambda_{j_2}\right)}\left(1-e^{-(\lambda_{l_1}+\lambda_{l_2})k\Delta t}\right)\left(1-e^{-(\lambda_{j_1}+\lambda_{j_2})k\Delta t}\right) + \\
&+ \frac{\Delta t^2}{M}\sum_{k<q}\frac{e^{\lambda_{l_1}q\Delta t+\lambda_{l_2}k\Delta t+\lambda_{j_1}q\Delta t+\lambda_{j_2}k\Delta t}}{\left(\lambda_{l_1}+\lambda_{l_2}\right)\left(\lambda_{j_1}+\lambda_{j_2}\right)}\left(1-e^{-(\lambda_{l_1}+\lambda_{l_2})k\Delta t}\right)\left(1-e^{-(\lambda_{j_1}+\lambda_{j_2})k\Delta t}\right)
\end{aligned} \tag{A.83}$$

$$= \left(\bar{C}_4 + C_4\left(\lambda_{l_1}, \lambda_{j_1}, \lambda_{l_2}, \lambda_{j_2}\right)O(T)\right)\frac{T^5}{\Delta t\mathbb{B}} + O\left(\frac{1}{\mathbb{B}}\right)\,. \tag{A.84}$$

Thus, the final result is obtained by the expression of MSE in (A.62), (A.70) and the above computations. Again, one can rely on symbolic computation to obtain the expression and corresponding Taylor approximations. □

The extension from Theorem 3.3.4 to the discounted finite-horizon results can be done in the same way as in the proof above (and adding the discount factor γ in $\hat{V}_M$) by using the expectation cost for any λ_1 and λ_2:

$$\int_0^T \gamma^t \mathbb{E}\left[\int_0^t e^{\lambda_1(t-b)}\mathrm{d}W(b)\int_0^s e^{\lambda_2(-b)}\mathrm{d}W(b)\right]\mathrm{d}t = \frac{1}{(\lambda_1+\lambda_2)}\left(\frac{\gamma^T e^{(\lambda_1+\lambda_2)T}-1}{\log(\gamma)+(\lambda_1+\lambda_2)} - \frac{\gamma^T-1}{\log(\gamma)}\right)\,. \tag{A.85}$$

A.4.2 Corollary for infinite-horizon, discounted

In the following the proof for Corollary 3.3.7 is detailed.

Proof. The proof follows a similar argument as the proof of Theorem 3.3.4 and the proof

of Theorem 3.3.6.

Continuing from (A.70), in infinite-horizon discounted setting, it holds:

$$\mathcal{I}_1\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,\alpha\right) \tag{A.86}$$

$$\begin{aligned}
&= \mathbb{E}\left\{\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\gamma^{k\Delta t}\left(\int_0^{k\Delta t}e^{\lambda_{l_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t}e^{\lambda_{j_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right) - \right.\right.\\
&- \left.\int_0^\infty\gamma^t\mathbb{E}\left[\left(\int_0^t e^{\lambda_{l_1}(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_{j_1}(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t\right]\times\\
&\times\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\gamma^{k\Delta t}\left(\int_0^{k\Delta t}e^{\lambda_{l_2}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t}e^{\lambda_{j_2}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right) - \right.\\
&- \left.\left.\int_0^\infty\gamma^t\mathbb{E}\left[\left(\int_0^t e^{\lambda_{l_2}(t-s)}\mathrm{d}W_\alpha(s)\right)\left(\int_0^t e^{\lambda_{j_2}(t-s)}\mathrm{d}W_\alpha(s)\right)\right]\mathrm{d}t\right]\right\},
\end{aligned} \tag{A.87}$$

and

$$\mathcal{I}_2\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,\alpha,\beta\right) \tag{A.88}$$

$$\begin{aligned}
&= \left[\frac{\Delta t}{(\lambda_{l_1}+\lambda_{j_1})}\left(\frac{1-\gamma^T e^{(\lambda_{l_1}+\lambda_{j_1})T}}{1-\gamma^{\Delta t}e^{(\lambda_{l_1}+\lambda_{j_1})\Delta t}}-\frac{1-\gamma^T}{1-\gamma^{\Delta t}}\right)-\frac{1}{(\lambda_{l_1}+\lambda_{j_1})}\left(\frac{1}{\log(\gamma)}-\frac{1}{\log(\gamma)+\lambda_{l_1}+\lambda_{j_1}}\right)\right]\times\\
&\times\left[\frac{\Delta t}{(\lambda_{l_2}+\lambda_{j_2})}\left(\frac{1-\gamma^T e^{(\lambda_{l_2}+\lambda_{j_2})T}}{1-\gamma^{\Delta t}e^{(\lambda_{l_2}+\lambda_{j_2})\Delta t}}-\frac{1-\gamma^T}{1-\gamma^{\Delta t}}\right)-\frac{1}{(\lambda_{l_2}+\lambda_{j_2})}\left(\frac{1}{\log(\gamma)}-\frac{1}{\log(\gamma)+\lambda_{l_2}+\lambda_{j_2}}\right)\right],
\end{aligned} \tag{A.89}$$

and

$$\mathcal{I}_3\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,\alpha,\beta\right) \tag{A.90}$$

$$= \mathbb{E}\left\{\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\gamma^{k\Delta t}\left(\int_0^{k\Delta t}e^{\lambda_{l_1}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t}e^{\lambda_{j_1}(k\Delta t-s)}\mathrm{d}W_\beta^{(i)}(s)\right)\right]\times \tag{A.91}$$

$$\times\left[\frac{\Delta t}{M}\sum_{i=1}^{M}\sum_{k=0}^{N-1}\gamma^{k\Delta t}\left(\int_0^{k\Delta t}e^{\lambda_{l_2}(k\Delta t-s)}\mathrm{d}W_\alpha^{(i)}(s)\right)\left(\int_0^{k\Delta t}e^{\lambda_{j_2}(k\Delta t-s)}\mathrm{d}W_\beta^{(i)}(s)\right)\right]\right\}. \tag{A.92}$$

Through similar arguments as in proof of Theorem 3.3.4, it is possible to conclude that $\mathcal{I}_1\left(M,h,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\alpha\right)$ has the same orders in Δt, $\mathbb{B}$ and T as the MSE result in Theorem 3.3.6.

Moreover, let $C_i(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T)$ denote some constants that depend on $\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T$.

Then:

$$\mathcal{I}_2\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,\alpha,\beta\right) \tag{A.93}$$
$$\begin{aligned}&=\overline{\sigma}^4\gamma^{2T}\left(C_1(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T)+C_2(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T)\Delta t\right)+\\&+\overline{\sigma}^4\gamma^{T}\left(C_3(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T)\Delta t^2+C_4(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T)\Delta t^3\right)+\\&+\overline{\sigma}^4\left(\frac{1}{144}+\gamma^T C_5(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T)\right)\Delta t^4+O(\Delta t^5),\end{aligned} \tag{A.94}$$

and

$$\mathcal{I}_3\left(M,\Delta t,T,\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,\alpha,\beta\right) \tag{A.95}$$
$$\begin{aligned}&=\frac{\Delta t^2}{M}\sum_{k=0}^{N-1}\frac{\left(e^{\left(\lambda_{l_1}+\lambda_{l_2}\right)k\Delta t}-1\right)\left(e^{\left(\lambda_{j_1}+\lambda_{j_2}\right)k\Delta t}-1\right)\Delta t^2\gamma^{2k\Delta t}}{\left(\lambda_{l_1}+\lambda_{l_2}\right)\left(\lambda_{j_1}+\lambda_{j_2}\right)}+\\&+\frac{\Delta t^2}{M}\sum_{k<q}\frac{e^{\lambda_{l_1}k\Delta t+\lambda_{l_2}q\Delta t+\lambda_{j_1}k\Delta t+\lambda_{j_2}q\Delta t}}{\left(\lambda_{l_1}+\lambda_{l_2}\right)\left(\lambda_{j_1}+\lambda_{j_2}\right)}\left(1-e^{-\left(\lambda_{l_1}+\lambda_{l_2}\right)k\Delta t}\right)\left(1-e^{-\left(\lambda_{j_1}+\lambda_{j_2}\right)k\Delta t}\right)\gamma^{(k+q)\Delta t}+\\&+\frac{\Delta t^2}{M}\sum_{k<q}\frac{e^{\lambda_{l_1}q\Delta t+\lambda_{l_2}k\Delta t+\lambda_{j_1}q\Delta t+\lambda_{j_2}k\Delta t}}{\left(\lambda_{l_1}+\lambda_{l_2}\right)\left(\lambda_{j_1}+\lambda_{j_2}\right)}\left(1-e^{-\left(\lambda_{l_1}+\lambda_{l_2}\right)k\Delta t}\right)\left(1-e^{-\left(\lambda_{j_1}+\lambda_{j_2}\right)k\Delta t}\right)\gamma^{(k+q)\Delta t}\end{aligned} \tag{A.96}$$

$$=C_6\left(\lambda_{l_1},\lambda_{j_1},\lambda_{l_2},\lambda_{j_2},\gamma,T\right)\frac{T^5}{\Delta t\mathbb{B}}+O\left(\frac{1}{\mathbb{B}}\right). \tag{A.97}$$

The Corollary is proved by combining the above results. □

A.4.3 The case when $\tilde{A}$ is a general stable matrix

Lemma A.4.1 (MSE when $\tilde{A}$ is a general stable matrix)**.** *Let $\tilde{A}$ be a stable $d\times d$ matrix with distinct eigenvalues $\lambda_1,\cdots,\lambda_m$ and corresponding multiplicities $q_1,\cdots,q_m$. There exist some constants $\{\bar{C}_i\}_{i=1}^m$, $\bar{C}_0$ and $C_j(\lambda_1,\cdots,\lambda_m,\gamma,T)$, such that the mean-squared error of the Monte-Carlo estimator in different setting satisfies:*

1. *Finite-Horizon undiscounted setting:*

$$MSE_T\in\left[\sum_{i=1}^m q_i\bar{C}_i MSE_T(\Delta t,\mathbb{B},\lambda_i),\right.$$
$$\left.C_1(\lambda_1,\cdots,\lambda_m,T)\overline{\sigma}^4T^2\Delta t^2+\frac{\left(\bar{C}_2+C_3(\lambda_1,\cdots,\lambda_m,T)O(T)\right)\overline{\sigma}^4T^{2n+3}}{\mathbb{B}\Delta t}+O(\Delta t^3)+O(\frac{1}{\mathbb{B}})\right], \tag{A.98}$$

where $MSE_T(\Delta t, \mathbb{B}, \lambda_i)$ is the mean-squared error of the Monte-Carlo estimator in Theorem 3.3.1 by replacing the drift u by λ_i.

2. *Finite-Horizon discounted setting:*

$$MSE_T \in \Bigg[\sum_{i=1}^{m} q_i \bar{C}_i MSE_T(\Delta t, \mathbb{B}, \gamma, \lambda_i),$$
$$C_4(\lambda_1, \cdots, \lambda_m, \gamma, T)\overline{\sigma}^4 \gamma^{2T} T^2 \Delta t^2 + C_5(\lambda_1, \cdots, \lambda_m, \gamma, T)\overline{\sigma}^4 \gamma^T \Delta t^3 + C_6(\lambda_1, \cdots, \lambda_m, T)\overline{\sigma}^4 \Delta t^4 +$$
$$+ \frac{(C_7(\lambda_1, \cdots, \lambda_m, \gamma, T))\,\overline{\sigma}^4 T^{2n-1}}{\mathbb{B}\Delta t} + O(\Delta t^5) + O(\frac{1}{\mathbb{B}}) \Bigg], \tag{A.99}$$

where $MSE_T(\Delta t, \mathbb{B}, \gamma, \lambda_i)$ is the mean-squared error of the Monte-Carlo estimator in Lemma A.3.1 by replacing the drift u by λ_i.

3. *Infinite-Horizon discounted setting:*

$$MSE_\infty \in \Bigg[\sum_{i=1}^{m} q_i \bar{C}_i MSE_\infty(\Delta t, \mathbb{B}, \gamma, \lambda_i),$$
$$(C_8(\lambda_1, \cdots, \lambda_m, \gamma, T) + C_9(\lambda_1, \cdots, \lambda_m, \gamma, T)\Delta t)\overline{\sigma}^4 \gamma^{2T} +$$
$$+ \Big(C_{10}(\lambda_1, \cdots, \lambda_m, \gamma, T)\Delta t^2 + C_{11}(\lambda_1, \cdots, \lambda_m, \gamma, T)\Delta t^3\Big)\,\overline{\sigma}^4 \gamma^T + C_{12}(\lambda_1, \cdots, \lambda_m, T)\overline{\sigma}^4 \Delta t^4 +$$
$$+ \frac{(C_{13}(\lambda_1, \cdots, \lambda_m, \gamma, T))\,\overline{\sigma}^4 T^{2n-1}}{\mathbb{B}\Delta t} + O(\Delta t^5) + O(\frac{1}{\mathbb{B}}) \Bigg], \tag{A.100}$$

where $MSE_\infty(\Delta t, \mathbb{B}, \gamma, \lambda_i)$ is the mean-squared error of the Monte-Carlo estimator in Theorem 3.3.6 by replacing the drift u by λ_i.

Proof. It is possible to understand that the proof of Lemma A.3.1 is based on the proof of Theorem 3.3.1 when adding a discount factor γ, and the proof of Theorem 3.3.6 is based on the proof of Lemma A.3.1 with the decomposition (3.31). By using the same flow direction, it is sufficient to show the result in case (1) and the results in case (2) and

(3) follows. Consider the decomposition of MSE_T in finite-horizon undiscounted setting:

$$\mathrm{MSE}_T = \mathbb{E}\left[(\hat{V}_M - V_T)^2\right] \tag{A.101}$$

$$= \mathbb{E}\left[\left(\hat{V}_M - \mathbb{E}\left[\hat{V}_M\right] + \mathbb{E}\left[\hat{V}_M\right] - V_T\right)^2\right] \tag{A.102}$$

$$= \underbrace{\mathbb{E}\left[\hat{V}_M^2\right] - \mathbb{E}\left[\hat{V}_M\right]^2}_{\text{Part1}} + \underbrace{\left(\mathbb{E}\left[\hat{V}_M\right] - V_T\right)^2}_{\text{Part2}} \tag{A.103}$$

Before the analysis of Part1 and Part2, the following mean-squared error notations is introduced for the finite-horizon undiscounted scalar case with drift λ_i:

$$\mathrm{MSE}_T(\Delta t, \mathbb{B}, \lambda_i) = \mathrm{Var}(\Delta t, \lambda_i) + \mathrm{Approximation}(\Delta t, \mathbb{B}, \lambda_i)\,, \tag{A.104}$$

where $\mathrm{Var}(\Delta t, \lambda_i) = \mathbb{E}\left[\hat{V}_M^2\right] - \mathbb{E}\left[\hat{V}_M\right]^2$ and $\mathrm{Approximation}(\Delta t, \mathbb{B}, \lambda_i) = \left(\mathbb{E}\left[\hat{V}_M\right] - V_T\right)^2$. For Part1:

$$\begin{aligned}
\mathbb{E}\left[\hat{V}_M^2\right] &= \frac{\Delta t^2}{M} \sum_{i,j,k,l} \mathbb{E}\left[x_i(k\Delta t)^\top \tilde{Q} x_i(k\Delta t) x_j(l\Delta t)^\top \tilde{Q} x_j(l\Delta t)\right] \\
&= \frac{\Delta t^2}{M^2} \sum_{i,j,k,l} \left[\mathbb{E}\left[x_i(k\Delta t)^\top \tilde{Q} x_i(k\Delta t)\right] \mathbb{E}\left[x_j(l\Delta t)^\top \tilde{Q} x_j(l\Delta t)\right] + 2\mathrm{Tr}\left(\tilde{Q}\mathbb{E}\left[x_i(k\Delta t) x_j(l\Delta t)^\top\right]\right)^2\right] \\
&= \Delta t^2 \sum_{k,l} \mathbb{E}\left[x(k\Delta t)^\top \tilde{Q} x(k\Delta t)\right] \mathbb{E}\left[x_j(l\Delta t)^\top \tilde{Q} x(l\Delta t)\right] + \\
&+ \frac{2\Delta t^2}{M} \sum_{k} \mathrm{Tr}\left(\tilde{Q}\mathbb{E}\left[x(k\Delta t) x(k\Delta t)^\top\right]\right)^2 + \frac{4\Delta t^2}{M} \sum_{k<l} \mathrm{Tr}\left(\tilde{Q}\mathbb{E}\left[x(k\Delta t) x(l\Delta t)^\top\right]\right)^2\,,
\end{aligned} \tag{A.105}$$

where the second equality is based on Isserlis' theorem and the trace operation. Notice that $\mathbb{E}\left[\hat{V}_M\right]^2 = \Delta t^2 \sum_{k,l} \mathbb{E}\left[x(k\Delta t)^\top \tilde{Q} x(k\Delta t)\right] \mathbb{E}\left[x(l\Delta t)^\top Q x(l\Delta t)\right]$, thus:

$$\mathbb{E}\left[\hat{V}_M^2\right] - \mathbb{E}\left[\hat{V}_M\right]^2 \tag{A.106}$$

$$= \frac{2\Delta t^2}{M} \sum_{k} \mathrm{Tr}\left(\tilde{Q}\mathbb{E}\left[x(k\Delta t) x(k\Delta t)^\top\right]\right)^2 + \frac{4\Delta t^2}{M} \sum_{k<l} \mathrm{Tr}\left(\tilde{Q}\mathbb{E}\left[x(k\Delta t) x(l\Delta t)^\top\right]\right)^2\,. \tag{A.107}$$

To analyze the above expression, decompose the matrix $\tilde{A}$ by its Jordan form, i.e. $\tilde{A} = P^{-1} J P$ for some invertible matrix P and $J = \mathrm{diag}(J_i, \cdots, J_m)$, where J_i is the Jordan block corresponding to the eigenvalue λ_i.

Notice that $e^{J(k\Delta t-s)} = \mathrm{diag}(e^{J_1(k\Delta t-s)}, \cdots, e^{J_m(k\Delta t-s)})$, where:

$$e^{J_i(k\Delta t-s)} = e^{\lambda_i(k\Delta t-s)} \begin{pmatrix} 1 & k\Delta t-s & \frac{(k\Delta t-s)^2}{2!} & \cdots & \frac{(k\Delta t-s)^{q_i-1}}{(q_i-1)!} \\ 0 & 1 & k\Delta t-s & \cdots & \frac{(k\Delta t-s)^{q_i-2}}{(q_i-2)!} \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \end{pmatrix}. \tag{A.108}$$

Combining with the fact that for any k, l, it holds:

$$\mathbb{E}\left[x(k\Delta t)x(l\Delta t)^\top\right] = \int_0^{k\Delta t\wedge l\Delta t} e^{\tilde{A}(k\Delta t-s)} e^{\tilde{A}^\top(l\Delta t-s)} \mathrm{d}s \tag{A.109}$$

$$= \int_0^{k\Delta t\wedge l\Delta t} P^{-1} e^{J(k\Delta t-s)} P P^\top e^{J^\top(l\Delta t-s)} P^{-\top} \mathrm{d}s\,, \tag{A.110}$$

and it is possible to conclude that for any $k \le l$, $\mathrm{Tr}\left(\tilde{Q}\mathbb{E}\left[x(k\Delta t)x(l\Delta t)^\top\right]\right)$ is a linear combination of $L_{1,i,j}$ and $L_{2,i,j}$ for all i, j, where:

$$L_{1,i,j} := C_{1,i,j} \int_0^{k\Delta t} e^{(\lambda_i(k\Delta t-s)+\lambda_j(l\Delta t-s))} \mathrm{d}s \tag{A.111}$$

$$= C_{1,i,j} \frac{e^{\lambda_i k\Delta t} + e^{\lambda_j}}{\lambda_i+\lambda_j} \left(1 - e^{-(\lambda_i+\lambda_j)k\Delta t}\right); \tag{A.112}$$

$$L_{2,i,j} := C_{2,i,j} \int_0^{k\Delta t} e^{(\lambda_i(k\Delta t-s)+\lambda_j(l\Delta t-s))} (k\Delta t-s)^{\tilde{q}_i} (l\Delta t-s)^{\tilde{q}_j} \mathrm{d}s\,, \tag{A.113}$$

in which $C_{1,i,j}$, $C_{i,j}$ are some constants and $\tilde{q}_i \in \{0, \cdots, q_i-1\}$, $\tilde{q}_j \in \{0, \cdots, q_j-1\}$. Concerning the integral in $L_{2,i,j}$, as $\tilde{q}_i + \tilde{q}_j \le n-1$, the following inequality holds:

$$\int_0^{k\Delta t} e^{(\lambda_i(k\Delta t-s)+\lambda_j(l\Delta t-s))} (k\Delta t-s)^{\tilde{q}_i} (l\Delta t-s)^{\tilde{q}_j} \mathrm{d}s$$

$$\le T^{n-1} \int_0^{k\Delta t} e^{(\lambda_i(k\Delta t-s)+\lambda_j(l\Delta t-s))} \mathrm{d}s\,. \tag{A.114}$$

This is true because of

$$\mathrm{Tr}\left(\tilde{Q}\,\mathbb{E}\left[x(k\Delta t)x(l\Delta t)^\top\right]\right)^2 = \sum_{i_1,j_1,i_2,j_2} \sum_{k,l} \prod_{l_1,l_2\in\{1,2\}} L_{l_1,i_1,j_1} L_{l_2,i_2,j_2}\,, \tag{A.115}$$

and all the terms are nonnegative.

Dropping all terms that include $L_{2,i,j}$ factor and only include the $L_{1,i,i}^2$ with $k = l$ terms, one can find the lower bound of Part1. That is to say, the lower bound of part 1 is $\sum_{i=1}^m q_i \bar{C}_i \mathrm{Var}(\Delta t, \lambda_i)$.

The upper bound of Part1 can be obtained by replacing all $L_{1,i,j}$ factors by $L_{2,i,j}$ and using the bound given in (A.114). This leads to the following expression for the upper bound of Part1:

$$\frac{\left(\bar{C}_2 + C_3(\lambda_1, \cdots, \lambda_m, T)O(T)\right)\overline{\sigma}^4 T^{2n+5}}{\mathbb{B}\Delta t} + O(\frac{1}{\mathbb{B}})\,. \tag{A.116}$$

For Part2, let $g(t) = \mathbb{E}\left[x(t)^\top \tilde{Q} x(t)\right]$ on $[0, T]$. Then $\mathbb{E}\left[\hat{V}_M\right]$ is the left Riemann sum approximation of $g(t)$, and by the property of Riemann approximation:

$$\left|\mathbb{E}\left[\hat{V}_M\right] - V_T\right| \approx 2\Delta t T g(T) + O(\Delta t^2)\,, \tag{A.117}$$

where

$$g(T) = \mathrm{Tr}\left(\tilde{Q}\,\mathbb{E}\left[x(T)x(T)^\top\right]\right) = \overline{\sigma}^2 \mathrm{Tr}\left(\tilde{Q}\int_0^T e^{\tilde{A}(t-s)} e^{\tilde{A}^\top (t-s)} \mathrm{d}s\right)\,, \tag{A.118}$$

which is a constant depends on $\lambda_1, \cdots, \lambda_m, T$. Thus:

$$(\mathbb{E}\left[\hat{V}_M\right] - V_T)^2 \approx C_1(\lambda_1, \cdots, \lambda_m, T)\overline{\sigma}^4 T^2 \Delta t^2 + O(\Delta t^3)\,, \tag{A.119}$$

which has the same order in Δt as the scalar case in finite-horizon undiscounted setting. Hence the result in (A.98) is obtained by combining the bound in Part1 and the approximation in Part2.

As explained in the beginning of this proof, in the finite-horizon discounted setting similar arguments as in the proof of (A.98) will be followed to obtain result (A.99).

For the infinite-horizon discounted setting, the corresponding part 1 in the MSE_∞ is the same as the part 1 in MSE_T of (A.99). Part2 is approximated by using the decomposition (3.31) and the fact that

$$V_{t,\infty} = \int_T^\infty \gamma^t \mathbb{E}\left[x(t)^\top \tilde{Q} x(t)\right] dt = \gamma^T C(\gamma, T, \lambda_1, \cdots, \lambda_m)\,. \tag{A.120}$$

To verify $V_{T,\infty}$ is $O(\gamma^T)$, one can find the bounds of $V_{T,\infty}$ by using similar arguments in the above proof of (A.98) and the following inequality:

$$\int_0^t e^{(\lambda_i+\lambda_j)(t-s)}(t-s)^{\tilde{q}_i+\tilde{q}_j}\mathrm{d}s \le t^{n-1}\int_0^t e^{(\lambda_i+\lambda_j)(t-s)}\mathrm{d}s = \frac{t^{n-1}}{(\lambda_i+\lambda_j)}\left(e^{(\lambda_i+\lambda_j)t} - 1\right). \tag{A.121}$$

Then the components in $\int_T^\infty \gamma^t \mathbb{E}\left[x(t)x(t)^\top\right] \mathrm{d}t$ is lower bounded by $\int_T^\infty \frac{\gamma^t}{(\lambda_i+\lambda_j)}\left(e^{(\lambda_i+\lambda_j)t} - 1\right)\mathrm{d}t$ and upper bounded by $\int_T^\infty \frac{\gamma^t t^{n-1}}{(\lambda_i+\lambda_j)}\left(e^{(\lambda_i+\lambda_j)t} - 1\right)\mathrm{d}t$. Through the approximation of in-

complete gamma function when T is large, it holds:

$$\int_T^{\infty} \frac{\gamma^t t^{n-1}}{(\lambda_i+\lambda_j)} \left(e^{(\lambda_i+\lambda_j)t}-1\right) \mathrm{d}t \approx \frac{\gamma^T T^{n-1}}{(\lambda_i+\lambda_j)} \left(e^{(\lambda_i+\lambda_j)T}-1\right) . \tag{A.122}$$

By noting that:

$$V_{T,\infty} = \mathrm{Tr}\left(\tilde{Q}\int_T^{\infty} \gamma^t \mathbb{E}\left[x(t)x(t)^{\top}\right] \mathrm{d}t\right) , \tag{A.123}$$

it is possible to show that $V_{T,\infty} = \gamma^T C(\gamma, T, \lambda_1, \cdots, \lambda_m)$.
This result leads to the fact that Part2 is given by:

$$\begin{aligned}&(C_8(\lambda_1,\cdots,\lambda_m,\gamma,T) + C_9(\lambda_1,\cdots,\lambda_m,\gamma,T)\Delta t)\,\overline{\sigma}^4\gamma^{2T} + \Big(C_{10}(\lambda_1,\cdots,\lambda_m,\gamma,T)\Delta t^2 \; + \\ &+ C_{11}(\lambda_1,\cdots,\lambda_m,\gamma,T)\Delta t^3\Big)\,\overline{\sigma}^4\gamma^T + C_{12}(\lambda_1,\cdots,\lambda_m,T)\overline{\sigma}^4\Delta t^4 + O(\Delta t^5)\,,\end{aligned} \tag{A.124}$$

which coincides with $\mathrm{Var}(\Delta t\lambda_i)$ in the infinite-horizon discounted scalar case. The results in (3) then follows. □

References

Abbasi-Yadkori, Yasin and Csaba Szepesvári (June 2011). "Regret Bounds for the Adaptive Control of Linear Quadratic Systems". In: *Proceedings of the 24th Annual Conference on Learning Theory*. Ed. by Sham M. Kakade and Ulrike von Luxburg. Vol. 19. Proceedings of Machine Learning Research. Budapest, Hungary: PMLR, pp. 1–26 (Cited in page 37).

Agarwal, R.P. and V. Lakshmikantham (1993). *Uniqueness And Nonuniqueness Criteria For Ordinary Differential Equations*. Series In Real Analysis. World Scientific Publishing Company (Cited in page 22).

Aït-Sahalia, Yacine (2002). "Maximum likelihood estimation of discretely sampled diffusions: A closed-form approximation approach". English (US). In: *Econometrica* 70.1, pp. 223–262 (Cited in page 51).

Alon, U. (2006). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC Mathematical and Computational Biology. Taylor & Francis (Cited in page 100).

Anderson, Brian D. O. and John B. Moore (1990). *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Inc. (Cited in page 112).

Arbabi, Hassan and Igor Mezić (2017a). "Ergodic Theory, Dynamic Mode Decomposition, and Computation of Spectral Properties of the Koopman Operator". In: *SIAM Journal on Applied Dynamical Systems* 16.4, pp. 2096–2126 (Cited in pages 58, 59).

— (Dec. 2017b). "Study of dynamics in post-transient flows using Koopman mode decomposition". In: *Phys. Rev. Fluids* 2 (12), p. 124402 (Cited in page 7).

Aronszajn, N. (1950). "Theory of Reproducing Kernels". In: *Transactions of the American Mathematical Society* 68.3, pp. 337–404 (Cited in page 74).

Arumugam, Dilip, David Abel, Kavosh Asadi, Nakul Gopalan, Christopher Grimm, Jun Ki Lee, Lucas Lehnert, and Michael L. Littman (2018). "Mitigating Planner Overfitting in Model-Based Reinforcement Learning". In: *CoRR* abs/1812.01129 (Cited in page 149).

Axler, S.J., H.H. Schaefer, M.P. Wolff, M.P.H. Wolff, F.W. Gehring, and K.A. Ribet (1999). *Topological Vector Spaces*. Graduate Texts in Mathematics. Springer New York (Cited in page 76).

Bagheri, Shervin (2013). "Koopman-mode decomposition of the cylinder wake". In: *Journal of Fluid Mechanics* 726. QC 20131125, pp. 596–623 (Cited in page 83).

Bahl, Srikhar, Mustafa Mukadam, Abhinav Gupta, and Deepak Pathak (2020). "Neural dynamic policies for end-to-end sensory motor learning". In: *Advances in Neural Information Processing Systems* 34 (Cited in page 36).

Baird, Leemon C (1994). "Reinforcement learning in continuous time: Advantage updating". In: *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 4. IEEE, pp. 2448–2453 (Cited in page 36).

Baker, G.L. and J.A. Blackburn (2008). *The Pendulum: A Case Study in Physics*. OUP Oxford (Cited in page 16).

Banach, Stefan (1922). "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales". fre. In: *Fundamenta Mathematicae* 3.1, pp. 133–181 (Cited in page 124).

Barto, Andrew, Satinder Singh, and Nuttapong Chentanez (Jan. 2004). "Intrinsically motivated learning of hierarchical collections of skills". In: *Proceedings of the 3rd International Conference on Developmental Learning* (Cited in page 126).

Barto, Andrew G., Richard S. Sutton, and Charles W. Anderson (1983). "Neuronlike adaptive elements that can solve difficult learning control problems". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.5, pp. 834–846 (Cited in page 148).

Baumeister, Johann (1987). *Stable Solution of Inverse Problems*. Advanced Lectures in Mathematics. Vieweg+Teubner Verlag Wiesbaden, pp. VIII, 256 (Cited in page 65).

Bayes, Thomas (1763). "An essay towards solving a problem in the doctrine of chances". In: *Philosophical Transactions of the Royal Society of London* 53, pp. 370–418 (Cited in page 92).

Bellman, Richard (1957). *Dynamic Programming*. Dover Publications (Cited in pages 108, 124).

— (1954). "The theory of dynamic programming". In: *Bulletin of the American Mathematical Society* 60.6, pp. 503 –515 (Cited in page 108).

Berger, Erik, Mark Sastuba, David Vogt, Bernhard Jung, and Heni Ben Amor (Mar. 2015). "Estimation of perturbations in robotic behavior using dynamic mode decomposition". English (US). In: *Advanced Robotics* 29.5. Publisher Copyright: 2015 Taylor & Francis and The Robotics Society of Japan., pp. 331–343 (Cited in page 59).

Berlinet, Alain and Christine Thomas-Agnan (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer New York, pp. XXII, 355 (Cited in page 66).

Bertalanffy, Ludwig (1962). "General Systems". In: *General system year book* 7-8 (Cited in page 11).

Bertsekas, Dimitri P. (1987). *Dynamic Programming: Deterministic and Stochastic Models*. USA: Prentice-Hall, Inc. (Cited in page 124).

Bertsekas, Dimitri P. and John N. Tsitsiklis (1989). *Parallel and Distributed Computation: Numerical Methods*. USA: Prentice-Hall, Inc. (Cited in page 124).

Bhandari, Jalaj and Daniel Russo (Apr. 2021). "On the Linear Convergence of Policy Gradient Methods for Finite MDPs". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 2386–2394 (Cited in pages 120, 128).

Bhatia, Abhishek, Jaan Altosaar, and Shixiang Gu (2017). "Proximity-constrained reinforcement learning". In: *Advances in Neural Information Processing Systems* (Cited in page 140).

Bijl, Hildo, Jan-Willem van Wingerden, Thomas B Schön, and Michel Verhaegen (2016). "Mean and variance of the LQG cost function". In: *Automatica* 67, pp. 216–223 (Cited in pages 37, 171).

Bintu, Lacramioara, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jané Kondev, and Rob Phillips (2005). "Transcriptional regulation by the numbers: models". In: *Current Opinion in Genetics & Development* 15.2. Chromosomes and expression mechanisms, pp. 116–124 (Cited in page 100).

Bousquet, Olivier and André Elisseeff (2002). "Stability and Generalization". In: *Journal of Machine Learning Research* 2.Mar, pp. 499–526 (Cited in page 73).

Bradtke, Steven (1992). "Reinforcement Learning Applied to Linear Quadratic Regulation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Hanson, J. Cowan, and C. Giles. Vol. 5. Morgan-Kaufmann (Cited in page 37).

Bradtke, Steven J. and Michael O. Duff (1994). "Reinforcement learning methods for continuous-time Markov decision problems". In: *Advances in Neural Information Processing Systems* (Cited in page 36).

Breiman, Leo (1996). "Heuristics of instability and stabilization in model selection". In: *The Annals of Statistics* 24.6, pp. 2350 –2383 (Cited in page 73).

Breschi, V., A. Sassella, and S. Formentin (2022). "On the Design of Regularized Explicit Predictive Controllers From Input-Output Data". In: *IEEE Transactions on Automatic Control*, pp. 1–7 (Cited in page 6).

Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba (2016). "Openai gym". In: *arXiv preprint arXiv:1606.01540* (Cited in pages 48, 49).

Brunton, Steven L., Bingni W. Brunton, Joshua L. Proctor, and J. Nathan Kutz (Feb. 2016). "Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control". In: *PLOS ONE* 11.2, pp. 1–19 (Cited in page 116).

Brunton, Steven L. and J. Nathan Kutz (2019). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. 1st. USA: Cambridge University Press (Cited in page 17).

Brunton, Steven L., Joshua L. Proctor, and J. Nathan Kutz (2016). "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the National Academy of Sciences* 113.15, pp. 3932–3937 (Cited in page 116).

Budišić, Marko, Ryan Mohr, and Igor Mezić (2012). "Applied Koopmanism". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.4, p. 047510 (Cited in pages 58, 83).

Burda, Yuri, Harrison Edwards, Amos Storkey, and Oleg Klimov (Oct. 2018). "Exploration by Random Network Distillation". In: *arXiv e-prints*, arXiv:1810.12894, arXiv:1810.12894 (Cited in page 140).

Burnham, K.P. and D.R. Anderson (2002). *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Verlag (Cited in page 149).

Bátkai, András, Marjeta Fijavz, and Abdelaziz Rhandi (Jan. 2017). *Positive Operator Semigroups*. Vol. 257. Birkhäuser (Cited in page 105).

Caccioppoli, Renato (1932). "Sugli elementi uniti delle trasformazioni funzionali: un teorema di esistenza e di unicità ed alcune sue applicazioni". it. In: *Rendiconti del Seminario Matematico della Università di Padova* 3, pp. 1–15 (Cited in page 124).

Cao, Tongyi and Akshay Krishnamurthy (2020). "Provably adaptive reinforcement learning in metric spaces". In: *Advances in Neural Information Processing Systems* 34 (Cited in page 37).

Carli, Francesca Paola, Tianshi Chen, and Lennart Ljung (2017). "Maximum Entropy Kernels for System Identification". In: *IEEE Transactions on Automatic Control* 62.3, pp. 1471–1477 (Cited in page 84).

Cerone, V., D. Regruto, and M. Abuabiah (2017). "Direct data-driven control design through set-membership errors-in-variables identification techniques". In: *2017 American Control Conference (ACC)*, pp. 388–393 (Cited in page 6).

Cesa-Bianchi, Nicolò, Yishay Mansour, and Ohad Shamir (July 2015). "On the Complexity of Learning with Kernels". In: *Proceedings of The 28th Conference on Learning Theory*. Ed. by Peter Grünwald, Elad Hazan, and Satyen Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, pp. 297–325 (Cited in page 102).

Chaloner, K. and I. Verdinelli (1995). "Bayesian experimental design: A review". In: *Statistical Science* 10.3, pp. 273–304 (Cited in page 37).

Chen, Kevin K., Jonathan H. Tu, and Clarence W. Rowley (2012). "Variants of Dynamic Mode Decomposition: Boundary Condition, Koopman, and Fourier Analyses". English. In: *Journal of Nonlinear Science* 22.6. Communicated by P. Newton., pp. 887–915 (Cited in page 83).

Chen, Tianshi (2018). "On kernel design for regularized LTI system identification". In: *Automatica* 90, pp. 109–122 (Cited in page 84).

Chen, Tianshi, Martin Skovgaard Andersen, Lennart Ljung, Alessandro Chiuso, and Gianluigi Pillonetto (2014). "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques". English. In: *IEEE Transactions on Automatic Control* 59.11, pp. 2933–2945 (Cited in page 84).

Chen, Tianshi, Tohid Ardeshiri, Francesca P. Carli, Alessandro Chiuso, Lennart Ljung, and Gianluigi Pillonetto (Apr. 2016). "Maximum Entropy Properties of Discrete-Time First-Order Stable Spline Kernel". In: *Automatica* 66.C, 34–38 (Cited in page 84).

Chen, Tianshi and Lennart Ljung (2015). "On kernel structures for regularized system identification (I): a machine learning perspective". In: *IFAC-PapersOnLine* 48.28. 17th IFAC Symposium on System Identification SYSID 2015, pp. 1035–1040 (Cited in page 84).

Chiuso, A., T. Chen, L. Ljung, and G. Pillonetto (2014). "On the design of multiple kernels for nonparametric linear system identification". In: *53rd IEEE Conference on Decision and Control*, pp. 3346–3351 (Cited in page 84).

Coddington, A. and N. Levinson (1955). *Theory of Ordinary Differential Equations*. International series in pure and applied mathematics. McGraw-Hill Companies (Cited in page 133).

Das, Suddhasattwa and Dimitrios Giannakis (June 2019). "Delay-Coordinate Maps and the Spectra of Koopman Operators". In: *Journal of Statistical Physics* 175.6, pp. 1107–1145 (Cited in page 86).

Das, Suddhasattwa and Dimitrios Giannakis (2020). "Koopman spectra in reproducing kernel Hilbert spaces". In: *Applied and Computational Harmonic Analysis* 49.2, pp. 573–607 (Cited in page 86).

Dean, Sarah, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu (Aug. 2020). "On the Sample Complexity of the Linear Quadratic Regulator". In: *Foundations of Computational Mathematics* 20.4, pp. 633–679 (Cited in pages 37, 148, 149).

— (2018). "Regret bounds for robust adaptive control of the linear quadratic regulator". In: *Advances in Neural Information Processing Systems* 31 (Cited in page 37).

Degris, Thomas, Martha White, and Richard S. Sutton (2012). "Off-Policy Actor-Critic". In: *Proceedings of the 29th International Coference on International Conference on*

Machine Learning. ICML'12. Edinburgh, Scotland: Omnipress, 179–186 (Cited in page 148).

Deisenroth, Marc Peter and Carl Edward Rasmussen (2011). "PILCO: A Model-Based and Data-Efficient Approach to Policy Search". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Bellevue, Washington, USA: Omnipress, 465–472 (Cited in page 149).

Dellnitz, Michael and Oliver Junge (2002). "Chapter 5 - Set Oriented Numerical Methods for Dynamical Systems". In: *Handbook of Dynamical Systems*. Ed. by Bernold Fiedler. Vol. 2. Handbook of Dynamical Systems. Elsevier Science, pp. 221–264 (Cited in page 19).

Dellnitz, Michael, Stefan Klus, and Adrian Ziessler (2017). "A Set-Oriented Numerical Approach for Dynamical Systems with Parameter Uncertainty". In: *SIAM Journal on Applied Dynamical Systems* 16.1, pp. 120–138 (Cited in page 19).

Devidze, Rati, Goran Radanovic, Parameswaran Kamalaruban, and Adish Singla (2021). "Explicable Reward Design for Reinforcement Learning Agents". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., pp. 20118–20131 (Cited in page 126).

Dill, K.A. and S. Bromberg (2011). *Molecular Driving Forces: Statistical Thermodynamics in Biology, Chemistry, Physics, and Nanoscience*. Garland Science (Cited in page 100).

Dinuzzo, Francesco (2015). "Kernels for Linear Time Invariant System Identification". In: *SIAM Journal on Control and Optimization* 53.5, pp. 3299–3317 (Cited in page 84).

Doya, Kenji (2000). "Reinforcement learning in continuous time and space". In: *Neural computation* 12.1, pp. 219–245 (Cited in page 36).

Droge, Greg and Magnus Egerstedt (2011). "Adaptive time horizon optimization in model predictive control". In: *Proceedings of the American Control Conference*, pp. 1843–1848 (Cited in page 36).

Du, Ding-Zhu, Panos M. Pardalos, and Weili Wu (2009). "History of optimizationHistory of Optimization". In: *Encyclopedia of Optimization*. Ed. by Christodoulos A. Floudas and Panos M. Pardalos. Boston, MA: Springer US, pp. 1538–1542 (Cited in page 66).

Dunford, N., J.T. Schwartz, W.G. Bade, and R.G. Bartle (1958). *Linear Operators: General theory*. Linear Operators. Interscience Publishers (Cited in page 67).

Einsiedler, M. and T. Ward (2017). *Functional Analysis, Spectral Theory, and Applications*. Graduate Texts in Mathematics. Springer International Publishing (Cited in page 76).

Elisseeff, Andre, Theodoros Evgeniou, and Massimiliano Pontil (Dec. 2005). "Stability of Randomized Learning Algorithms". In: *Journal of Machine Learning Research* 6, 55–79 (Cited in page 73).

Eschmann, Jonas (2021). "Reward Function Design in Reinforcement Learning". In: *Reinforcement Learning Algorithms: Analysis and Applications*. Ed. by Boris Belousov, Hany Abdulsamad, Pascal Klink, Simone Parisi, and Jan Peters. Cham: Springer International Publishing, pp. 25–33 (Cited in page 126).

Everitt, Brian (2002). *The Cambridge dictionary of statistics*. Cambridge, UK; New York: Cambridge University Press (Cited in page 149).

Fazel, Maryam, Rong Ge, Sham Kakade, and Mehran Mesbahi (July 2018). "Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1467–1476 (Cited in pages 37, 120, 128).

Feinberg, Vladimir, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, and Sergey Levine (2018). "Model-Based Value Estimation for Efficient Model-Free Reinforcement Learning". In: *CoRR* abs/1803.00101 (Cited in page 6).

Formentin, S. (2012). *Direct Data-driven Control System Design: Theory and Applications*. Advances in mechatronics. Trauner (Cited in page 6).

Fraenkel, A.A., Y. Bar-Hillel, and A. Levy (1973). *Foundations of Set Theory*. ISSN. Elsevier Science (Cited in page 25).

Fréchet, M. (1907). "Sur les ensembles de fonctions et les opérations linéaires." French. In: *C. R. Acad. Sci., Paris* 144, pp. 1414–1416 (Cited in page 28).

Fukumizu, Kenji, Le Song, and Arthur Gretton (Dec. 2013). "Kernel Bayes' Rule: Bayesian Inference with Positive Definite Kernels". In: *Journal of Machine Learning Research* 14.1, 3753–3783 (Cited in pages 78, 86).

Gard, T.C. and T.C. Gard (1988). *Introduction to Stochastic Differential Equations*. Monographs and textbooks in pure and applied mathematics. M. Dekker (Cited in page 31).

Gardner, Jacob R., Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew Gordon Wilson (2018). "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration". In: *CoRR* abs/1809.11165 (Cited in page 159).

Gardner, Timothy S., Charles R. Cantor, and James J. Collins (Jan. 2000). "Construction of a genetic toggle switch in Escherichia coli". In: *Nature* 403.6767, pp. 339–342 (Cited in page 100).

Gaskett, Chris, David Wettergreen, and Alexander Zelinsky (1999). "Q-Learning in Continuous State and Action Spaces". In: *Advanced Topics in Artificial Intelligence*. Ed. by Norman Foo. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 417–428 (Cited in page 127).

Gaspard, Pierre (1998). *Chaos, Scattering and Statistical Mechanics*. Cambridge Nonlinear Science Series. Cambridge University Press (Cited in page 18).

Georgiou, Tryphon T. and Anders Lindquist (2013). "The Separation Principle in Stochastic Control, Redux". In: *IEEE Transactions on Automatic Control* 58.10, pp. 2481–2494 (Cited in page 37).

Giannakis, Dimitrios (2017). "Data-driven spectral decomposition and forecasting of ergodic dynamical systems". In: *Applied and Computational Harmonic Analysis* (Cited in page 85).

Goldstine, H.H. (2012). *A History of the Calculus of Variations from the 17th through the 19th Century*. Studies in the History of Mathematics and Physical Sciences. Springer New York (Cited in page 66).

Gowers, Timothy, June Barrow-Green, and Imre Leader, eds. (2010). *The Princeton Companion to Mathematics*. Princeton: Princeton University Press (Cited in page 26).

Greensmith, Evan, Peter Bartlett, and Jonathan Baxter (2001). "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press (Cited in page 127).

Griewank, Andreas and Andrea Walther (2008). "2. A Framework for Evaluating Functions". In: *Evaluating Derivatives*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, pp. xxi + 426 (Cited in page 141).

Gu, Shixiang, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine (2016). "Continuous Deep Q-Learning with Model-Based Acceleration". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2829–2838 (Cited in pages 6, 37).

Guého, Damien, Puneet Singla, and Manoranjan Majji (2021). "Time-Varying Koopman Operator Theory for Nonlinear Systems Prediction". In: *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2294–2299 (Cited in page 22).

Haaser, N.B. and J.A. Sullivan (1971). *Real Analysis*. The University series in mathematics. Van Nostrand Reinhold Company (Cited in page 28).

Halmos, Paul R. (1950). *Measure Theory*. D. Van Nostrand Company, Inc., New York, pp. xi+304 (Cited in page 28).

Hasselt, Hado van (2012). "Reinforcement Learning in Continuous State and Action Spaces". In: *Reinforcement Learning*. Ed. by Marco A. Wiering and Martijn van Otterlo. Vol. 12. Adaptation, Learning, and Optimization. Springer, pp. 207–251 (Cited in page 127).

Hasselt, Hado van and Marco A. Wiering (2007). "Reinforcement Learning in Continuous Action Spaces". In: *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pp. 272–279 (Cited in page 127).

Hasselt, Hado Philip van (2011). "Insights in reinforcement rearning : formal analysis and empirical evaluation of temporal-difference learning algorithms". PhD thesis. Utrecht University, Netherlands (Cited in page 148).

Hemati, Maziar S., Matthew O. Williams, and Clarence W. Rowley (2014). "Dynamic mode decomposition for large and streaming datasets". In: *Physics of Fluids* 26.11, p. 111701 (Cited in page 83).

Hewitt, E. and K. Stromberg (1965). *Real and Abstract Analysis*. Springer (Cited in page 28).

Hirsch, Morris W. and Stephen Smale (1974). *Differential equations, dynamical systems, and linear algebra*. Pure and applied mathematics 60. San Diego: Academic Press. XI, 358 (Cited in page 106).

Houthooft, Rein, Xi Chen, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel (2016). "VIME: Variational Information Maximizing Exploration". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc. (Cited in page 140).

Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press (Cited in page 124).

Iacob, Lucian Cristian, Gerben Izaak Beintema, Maarten Schoukens, and Roland Tóth (2021). "Deep Identification of Nonlinear Systems in Koopman Form". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. Austin, TX, USA: IEEE Press, 2288–2293 (Cited in page 18).

Iacob, Lucian Cristian, Roland Tóth, and Maarten Schoukens (2022). *Koopman Form of Nonlinear Systems with Inputs* (Cited in pages 114, 120).

Isidori, Alberto (1995). *Nonlinear Control Systems, Third Edition*. Communications and Control Engineering. Springer (Cited in page 113).

Jaakkola, Tommi, Michael I. Jordan, and Satinder P. Singh (1993). "Convergence of Stochastic Iterative Dynamic Programming Algorithms". In: *Proceedings of the 6th*

International Conference on Neural Information Processing Systems. NIPS'93. Denver, Colorado: Morgan Kaufmann Publishers Inc., 703–710 (Cited in page 148).

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer (Cited in page 52).

Jech, T.J. (2008). *The Axiom of Choice*. Dover Books on Mathematics Series. Dover Publications (Cited in page 25).

Jefferys, William H. and James O. Berger (Aug. 1991). "Sharpening Ockham's razor on a Bayesian strop". In: (Cited in page 73).

Jiang, Nan, Alex Kulesza, Satinder Singh, and Richard Lewis (2016). "The dependence of effective planning horizon on model accuracy". In: *Proceedings of the International Joint Conference on Artificial Intelligence* (Cited in page 36).

John, George H. (1994). "When the Best Move Isn't Optimal: Q-Learning with Exploration". In: *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*. AAAI'94. Seattle, Washington: AAAI Press, p. 1464 (Cited in page 148).

Jonathan, Baxter and Bartlett Peter (1999). "Direct gradient-based reinforcement learning". In: *Journal of Artificial Intelligence Research* (Cited in page 127).

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). "Reinforcement learning: A survey". In: *Journal of Artificial Intelligence Research* 4, pp. 237–285 (Cited in pages 122, 124).

Kakade, Sham M (2001). "A Natural Policy Gradient". In: *Advances in Neural Information Processing Systems*. Ed. by T. Dietterich, S. Becker, and Z. Ghahramani. Vol. 14. MIT Press (Cited in page 127).

Kakade, Sham M. and Jason D. Lee (2018). "Provably Correct Automatic Subdifferentiation for Qualified Programs". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 7125–7135 (Cited in page 141).

Kallenberg, Olav (2002). *Foundations of modern probability*. Second. Probability and its Applications (New York). Springer-Verlag, New York, pp. xx+638 (Cited in page 31).

Kamien, Morton I and Nancy L Schwartz (1971). "Sufficient conditions in optimal control theory". In: *Journal of Economic Theory* 3.2, pp. 207–214 (Cited in page 107).

Kawahara, Yoshinobu (2016). "Dynamic Mode Decomposition with Reproducing Kernels for Koopman Spectral Analysis". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc. (Cited in page 85).

Khalil, H.K. (2002). *Nonlinear Systems*. Pearson Education. Prentice Hall (Cited in page 113).

Kim, Jeongho, Jaeuk Shin, and Insoon Yang (2021). "Hamilton-Jacobi Deep Q-learning for deterministic continuous-time systems with Lipschitz continuous controls". In: *Journal of Machine Learning Research* 22 (Cited in page 36).

Kirk, D.E. (2004). *Optimal Control Theory: An Introduction*. Dover Books on Electrical Engineering Series. Dover Publications (Cited in pages 107, 108).

Klus, Stefan, Feliks Nüske, Sebastian Peitz, Jan-Hendrik Niemann, Cecilia Clementi, and Christof Schütte (2020). "Data-driven approximation of the Koopman generator: Model reduction, system identification, and control". In: *Physica D: Nonlinear Phenomena* 406.132416 (Cited in page 32).

Klus, Stefan, Ingmar Schuster, and Krikamol Muandet (2020). "Eigendecompositions of Transfer Operators in Reproducing Kernel Hilbert Spaces". In: *Journal of Nonlinear Science* 30.1, pp. 283–315 (Cited in page 86).

Kober, Jens, J. Andrew Bagnell, and Jan Peters (2013). "Reinforcement Learning in Robotics: A Survey". In: *International Journal of Robotics Research* 32.11, pp. 1238–1274 (Cited in page 4).

Koopman, B. O. (1931). "Hamiltonian Systems and Transformation in Hilbert Space". In: *Proceedings of the National Academy of Sciences* 17.5, pp. 315–318 (Cited in pages 7, 18).

Korda, Milan and Igor Mezić (2018). "On Convergence of Extended Dynamic Mode Decomposition to the Koopman Operator". In: *Journal of Nonlinear Science* 28, pp. 687–710 (Cited in page 58).

Korda, Milan and Igor Mezić (2018). "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control". In: *Automatica* 93, pp. 149–160 (Cited in pages 111, 132).

Krauth, Karl, Stephen Tu, and Benjamin Recht (2019). "Finite-time Analysis of Approximate Policy Iteration for the Linear Quadratic Regulator". In: *Advances in Neural Information Processing Systems 32*, pp. 8514–8524 (Cited in page 37).

Kress, Rainer (1989). "Tikhonov Regularization". In: *Linear Integral Equations*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 243–258 (Cited in page 71).

Kurebayashi, Wataru, Sho Shirasaka, and Hiroya Nakao (Nov. 2016). "Optimal Parameter Selection for Kernel Dynamic Mode Decomposition". In: *Proceedings of The International Symposium on Nonlinear Theory and Its Applications*. Vol. 370. NOLTA, pp. 370–373 (Cited in page 85).

Kusse Bruce, R. and A. Westwig Erik (2006). "Solutions to Laplace's Equation". In: *Mathematical Physics*. John Wiley & Sons, Ltd. Chap. 11, pp. 424–490 (Cited in page 26).

Kwakernaak, Huibert and Raphael Sivan (1972). *Linear Optimal Control Systems*. John Wiley & amp; Sons, Inc. (Cited in page 112).

Kühner, Viktoria (2019). "What can Koopmanism do for attractors in dynamical systems?" In: *Journal of Analysis* 29.2, 449–471 (Cited in page 105).

Lamprecht, Ingolf and A. I. Zotin, eds. (2019). *Thermodynamics and Regulation of Biological Processes*. Berlin, Boston: De Gruyter (Cited in page 100).

Lan, Yueheng and Igor Mezić (2013). "Linearization in the large of nonlinear systems and Koopman operator spectrum". In: *Physica D: Nonlinear Phenomena* 242, pp. 42–53 (Cited in page 105).

Lancaster, Peter and L Rodman (1995). *The Algebraic Riccati Equation*. Oxford University Press (Cited in page 113).

Lasota, A. and M.C. Mackey (2013). *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. Applied Mathematical Sciences. Springer New York (Cited in page 105).

Lazaric, Alessandro, Marcello Restelli, and Andrea Bonarini (2007). "Reinforcement Learning in Continuous Action Spaces through Sequential Monte Carlo Methods". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc. (Cited in page 127).

Le Ngo, Anh Cat, John See, and Raphael C.-W. Phan (2017). "Sparsity in Dynamics of Spontaneous Subtle Emotions: Analysis and Application". In: *IEEE Transactions on Affective Computing* 8.3, pp. 396–411 (Cited in page 59).

Lee, E.B., L. Markus, Karreman Mathematics Research Collection, Society for Industrial, and Applied Mathematics (1967). *Foundations of Optimal Control Theory*. SIAM series in applied mathematics. Wiley (Cited in page 107).

Lee, Jaeyoung and Richard S. Sutton (2021). "Policy iterations for reinforcement learning problems in continuous time and space — Fundamental theory and methods". In: *Automatica* 126 (Cited in page 36).

Lefschetz, Solomon (1963). *Differential equations: Geometric theory*. Second edition. Pure and Applied Mathematics, Vol. VI. Interscience Publishers, a division of John Wiley & Sons, New York - London, pp. x+390 (Cited in page 106).

Lele, Sanjiva K. and Joseph W. Nichols (2014). "A second golden age of aeroacoustics?" In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 372.2022, p. 20130321 (Cited in page 59).

Lemons, D.S., P. Langevin, and A. Gythiel (2002). *An Introduction to Stochastic Processes in Physics*. Johns Hopkins Paperback. Johns Hopkins University Press (Cited in page 155).

Lewis, Frank L., Draguna Vrabie, and Kyriakos G. Vamvoudakis (2012). "Reinforcement Learning and Feedback Control: Using Natural Decision Methods to Design Optimal Adaptive Controllers". In: *IEEE Control Systems Magazine* 32.6, pp. 76–105 (Cited in page 36).

Li, Weiwei and Emanuel Todorov (Oct. 26, 2004). "Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems." In: *ICINCO (1)*. Ed. by Helder Araújo, Alves Vieira, José Braz, Bruno Encarnação, and Marina Carvalho. INSTICC Press, pp. 222–229 (Cited in page 148).

Lindquist, Anders (1990). "Linear Stochastic Systems (Peter E. Caines)". In: *SIAM Review* 32.2, pp. 325–328 (Cited in page 36).

Liu, Han, Kathryn Roeder, and Larry Wasserman (2010). "Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models". In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., 1432–1440 (Cited in page 73).

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR (Cited in page 52).

Luketina, Jelena, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel (July 2019). "A Survey of Reinforcement Learning Informed by Natural Language". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 6309–6317 (Cited in page 4).

Lutter, Michael, Shie Mannor, Jan Peters, Dieter Fox, and Animesh Garg (2021). "Value iteration in continuous actions, states and time". In: *Proceedings of the International Conference on Machine Learning* (Cited in page 36).

Mangasarian, O. L. (1966). "Sufficient Conditions for the Optimal Control of Nonlinear Systems". In: *SIAM Journal on Control* 4.1, pp. 139–152 (Cited in page 107).

Mann, Jordan and J. Nathan Kutz (2016). "Dynamic mode decomposition for financial trading strategies". In: *Quantitative Finance* 16.11, pp. 1643–1655 (Cited in page 59).

Marconato, Anna, Maarten Schoukens, and Johan Schoukens (Oct. 2016). "Filter-based regularisation for impulse response modelling". In: *IET Control Theory & Applications* 11 (Cited in page 84).

Mauroy, Alexander and Igor Mezić (2012). "On the use of Fourier averages to compute the global isochrons of (quasi)periodic dynamics". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.3, p. 033112 (Cited in page 105).

Mauroy, Alexandre (2021). “Koopman Operator Framework for Spectral Analysis and Identification of Infinite-Dimensional Systems”. In: *Mathematics* 9.19 (Cited in page 7).

Mauroy, Alexandre and Jorge M. Gonçalves (2020). “Koopman-Based Lifting Techniques for Nonlinear Systems Identification”. In: *IEEE Transactions on Automatic Control* 65.6, pp. 2550–2565 (Cited in pages 53, 60, 61, 63, 84, 90, 99).

Mauroy, Alexandre and Igor Mezić (2016). “Global Stability Analysis Using the Eigenfunctions of the Koopman Operator”. In: *IEEE Transactions on Automatic Control* 61.11, pp. 3356–3369 (Cited in page 105).

Mauroy, Alexandre, Igor Mezić, and Jeff Moehlis (2013). “Isostables, isochrons, and Koopman spectrum for the action-angle representation of stable fixed point dynamics”. In: *Physica D: Nonlinear Phenomena* 261, pp. 19–30 (Cited in page 105).

Mauroy, Alexandre, Igor Mezić, and Susuki Yoshihiko, eds. (2020). *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications.* Springer Cham, pp. XXIII, 556 (Cited in pages 7, 15, 18–20, 26, 30, 51, 109, 110).

Mauroy, Alexandre and Aivar Sootla (Dec. 2017). “Geometric properties of isostables and basins of attraction of monotone systems”. English. In: *IEEE Transactions on Automatic Control* 62.12, pp. 6183–6194 (Cited in page 105).

McShane, E. J. (1989). “The Calculus of Variations from the Beginning Through Optimal Control Theory”. In: *SIAM Journal on Control and Optimization* 27.5, pp. 916–939 (Cited in page 107).

Meinshausen, Nicolai and Peter Bühlmann (2010). “Stability Selection”. In: *Journal of the Royal Statistical Society, Series B* 72, pp. 417–473 (Cited in page 72).

Mendelson, Shahar and Joseph Neeman (2010). “Regularization in kernel learning”. en. In: *Ann. Statist.* 38.1, pp. 526–565 (Cited in page 67).

Mezić, Igor (2013). “Analysis of Fluid Flows via Spectral Properties of the Koopman Operator”. In: *Annual Review of Fluid Mechanics* 45.1, pp. 357–378 (Cited in pages 58, 83).

— (Aug. 2005). “Spectral Properties of Dynamical Systems, Model Reduction and Decompositions”. In: *Nonlinear Dynamics* 41, pp. 309–325 (Cited in pages 58, 105).

Mezić, Igor (2015). “On applications of the spectral theory of the Koopman operator in dynamical systems and control theory”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 7034–7041 (Cited in page 105).

— (2020). “Spectrum of the Koopman Operator, Spectral Expansions in Functional Spaces, and State-Space Geometry”. In: *Journal of Nonlinear Science* 30.5, pp. 2091–2145 (Cited in page 7).

Mezić, Igor and Andrzej Banaszuk (2004). "Comparison of systems with complex behavior". In: *Physica D: Nonlinear Phenomena* 197, pp. 101–133 (Cited in pages 18, 59, 105).

Mezić, Igor and Stephen Wiggins (1999). "A method for visualization of invariant sets of dynamical systems based on the ergodic partition". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 9.1, pp. 213–218 (Cited in page 105).

Mishura, Yu, S Posashkova, and S Posashkov (2011). "Continuous dependence of solutions of stochastic differential equations driven by standard and fractional Brownian motion on a parameter". In: *Theory of Probability and Mathematical Statistics* 83, pp. 111–126 (Cited in page 133).

Mohr, Ryan and Igor Mezi'c (2014). "Construction of eigenfunctions for scalar-type operators via Laplace averages with connections to the Koopman operator". In: *arXiv: Spectral Theory* (Cited in page 59).

Montazeri, Hesam, Sajjad Moradi, and Reza Safabakhsh (2011). "Continuous state/action reinforcement learning: A growing self-organizing map approach". In: *Neurocomputing* 74.7, pp. 1069–1082 (Cited in page 127).

Moore, G.H. (2012). *Zermelo's Axiom of Choice: Its Origins, Development, and Influence*. Dover books on mathematics. Dover Publications (Cited in page 67).

Müller, David, Andreas Otto, and Günter Radons (June 2017). "From dynamical systems with time-varying delay to circle maps and Koopman operators". In: *Physical Review E* 95 (6), p. 062214 (Cited in page 22).

Munos, Rémi (2006). "Policy gradient in continuous time". In: *Journal of Machine Learning Research* 7 (Cited in page 36).

Munos, Rémi and Paul Bourgine (1997). "Reinforcement learning for continuous stochastic control problems". In: *Advances in neural information processing systems* 10 (Cited in page 36).

Narici, L. and E. Beckenstein (2010). *Topological Vector Spaces*. Chapman & Hall/CRC Pure and Applied Mathematics. CRC Press (Cited in pages 24, 68).

Ng, Andrew Y. and Michael Jordan (2000). "PEGASUS: A Policy Search Method for Large MDPs and POMDPs". In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. UAI'00. Stanford, California: Morgan Kaufmann Publishers Inc., 406–415 (Cited in page 127).

Nikodým, Otton Martin (1930). "Sur une généralisation des intégrales de M. J. Radon". In: *Fundamenta Mathematicae* 15, pp. 131–179 (Cited in page 28).

Oppenheim, A.V., A.S. Willsky, and I.T. Young (1983). *Signals and Systems*. Prentice-Hall signal processing series. Prentice-Hall (Cited in page 65).

Osband, Ian, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy (2016). "Deep Exploration via Bootstrapped DQN". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc. (Cited in page 140).

Pavliotis, G.A. (2014). *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*. Texts in Applied Mathematics. Springer - New York (Cited in pages 30, 55).

Phillips, Rob, Jane Kondev, and Julie Theriot (Nov. 2008). *Physical Biology of the Cell*. New York: Garland Science, Taylor & Francis Group (Cited in page 100).

Piga, Dario, Simone Formentin, and Alberto Bemporad (2018). "Direct Data-Driven Control of Constrained Systems". In: *IEEE Transactions on Control Systems Technology* 26.4, pp. 1422–1429 (Cited in page 6).

Pillonetto, G., A. Chiuso, and G. De Nicolao (Feb. 2011). "Prediction error identification of linear systems: a nonparametric Gaussian regression approach". In: *Automatica* 47, pp. 291–305 (Cited in page 84).

Pontryagin, L.S., V.G. Boltyanskii, R.V. Gamkrelidze, E.F. Mishchenko, and D.E. Brown (1962). *The Mathematical Theory of Optimal Processes*. International series of monographs in pure and applied mathematics. Wiley - New York (Cited in page 107).

Proctor, Joshua L., Steven L. Brunton, and J. Nathan Kutz (2016). "Dynamic Mode Decomposition with Control". In: *SIAM Journal on Applied Dynamical Systems* 15.1, pp. 142–161 (Cited in pages 117, 118).

— (2018). "Generalizing Koopman Theory to Allow for Inputs and Control". In: *SIAM Journal on Applied Dynamical Systems* 17.1, pp. 909–930 (Cited in pages 109, 111, 118).

Prugovečki, E. (1971). *Quantum Mechanics in Hilbert Space*. Pure and applied mathematics : a series of monographs and textbooks. Academic Press (Cited in page 67).

Puterman, Martin L. and Moon Chirl Shin (1978). "Modified Policy Iteration Algorithms for Discounted Markov Decision Problems". In: *Management Science* 24.11, pp. 1127–1137 (Cited in page 124).

Racanière, Sébastien et al. (2017). "Imagination-Augmented Agents for Deep Reinforcement Learning". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 5694–5705 (Cited in page 6).

Rantzer, Anders (2001). "A dual to Lyapunov's stability theorem". In: *Systems & Control Letters* 42.3, pp. 161–168 (Cited in page 105).

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, pp. I–XVIII, 1–248 (Cited in pages 86, 92, 93, 152).

Recht, Benjamin (2018). "A Tour of Reinforcement Learning: The View from Continuous Control." In: *CoRR* abs/1806.09460 (Cited in pages 121, 147).

Reid, W.T. (1972). *Riccati Differential Equations*. Mathematics in science and engineering: a series of monographs and textbooks. Academic Press (Cited in page 113).

Renardy, M. and R.C. Rogers (2004). *An Introduction to Partial Differential Equations*. Texts in Applied Mathematics. Springer - New York (Cited in page 26).

Riesz, F. (1907). "Sur une espèce de géométrie analytique des systèmes de fonctions sommables." French. In: *C. R. Acad. Sci., Paris* 144, pp. 1409–1411 (Cited in page 28).

Rowley, Clarence W., Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson (2009). "Spectral analysis of nonlinear flows". In: *Journal of Fluid Mechanics* 641, 115–127 (Cited in pages 59, 83).

Royden, H.L. and R.H. L (1988). *Real Analysis*. Mathematics and statistics. Macmillan (Cited in page 26).

Rudin, Walter (1953). *Principles of mathematical analysis*. McGraw-Hill Book Company, Inc., New York-Toronto-London, pp. ix+227 (Cited in page 124).

— (1987). *Real and Complex Analysis, 3rd Ed.* USA: McGraw-Hill, Inc. (Cited in page 67).

Rummery, G. A. and M. Niranjan (1994). *On-Line Q-Learning Using Connectionist Systems*. Tech. rep. TR 166. Cambridge, England: Cambridge University Engineering Department (Cited in page 148).

Ryan, Richard M. and Edward L. Deci (2000). "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions". In: *Contemporary Educational Psychology* 25.1, pp. 54–67 (Cited in page 126).

Sabatier, P.C. (1987). "A FEW GEOMETRICAL FEATURES OF INVERSE AND ILL-POSED PROBLEMS". In: *Inverse and Ill-Posed Problems*. Ed. by Heinz W. Engl and C.W. Groetsch. Academic Press, pp. 1–18 (Cited in page 65).

Schaefer, H. H. (1966). *Topological Vector Spaces*. Macmillan New York (Cited in page 29).

Schechter, E. (1996). *Handbook of Analysis and Its Foundations*. Elsevier Science (Cited in page 69).

Schmelzer, Bernhard (2010). "On solutions of stochastic differential equations with parameters modeled by random sets". In: *International Journal of Approximate Reasoning* 51.9. Imprecise probability in statistical inference and decision making, pp. 1159–1171 (Cited in page 133).

Schmidhuber, J. (1991). "Curious model-building control systems". In: *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, 1458–1463 vol.2 (Cited in page 126).

Schölkopf, Bernhard, Ralf Herbrich, and Alex J. Smola (2001). "A Generalized Representer Theorem". In: *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*. COLT '01/EuroCOLT '01. Berlin, Heidelberg: Springer-Verlag, 416–426 (Cited in pages 66, 69, 70, 86).

Schölkopf, Bernhard, Patrice Simard, Alex Smola, and Vladimir Vapnik (1997). "Prior Knowledge in Support Vector Kernels". In: *Advances in Neural Information Processing Systems*. Ed. by M. Jordan, M. Kearns, and S. Solla. Vol. 10. MIT Press (Cited in page 73).

Schrittwieser, Julian et al. (Dec. 2020). "Mastering Atari, Go, chess and shogi by planning with a learned model". In: *Nature* 588.7839, pp. 604–609 (Cited in pages 6, 148).

Schulman, John, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz (July 2015). "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1889–1897 (Cited in page 140).

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). "Proximal Policy Optimization Algorithms." In: *CoRR* abs/1707.06347 (Cited in page 140).

Seijen, Harm van, Hado van Hasselt, Shimon Whiteson, and Marco Wiering (2009). "A theoretical and empirical analysis of Expected Sarsa". In: *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pp. 177–184 (Cited in page 148).

Selvi, Daniela, Dario Piga, Giorgio Battistelli, and Alberto Bemporad (2021). "Optimal direct data-driven control with stability guarantees". In: *Eur. J. Control* 59, pp. 175–187 (Cited in page 6).

Shah, Rajen D. and Richard J. Samworth (Jan. 2013). "Variable selection with error control: another look at stability selection". In: *Journal of the Royal Statistical Society - Series B* 75.1, pp. 55–80 (Cited in page 73).

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, pp. I–XVI, 1–397 (Cited in pages 85, 117, 149).

Silver, David, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller (June 2014). "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, pp. 387–395 (Cited in page 127).

Silver, David et al. (Jan. 2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489 (Cited in page 4).

Sinclair, Sean R., Siddhartha Banerjee, and Christina Lee Yu (2019). "Adaptive discretization for episodic reinforcement learning in metric spaces". In: *Proceedings of the ACM Conference on Measurement and Analysis of Computing Systems* (Cited in page 37).

Singh, Satinder, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvári (Mar. 2000). "Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms". In: *Machine Learning* 38.3, pp. 287–308 (Cited in page 148).

Söderström, T. and P. Stoica (1988). *System Identification*. USA: Prentice-Hall, Inc. (Cited in page 52).

Sootla, Aivar and Alexandre Mauroy (2017). "Geometric Properties of Isostables and Basins of Attraction of Monotone Systems". In: *IEEE Transactions on Automatic Control* 62.12, pp. 6183–6194 (Cited in page 105).

Stadie, Bradly C., Sergey Levine, and Pieter Abbeel (2015). "Incentivizing Exploration In Reinforcement Learning With Deep Predictive Models". In: *CoRR* abs/1507.00814 (Cited in page 140).

Stodden, V., F. Leisch, and R.D. Peng (2014). *Implementing Reproducible Research*. Chapman & Hall/CRC The R Series. Taylor & Francis (Cited in page 72).

Strogatz, Steven H. (2000). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press (Cited in page 15).

Sun, Wei (2015). "Stability of machine learning algorithms". PhD thesis. Purdue University (Cited in page 72).

Sun, Wei, Junhui Wang, and Yixin Fang (2013). "Consistent Selection of Tuning Parameters via Variable Selection Stability". In: *Journal of Machine Learning Research* 14.107, pp. 3419–3440 (Cited in page 73).

Sun, Yue and Maryam Fazel (2021). "Learning Optimal Controllers by Policy Gradient: Global Optimality via Convex Parameterization". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. Austin, TX, USA: IEEE Press, 4576–4581 (Cited in pages 120, 128).

Susuki, Yoshihiko and Igor Mezić (2014). "Nonlinear Koopman modes and power system stability assessment without models". In: *2014 IEEE PES General Meeting | Conference & Exposition*, pp. 1–1 (Cited in page 105).

Sutton, Richard S (1995). "Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding". In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky, M.C. Mozer, and M. Hasselmo. Vol. 8. MIT Press (Cited in page 148).

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. Second. The MIT Press (Cited in pages 4, 5, 121, 124, 126, 127, 147).

Sutton, Richard S, David McAllester, Satinder Singh, and Yishay Mansour (1999). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Ed. by S. Solla, T. Leen, and K. Müller. Vol. 12. MIT Press (Cited in pages 127, 138).

Sutton, Richard Stuart (1984). "Temporal Credit Assignment in Reinforcement Learning". AAI8410337. PhD thesis (Cited in page 148).

Szepesvári, Csaba (2010). *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (Cited in page 122).

Taira, Kunihiko, Steven L. Brunton, Scott T. M. Dawson, Clarence W. Rowley, Tim Colonius, Beverley J. McKeon, Oliver T. Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S. Ukeiley (2017). "Modal Analysis of Fluid Flows: An Overview". In: *AIAA Journal* 55.12, pp. 4013–4041 (Cited in page 83).

Takeishi, Naoya (Nov. 2019). "Kernel Learning for Data-Driven Spectral Analysis of Koopman Operators". In: *Proceedings of The Eleventh Asian Conference on Machine Learning*. Ed. by Wee Sun Lee and Taiji Suzuki. Vol. 101. Proceedings of Machine Learning Research. PMLR, pp. 956–971 (Cited in page 85).

Tallec, Corentin, Léonard Blier, and Yann Ollivier (2019a). "Making Deep Q-learning methods robust to time discretization". In: *International Conference on Machine Learning (ICML)*. 97, pp. 6096–6104 (Cited in page 36).

— (2019b). "Making Deep Q-learning methods robust to time discretization". In: *International Conference on Machine Learning (ICML)*. 97, pp. 6096–6104 (Cited in page 49).

Tang, Haoran, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel (2017). "#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Pro-*

cessing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, pp. 2753–2762 (Cited in page 140).

Theodoridis, Sergios and Konstantinos Koutroumbas (2009). *Pattern Recognition, Fourth Edition*. Academic Press (Cited in page 85).

Tibshirani, Robert (1996). "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society (Series B)* 58, pp. 267–288 (Cited in page 117).

— (1997). "The lasso Method for Variable Selection in the Cox Model". In: *Statistics in Medicine* 16.4, pp. 385–395 (Cited in page 117).

Todorov, Emanuel, Tom Erez, and Yuval Tassa (2012). "Mujoco: A physics engine for model-based control". In: *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, pp. 5026–5033 (Cited in pages 48, 49).

Tsitsiklis, John N. (Sept. 1994). "Asynchronous Stochastic Approximation and Q-Learning". In: *Journal of Machine Learning Research* 16.3, 185–202 (Cited in page 148).

Tu, Jonathan H., Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz (2014). "On dynamic mode decomposition: Theory and applications". In: *Journal of Computational Dynamics* 1.2, pp. 391–421 (Cited in pages 7, 51, 59, 61, 84).

Tu, Stephen and Benjamin Recht (2018). "Least-squares temporal difference learning for the linear quadratic regulator". In: *International Conference on Machine Learning*. PMLR, pp. 5005–5014 (Cited in page 37).

— (June 2019a). "The Gap Between Model-Based and Model-Free Methods on the Linear Quadratic Regulator: An Asymptotic Viewpoint". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, pp. 3036–3083 (Cited in pages 5, 37, 148).

— (June 2019b). "The Gap Between Model-Based and Model-Free Methods on the Linear Quadratic Regulator: An Asymptotic Viewpoint". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. PMLR, pp. 3036–3083 (Cited in page 148).

Vaidya, Umesh and Prashant G. Mehta (2008). "Lyapunov Measure for Almost Everywhere Stability". In: *IEEE Transactions on Automatic Control* 53.1, pp. 307–323 (Cited in page 105).

Veillard, Antoine, Daniel Racoceanu, and Stephane Bressan (2011). "Incorporating Prior-Knowledge in Support Vector Machines by Kernel Adaptation". In: *2011 IEEE 23rd*

International Conference on Tools with Artificial Intelligence, pp. 591–596 (Cited in page 73).

Vogel, Curtis R. (1987). "An Overview of Numerical Methods for Nonlinear Ill-Posed Problems". In: *Inverse and Ill-Posed Problems*. Ed. by Heinz W. Engl and C.W. Groetsch. Academic Press, pp. 231–245 (Cited in page 65).

Wahba, Grace (2003). "An introduction to reproducing kernel hilbert spaces and why they are so useful". In: *IFAC Proceedings Volumes* 36.16. 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27-29 August, 2003, pp. 525–528 (Cited in page 9).

— (1983). "Bayesian "Confidence Intervals" for the Cross-Validated Smoothing Spline". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 45.1, pp. 133–150 (Cited in page 11).

Wang, Haoran, Thaleia Zariphopoulou, and Xun Yu Zhou (2020). "Reinforcement Learning in Continuous Time and Space: A Stochastic Control Approach". In: *Journal of Machine Learning Research* 21.198, pp. 1–34 (Cited in page 37).

Watkins, C. J. C. H. (1989). "Learning from Delayed Rewards". PhD thesis. King's College, Oxford (Cited in pages 124, 148).

Watkins, Christopher J. C. H. and Peter Dayan (May 1992). "Q-learning". In: *Machine Learning* 8.3, pp. 279–292 (Cited in page 148).

Wiggins, S. (2003). *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Texts in Applied Mathematics. Springer - New York (Cited in page 15).

Willems, J.L. (1970). *Stability Theory of Dynamical Systems*. Major Issues in History. Wiley Interscience Division (Cited in page 106).

Williams, Matthew O., Maziar S. Hemati, Scott T.M. Dawson, Ioannis G. Kevrekidis, and Clarence W. Rowley (2016). "Extending Data-Driven Koopman Analysis to Actuated Systems". In: *IFAC-PapersOnLine* 49.18. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016, pp. 704–709 (Cited in pages 118, 120).

Williams, Matthew O., Ioannis G. Kevrekidis, and Clarence W. Rowley (2015). "A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition". In: *Journal of Nonlinear Science* 25.6, pp. 1307–1346 (Cited in pages 7, 51, 59, 84, 93).

Williams, Matthew O., Clarence W. Rowley, and Ioannis G. Kevrekidis (2015). "A kernel-based method for data-driven koopman spectral analysis". In: *Journal of Computational Dynamics* 2.2, pp. 247–265 (Cited in pages 85, 86).

Williams, R. J. (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8, pp. 229–256 (Cited in pages 124, 148).

Williams, Ronald J. (1988). *Toward a theory of reinforcement-learning connectionist systems*. Tech. rep. NU-CCS-88-3. Northeastern University, College of Computer Science (Cited in pages 124, 148).

Williamson, Robert and Ludvik Janos (1987). "Constructing metrics with the Heine-Borel property". In: (Cited in page 69).

Witten, I. H. (1977). "An Adaptive Optimal Controller for Discrete-Time Markov Environments". In: *Information and Control* 34, pp. 286–295 (Cited in page 148).

Yasinsky, V. K. and I. V. Malyka (Nov. 2012). "Parametric continuity of solutions to stochastic functional differential equations with poisson perturbations". In: *Cybernetics and Systems Analysis* 48.6, pp. 846–860 (Cited in page 133).

Yildiz, Cagatay, Markus Heinonen, and Harri Lähdesmäki (2021). "Continuous-time Model-based Reinforcement Learning". In: *International Conference on Machine Learning*. PMLR, pp. 12009–12018 (Cited in page 36).

Zanini, Francesco and Alessandro Chiuso (2021a). "Data-Driven Control of Nonlinear Systems: Learning Koopman Operators for Policy Gradient". In: *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6491–6496 (Cited in pages 121, 126).

— (2021b). "Estimating Koopman operators for nonlinear dynamical systems: a nonparametric approach". In: *IFAC-PapersOnLine* 54.7. 19th IFAC Symposium on System Identification SYSID 2021, pp. 691–696 (Cited in page 83).

— (2022). "Value function estimation in Reinforcement Learning: a Koopman operator approach." In: *[ACCEPTED AT] 2022 61th IEEE Conference on Decision and Control (CDC)*, pp. 6491–6496 (Cited in page 147).

Zanini, Francesco, Vincent Zhang, Johannes Kirschner, Junxi Zhang, Alex Ayoub, Masood Dehghan, and Dale Schuurmans (2023). "Managing temporal resolution in continuous value estimation: a fundamental trade-off". In: *[TO APPEAR AT] The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali Rwanda, May 1-5, 2022* (Cited in page 35).

Zermelo, E. (1904). "Beweis, daß jede Menge wohlgeordnet werden kann. (Aus einem an Herrn Hilbert gerichteten Briefe)". In: *Mathematische Annalen* 59, pp. 514–516 (Cited in page 25).

Zhang, Hao, Clarence W. Rowley, Eric A. Deem, and Louis N. Cattafesta (2019). "Online Dynamic Mode Decomposition for Time-Varying Systems". In: *SIAM Journal on Applied Dynamical Systems* 18.3, pp. 1586–1609 (Cited in page 22).

Zhang, Kaiqing, Alec Koppel, Hao Zhu, and Tamer Başar (2020). "Global Convergence of Policy Gradient Methods to (Almost) Locally Optimal Policies". In: *SIAM Journal on Control and Optimization* 58.6, pp. 3586–3612 (Cited in pages 120, 128).

Zheng, Yang, Luca Furieri, Maryam Kamgarpour, and Na Li (June 2021). "Sample Complexity of Linear Quadratic Gaussian (LQG) Control for Output Feedback Systems". In: *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. Ed. by Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger. Vol. 144. Proceedings of Machine Learning Research. PMLR, pp. 559–570 (Cited in page 149).