

Data Driven Science & Engineering

Machine Learning, Dynamical Systems, and Control

Steven L. Brunton

Department of Mechanical Engineering
University of Washington

J. Nathan Kutz

Department of Applied Mathematics
University of Washington

Chapter 11

Reinforcement Learning

Reinforcement learning (RL) is a major branch of machine learning that is concerned with how to learn control laws and policies to interact with a complex environment from experience [695, 369]. Thus, RL is situated at the growing intersection of control theory and machine learning [601], and it is among the most promising fields of research towards generalized artificial intelligence and autonomy. Both machine learning and control theory fundamentally rely on optimization, and likewise, RL involves a set of optimization techniques within an experiential framework for learning how to interact with the environment.

In reinforcement learning, an *agent*¹ senses the state of its environment and learns take appropriate actions to optimize future rewards. The ultimate goal in RL is to learn an effective control strategy or set of actions through positive or negative reinforcement. This search may involve trial-and-error learning, model-based optimization, or a combination of both. In this way, reinforcement learning is fundamentally biologically inspired, mimicking how animals learn to interact with their environment through positive and negative reward feedback from trial-and-error experience. Much of the history of reinforcement learning, and machine learning more broadly, has been linked to studies of animal behavior and the neurological basis of decisions, control, and learning [521, 657, 201, 199]. For example, Pavlov's dog is an illustration that animals learn to associate environmental cues with a food reward [563]. The term *reinforcement* refers to the rewards, such as food, used to reinforce desirable actions in humans and animals. However, in animal systems reinforcement is ultimately achieved through cellular and molecular learning rules.

Multiple textbooks have been written on this topic, which spans almost a century of progress. Major advances in deep reinforcement learning are also rapidly changing the landscape. This chapter is not meant to be comprehensive; rather, it aims to provide a solid foundation, to introduce key concepts and leading approaches, and to lower the barrier to entry in this exciting field.

¹Ironically, from the perspective of reinforcement learning, in *The Matrix*, Neo is actually the agent learning to interact with his environment.

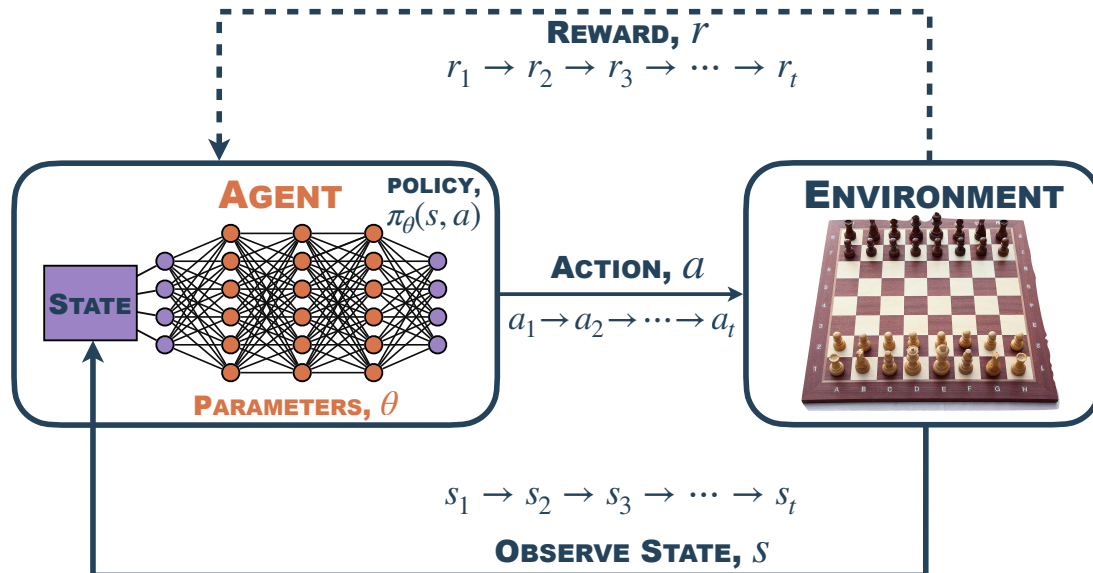


Figure 11.1: Schematic of reinforcement learning, where an agent senses its environmental state s and takes actions a according to a policy π that is optimized through learning to maximize future rewards r . In this case, a deep neural network is used to represent the policy π . This is known as a *deep policy network*.

11.1 Overview and Mathematical Formulation

Figure 11.1 provides a schematic overview of the reinforcement learning framework. An RL agent senses the state of its environment and learns to take appropriate actions to achieve optimal immediate or delayed rewards. Specifically, the RL agent arrives at a sequence of different states $s_k \in \mathcal{S}$ by performing actions $a_k \in \mathcal{A}$, with the selected actions leading to positive or negative rewards r_k used for learning. The sets \mathcal{S} and \mathcal{A} denote the sets of possible states and actions, respectively. Importantly, the RL agent is capable of learning delayed rewards, which is critical for systems where the optimal solution involves a multi-step procedure. Rewards may be thought of as sporadic and time-delayed labels, leading to RL being considered a third major branch of machine learning, called *semi-supervised* learning, which complements the other two branches of supervised and unsupervised learning. One canonical example is learning a set of moves, or a long term strategy, to win a game of chess. As is the case with human learning, RL often begins with an unstructured *exploration*, where trial-and-error are used to learn the rules, followed by *exploitation*, where a strategy is chosen and optimized within the learned rules.

The Policy

An RL agent senses the state of its environment s and takes actions a through a policy π that is optimized through learning to maximize future rewards r . Reinforcement learning is often formulated as an optimization problem to learn the policy $\pi(s, a)$,

$$\pi(s, a) = \Pr(a = a \mid s = s), \quad (11.1)$$

which is the probability of taking action a given state s , to maximize the total future rewards. In the simplest formulation, the policy may be a look-up table that is defined on the discrete state and action spaces \mathcal{S} and \mathcal{A} , respectively. However, for most problems, representing and learning this policy becomes prohibitively expensive, and π must instead be represented as an approximate function that is parameterized by a lower-dimensional vector θ :

$$\pi(s, a) \approx \pi(s, a, \theta). \quad (11.2)$$

Often, this parameterized function will be denoted $\pi_\theta(s, a)$. Function approximation is the basis of deep reinforcement learning in Sec. 11.4 where it is possible to represent these complex functions using deep neural networks.

Note that in the literature, there is often an abuse of notation, where $\pi(s, a)$ is used to denote the action taken, rather than the *probability* density of taking an action a given a state observation s . In the case of a deterministic policy, such as a greedy policy, then it may be possible to use $a = \pi(s)$ to represent the action taken. We will attempt to be clear throughout when choosing one convention over another.

The Environment: a Markov Decision Process (MDP)

In general, the measured state of the system may be a partial measurement of a higher-dimensional environmental state that evolves according to a stochastic, nonlinear dynamical system. However, for simplicity, most introductions to RL assume that the state evolves according to a Markov decision process (MDP), so that the probability of the system occurring in the current state is determined only by the previous state. We will begin with this simple formulation. However, even though it is often assumed that the state evolves according to an MDP, it is often the case that this model is not known, motivating the use of “*model-free*” RL strategies discussed in Sec. 11.3. Similarly, when a model is not known, it may be possible to first learn an MDP using data-driven methods and then use this for “*model-based*” reinforcement learning, as in Sec. 11.2.

An MDP consists of a set of states \mathcal{S} , a set of actions \mathcal{A} , and a set of rewards \mathcal{R} , along with the probability of transitioning from state s_k at time t_k to state s_{k+1} at time t_{k+1} given action a_k ,

$$P(s', s, a) = \Pr(s_{k+1} = s' \mid s_k = s, a_k = a), \quad (11.3)$$

and a reward function R

$$R(\mathbf{s}', \mathbf{s}, \mathbf{a}) = \Pr(\mathbf{r}_{k+1} \mid \mathbf{s}_{k+1} = \mathbf{s}', \mathbf{s}_k = \mathbf{s}, \mathbf{a}_k = \mathbf{a}). \quad (11.4)$$

Sometimes the transition probability $P(\mathbf{s}', \mathbf{s}, \mathbf{a})$ will be written as $P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$. Again, sometimes there will be an abuse of notation, where a chosen policy π will be used instead of the action \mathbf{a} in the argument of either P or R above. In this case, it is assumed that this applies a sum over states, as in

$$P(\mathbf{s}', \mathbf{s}, \pi) = \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{s}, \mathbf{a}) P(\mathbf{s}', \mathbf{s}, \mathbf{a}). \quad (11.5)$$

Thus, an MDP generalizes the notion of a Markov process to include actions and rewards, making it suitable for decision making and control. A simple Markov process is a set of states \mathcal{S} and a probability of transitioning from one state to the next. The defining property of a Markov process and an MDP is that the probability of being in a future state is entirely determined by the current state, and not by previous states or hidden variables. The MDP framework is closely related to transition state theory and the Perron-Frobenius operator, which is the adjoint of the Koopman operator from Section 7.4.

In the case of a simple Markov process with a finite set of states \mathcal{S} , then it is possible to let $\mathbf{s} \in \mathbb{R}^n$ be a vector of the probability of being in each of the n states, in which case the Markov process $P(\mathbf{s}', \mathbf{s})$ may be written in terms of a transition matrix, also known as a stochastic matrix, or a probability matrix, \mathbf{T} :

$$\mathbf{s}' = \mathbf{T}\mathbf{s}, \quad (11.6)$$

where each column of \mathbf{T} must add up to 1, which is a statement of conservation of probability that given a particular state \mathbf{s} , *something* must happen after the transition to \mathbf{s}' . Similarly, for an MDP, given a policy π , the transition process may be written as

$$\mathbf{s}' = \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{s}, \mathbf{a}) \mathbf{T}_a \mathbf{s}. \quad (11.7)$$

Now for each action \mathbf{a} , \mathbf{T}_a is a Markov process with all columns summing to 1.

One of the defining properties of a Markov process is that the system asymptotically approaches a steady state $\boldsymbol{\mu}$, which is the eigenvector of \mathbf{T} corresponding to eigenvalue 1. Similarly, given a policy π , an MDP asymptotically approaches a steady state $\boldsymbol{\mu}_\pi = \sum_a \pi(\mathbf{s}, \mathbf{a}) \boldsymbol{\mu}_a$.

This brings up another notational issue, where for continuous processes, $\mathbf{s} \in \mathbb{R}^n$ describes the continuous state vector in an n -dimensional vector space, as in Chapters 7 and 8, while for discrete state spaces, $\mathbf{s} \in \mathbb{R}^n$ denotes a vector of probabilities of belonging to one of n finite states. It is important to carefully consider which notation is being used for a given problem, as these formulations have different dynamics (i.e., differential equation versus MDP) and interpretations (i.e., deterministic dynamics versus probabilistic transitions).

The Value Function

Given a policy π , we next define a value function that quantifies the desirability of being in a given state:

$$V_{\pi}(\mathbf{s}) = \mathbb{E} \left(\sum_k \gamma^k \mathbf{r}_k \mid \mathbf{s}_0 = \mathbf{s} \right), \quad (11.8)$$

where \mathbb{E} is the expected reward over the time steps k , subject to a *discount rate* γ . Future rewards are discounted, reflecting the economic principle that current rewards are more valuable than future rewards. Often, the subscript π is omitted from the value function, in which case we refer to the value function for the best possible policy:

$$V(\mathbf{s}) = \max_{\pi} \mathbb{E} \left(\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_k \mid \mathbf{s}_0 = \mathbf{s} \right). \quad (11.9)$$

One of the most important properties of the value function is that the value at a state \mathbf{s} may be written recursively as

$$V(\mathbf{s}) = \max_{\pi} \mathbb{E} \left(\mathbf{r}_0 + \sum_{k=1}^{\infty} \gamma^k \mathbf{r}_k \mid \mathbf{s}_1 = \mathbf{s}' \right), \quad (11.10)$$

which implies that

$$V(\mathbf{s}) = \max_{\pi} \mathbb{E} (\mathbf{r}_0 + \gamma V(\mathbf{s}')), \quad (11.11)$$

where $\mathbf{s}' = \mathbf{s}_{k+1}$ is the next state after $\mathbf{s} = \mathbf{s}_k$ given action \mathbf{a}_k , and the expectation is over actions selected from the optimal policy π . This expression, known as *Bellman's equation*, is a statement of Bellman's principle of optimality, and it is a central result that underpins modern RL.

Given the value function, it is possible to extract the optimal policy as

$$\pi = \operatorname{argmax}_{\pi} \mathbb{E} (\mathbf{r}_0 + \gamma V(\mathbf{s}')), \quad (11.12)$$

Goals and Challenges of Reinforcement Learning

Learning the policy π , the value function V , or jointly learning both, is the central challenge in RL. Depending on the assumed structure of π , the size and evolution dynamics of \mathcal{S} , and the reward landscape R , determining an optimal policy may range from a closed form optimization to a rather high-dimensional unstructured optimization. Thus, a large number of trials must often be evaluated in order to determine an optimal policy. In practice, reinforcement learning may be very expensive to train, and it might not be the right strategy for

problems where testing a policy is expensive or potentially unsafe. Similarly, in many cases, there are simpler control strategies than RL, such as LQR or MPC; when these approaches are effective, they are often preferable. Reinforcement learning is, therefore, well-suited for situations where some combination of the following are true: evaluating a policy is inexpensive, as in board games; there are sufficient resources to perform a near brute-force optimization, as in evolutionary optimization; no other control strategy works.

Although RL is typically formulated within the mathematical framework of MDPs, many real world applications do not satisfy these assumptions. For example, the dynamics may depend on the state history or on hidden or latent variables. Similarly, the evolution dynamics may be entirely deterministic, yet chaotic. However, as we will see, it is often possible to develop approximate probabilistic transition state models for chaotic dynamics or to augment the environment state to include past states for systems with memory or hidden variables. Often, the underlying MDP transition probability and reward functions are not known *a priori*, and must either be learned ahead of time through some exploration phase, or alternative model-free optimization techniques must be used. Finally, many of the theoretical convergence results, and indeed many of the fundamental RL algorithms, only apply to *finite* MDPs, which are characterized by finite actions \mathcal{A} and states \mathcal{S} . Games, such as chess, fall into this category, even though the number of states may be combinatorially large. Even continuous dynamical systems, such as a pendulum on a cart, may be approximated by a finite MDP through a discretization or quantization process.

There is typically much less supervisory information available to an RL agent than is available in classical supervised learning. One of the central challenges of reinforcement learning is that rewards are often extremely rare and may be significantly delayed from a sequence of good control actions. This challenge leads to the so-called credit assignment problem, coined by Minsky [514] to describe the challenge of knowing what action sequence was responsible for the reward ultimately received. These sparse and delayed rewards have been a central challenge in RL for six decades, and they are still a focus of research today. The resulting optimization problem is computationally expensive and data intensive, requiring considerable trial and error.

Today, reinforcement learning is being used to learn sophisticated control policies for complex open-world problems in autonomy and propulsion (e.g., self-driving cars, learning to swim and fly, etc.) and as a general learning environment for rule-constrained games (e.g., checkers, backgammon, chess, go, Atari, etc.). Much of the history of RL may be traced through the success on increasingly challenging board games, from checkers [628] to backgammon [712] and more recently to chess and go [671]. These games serve to illustrate many of the central challenges that are still faced in RL, including the curse of dimensionality and the credit assignment problem.

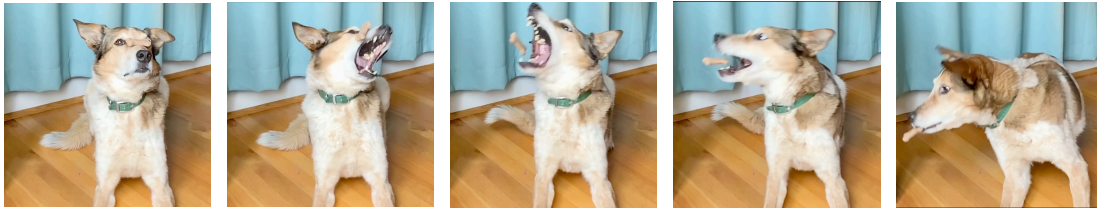


Figure 11.2: Reinforcement learning is inspired by biological learning with sparse rewards. Mordecai is trained to balance a treat on his nose until a command is given, after which he grabs it out of the air. *Credit: Bing Brunton for image and training.*

Motivating examples

It is helpful to understand RL through simple examples. Consider a mouse in a maze. The mouse is the agent, and the environment is the maze. The mouse measures the local state in its environment; it does not have access to a full top-down view of the maze, but instead it knows its current local environment and what past actions it has taken. The mouse has *agency* to take some action about what to do next, for example whether to turn left, turn right, or go forward. Typically, the mouse does not receive a reward until the end of the maze. If the mouse received a reward after each correct turn, it would have a much simpler learning task. Setting such a curriculum is a strategy to help teach animals, whereby initially dense rewards are sparsified throughout the learning process.

More generally, RL may be used to understand animal behavior, ranging from semi-supervised training to naturalistic behaviors. Figure 11.2 shows a trained behavior where a treat is balanced on Mordecai's nose until a command is given, after which he is able to grab it out of the air. Often, training animals to perform complex tasks involves expert human guidance to provide intermediate rewards or secondary reinforcers, such as using a clicker to indicate a future reward. In animal training and in RL, the more proximal the reward is in time to the action, the easier it is to learn the task. The connection between learning and temporal proximity is the basis of *temporal difference* learning, which is a powerful concept in RL, and this is also important to our understanding of the chemical basis for addiction [605].

It is also helpful to consider two-player games, such as tic-tac-toe, checkers, backgammon, chess, and go. In these games, the agent is one of the players, and the environment encompasses the rules of the game along with an adversarial opponent. These examples are also interesting because there is an element of randomness or stochasticity in the environment, either because of the fundamental rules (e.g., a dice-roll in backgammon) or because of an opponent's probabilistic strategy. Thus, it may be advantageous for the agent to also adopt a probabilistic policy, which is in contrast to much of the theory of classical control for deterministic systems. Similarly, a probabilistic strategy may be

important when learning how to play.

In most games, the reward signal comes at the end of the game after the agent has won or lost. Again, this makes the learning process exceedingly challenging, as it is initially unclear which subsequence of actions were particularly important in driving the outcome. For example, an agent may play an excellent chess opening and midgame and then lose at the end because of a few bad moves. Should the agent discard the entire first half of the game, or worse yet, attribute this to a negative reward? Thus, it is clear that a major part of learning an effective policy is understanding the value of being in a given state s . In a game like chess, where the number of states is combinatorially large, there are too many states to count, and it is intractable to map out the exact value of all board states. Instead, players create simple heuristic rules-of-thumb about what are good board positions, e.g. assigning points to the various pieces to keep track of a rough score. This intermediate score provides a denser reward structure throughout the game. However, these heuristics are sub-optimal and may be susceptible to gambits, where the opponent sacrifices a piece for an immediate point loss in order to eventually move to a more favorable global state s . In backgammon, an intermediate point total may be more explicitly computed as the total number of *pips*, or points that a player must roll to move all pieces home and off the board. Although this makes it relatively simple to estimate the strength of a board position, the discrete nature of the die roll and game mechanics makes this a sub-optimal approximation, as the number of required *dice-rolls* or *turns* may also be a useful measure.

Thinking through games like these illustrates many of the modern strategies to improve the learning rates and sample efficiency of RL, including hindsight replay, temporal difference learning, look ahead, and reward shaping, which we will discuss in the following sections. For example, playing against a skilled teacher can dramatically improve the learning rate, as the teacher provides guidance about whether or not a move is good, and why, adding information to help shape proxy metrics that can be used as intermediate rewards and models that can accelerate the learning process.

Categorization of RL Techniques

Nearly all problems in machine learning and control theory involve challenging optimization problems. In the case of machine learning, the parameters of a model are optimized to best fit the training data, as measured by a loss function. In the case of control, a set of control performance metrics are optimized subject to the constraints of the dynamics. Reinforcement learning is no different, as it is at the intersection of machine learning and control theory.

There are many approaches to learn an optimal policy π , which is the ultimate goal of RL. A major dichotomy in reinforcement learning is that of *model-*

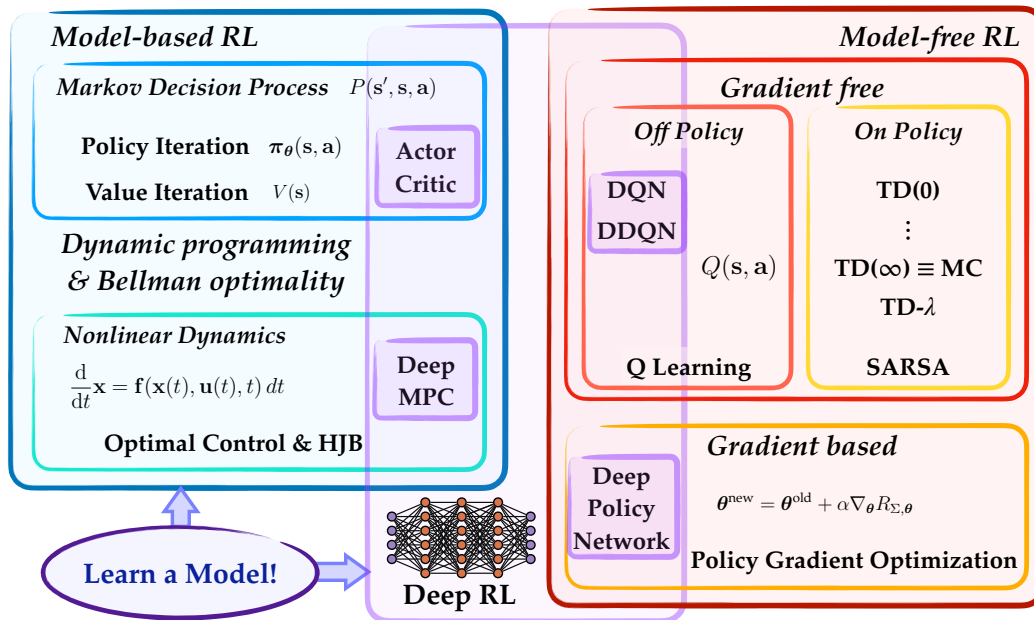


Figure 11.3: Rough categorization of reinforcement learning techniques. This organization is not comprehensive, and some of the lines are becoming blurred. The first major dichotomy is between model-based and model-free RL techniques. Next, within model-free RL, there is a dichotomy between gradient-based and gradient-free methods. Finally, within gradient-free methods, there is a dichotomy between on-policy and off-policy methods.

based RL versus model-free RL. When there is a known model for the environment, there are several strategies for learning either the optimal policy or value function through what is known as *policy iteration* or *value iteration*, which are forms of dynamic programming using the Bellman equation. When there is no model for the environment, alternative strategies, such as Q-learning, must be employed. The reinforcement learning optimization problem may be particularly challenging for high-dimensional systems with unknown, nonlinear, stochastic dynamics and sparse and delayed rewards. All of these techniques may be combined with function approximation techniques, such as neural networks, for approximating the policy π , the value function V , or the quality function Q (discussed in subsequent sections), making them more useful for high-dimensional systems. These model-based, model-free, and deep learning approaches will be discussed below. Note that this section only provides a glimpse of the many optimization approaches used to solve RL problems, as this is a vast and rapidly growing field.

11.2 Model-Based Optimization and Control

This section provides a high-level overview of some essential model-based optimization and control techniques. Some don't consider these techniques to be reinforcement learning, as they don't involve learning an optimal strategy through trial-and-error experience. However, they are closely related. It is possible to learn a model through trial-and-error, and then use this model with these techniques, which would be considered RL.

For the simplified case of a known model that is a finite MDP, it is possible to learn either the optimal policy or value function through what is known as *policy iteration* or *value iteration*, which are forms of dynamic programming using the Bellman equation. Dynamic programming [70, 71, 83, 735, 81, 82, 618] is a powerful approach that is used for general optimal nonlinear control and reinforcement learning, among other tasks. These algorithms provide a mathematically simplified optimization framework that helps to introduce essential concepts used throughout.

More generally, RL optimization is related to the field of optimal nonlinear control, which has deep roots in variational theory going back to Bernoulli and the Brachistochrone problem nearly four centuries ago. We will explore this connection to nonlinear control theory in Sec. 11.6.

Dynamic programming

Dynamic programming is a mathematical framework introduced by Richard E. Bellman [70, 71] to solve large multi-step optimization problems, such as those found in decision making and control. Policy iteration and value iteration, discussed below, are two examples of the use of dynamic programming in reinforcement learning. To solve these multi-step optimizations, dynamic programming reformulates the large optimization problem as a recursive optimization in terms of smaller sub-problems, so that only a local decision need be optimized. This approach relies on Bellman's principle of optimality, which states that a large multi-step control policy must also be locally optimal in every sub-sequence of steps.

The Bellman equation in (11.11) indicates that the large optimization problem over an entire state-action trajectory (s_k, a_k) may be broken into a recursive optimization at each point along the trajectory. As long as the value function is known at the next point s' , it is possible to solve the optimization at point s simply by optimizing the policy $\pi(s, a)$ at this point. Of course, this assumes that the value function is known at *all* possible next states s_{k+1} , which is a function of the current state s_k , the current action a_k , and the dynamics governing the system; this becomes even more complex for non-MDP dynamics, such as the nonlinear control formulation in the next subsection. For even moderately

large problems, this suffers from the curse of dimensionality, and approximate solution methods must be employed.

When tractable, dynamic programming (i.e., the process of breaking a large problem into smaller overlapping sub-problems) provides a globally optimal solution. There are two main approaches to dynamic programming, referred to as *top down* and *bottom up*:

Top down: The top-down approach involves maintaining a table of sub-problems that are referred to when solving larger problems. For a new problem, the table is checked to see if the relevant sub-problem has been solved. If so, it is used, and if not, the sub-problem is solved. This tabular storage is called *memoization* and becomes combinatorially complex for many problems.

Bottom up: The bottom-up approach involves starting by solving the smallest sub-problems first, and then combining these to form the larger problems. This may be thought of as working backwards from every possible goal state, finding the best previous action to get there, then going back two steps, then going back three steps, etc.

Although dynamic programming still represents a brute-force search through all sub-problems, it is still more efficient than a naive brute-force search. In some cases, it reduces the computational complexity to an algorithm that scales linearly with the number of sub-problems, although this may still be combinatorially large, as in the example of the game of chess. Dynamic programming is closely related to divide-and-conquer techniques, such as quick sort, except that divide-and-conquer applies to *non-overlapping* or *non-recursive* (i.e., independent) sub-problems, while dynamic programming applies to overlapping, or recursively interdependent sub-problems.

However, the recursive strategy suggests approximate solution techniques, such as the alternating directions method, where a sub-optimal solution is initialized and the value function is iterated over. This will be discussed next.

Policy iteration

Policy iteration is a two step optimization procedure to simultaneously find an optimal value function V_π and the corresponding optimal policy π .

First, a candidate policy π is evaluated, resulting in the value function for this fixed policy. This typically involves a brute force calculation of the value function for this policy starting at many or all initial states. The policy may need to be simulated for a long time depending on the reward delay and discounting factor γ .

Next, the value function is fixed, and the policy is optimized to improve the expected rewards by taking different actions at a given state. This optimization

relies on the alternative recursive formulation of the value function in (11.8) due to Bellman's equation (11.11):

$$V_{\pi}(\mathbf{s}) = \mathbb{E}(R(\mathbf{s}', \mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) + \gamma V_{\pi}(\mathbf{s}')) \quad (11.13a)$$

$$= \sum_{\mathbf{s}'} P(\mathbf{s}' | \mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) (R(\mathbf{s}', \mathbf{s}, \boldsymbol{\pi}(\mathbf{s})) + \gamma V_{\pi}(\mathbf{s}')). \quad (11.13b)$$

Note that in this expression, we have assumed a deterministic policy $\mathbf{a} = \boldsymbol{\pi}(\mathbf{s})$, otherwise, (11.13b) would involve a second summation over $\mathbf{a} \in \mathcal{A}$, with the expression multiplied by $\boldsymbol{\pi}(\mathbf{s}, \mathbf{a})$.

It is then possible to fix $V_{\pi}(\mathbf{s}')$ and optimize over the policy in the first term. In particular, the new deterministic optimal policy at the state \mathbf{s} is given by:

$$\boldsymbol{\pi}(\mathbf{s}) = \operatorname{argmax}_{\mathbf{a} \in \mathcal{A}} \mathbb{E}(R(\mathbf{s}', \mathbf{s}, \mathbf{a}) + \gamma V_{\pi}(\mathbf{s}')). \quad (11.14)$$

Once the policy is updated, the process repeats, fixing this policy to update the value function, and then using this updated value function to improve the policy. The process is repeated until both the policy and the value function converge to within a specified tolerance. It is important to note that this procedure is both expensive and prone to finding local minima. It also resembles the alternating descent method that is widely used in optimization and machine learning.

The formulation in (11.13b) makes it clear that it may be possible to optimize backwards from a state known to give a reward with high probability. Additionally, this approach requires having a model for P and R to predict the next state \mathbf{s}' , making this a *model-based* approach.

Value iteration

Value iteration is similar to policy iteration, except that at every iteration only the value function is updated, and the optimal policy is extracted from this value function at the end. First, the value function is initialized, typically either with zeros or at random. Then, for all states $\mathbf{s} \in \mathcal{S}$, the value function is updated by returning the maximum value at that state across all actions $\mathbf{a} \in \mathcal{A}$, holding the value function fixed at all other states $\mathbf{s}' \in \mathcal{S} \setminus \mathbf{s}$:

$$V(\mathbf{s}) = \max_{\mathbf{a}} \sum_{\mathbf{s}'} P(\mathbf{s}' | \mathbf{s}, \mathbf{a}) (R(\mathbf{s}', \mathbf{s}, \mathbf{a}) + \gamma V(\mathbf{s}')). \quad (11.15)$$

This iteration is repeated until a convergence criterion is met.

After the value function converges, it is possible to extract the optimizing policy $\boldsymbol{\pi}$:

$$\boldsymbol{\pi}(\mathbf{s}, \mathbf{a}) = \operatorname{argmax}_{\mathbf{a}} \sum_{\mathbf{s}'} P(\mathbf{s}' | \mathbf{s}, \mathbf{a}) (R(\mathbf{s}', \mathbf{s}, \mathbf{a}) + \gamma V(\mathbf{s}')). \quad (11.16)$$

Although value iteration typically requires fewer steps per iteration, policy iteration often converges in fewer iterations. This may be due to the fact that the value function is often more complex than the policy function, requiring more parameters to optimize over.

Note that the value function in RL typically refers to a discounted sum of future rewards that should be maximized, while in nonlinear control it refers to an integrated cost that should be minimized. The phrase *value function* is particularly intuitive when referring to accumulated rewards in the economic sense, as it quantifies the *value* of being in a given state. However, in the case of nonlinear control theory, the *value function* is more accurately thought of as quantifying the *numerical value* of the cost function evaluated on the optimal trajectory. This notation can be confusing and is worth careful consideration depending on the context.

Quality function

Both policy iteration and value iteration rely on the quality function $Q(\mathbf{s}, \mathbf{a})$, which is defined as

$$Q(\mathbf{s}, \mathbf{a}) = \mathbb{E}(R(\mathbf{s}', \mathbf{s}, \mathbf{a}) + \gamma V(\mathbf{s}')) \quad (11.17a)$$

$$= \sum_{\mathbf{s}'} P(\mathbf{s}' | \mathbf{s}, \mathbf{a}) (R(\mathbf{s}', \mathbf{s}, \mathbf{a}) + \gamma V(\mathbf{s}')). \quad (11.17b)$$

In a sense, the optimal policy $\pi(\mathbf{s}, \mathbf{a})$ and the optimal value function $V(\mathbf{s}, \mathbf{a})$ contain redundant information, as one can be determined from the other via the quality function $Q(\mathbf{s}, \mathbf{a})$:

$$\pi(\mathbf{s}, \mathbf{a}) = \operatorname{argmax}_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}) \quad (11.18a)$$

$$V(\mathbf{s}) = \max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}). \quad (11.18b)$$

This formulation will be used for model-free Q-learning [757, 734, 244] in Section 11.3.

11.3 Model-Free Reinforcement Learning and Q-Learning

Both policy iteration and value iteration above rely on the *quality* function $Q(\mathbf{s}, \mathbf{a})$, which describes the joint desirability of a given state/action pair. Policy iteration (11.14) and value iteration (11.15) are both model-based reinforcement learning strategies, where it is assumed that the MDP model is known: each iteration requires a one-step look ahead, or model-based prediction of the next state \mathbf{s}' given the current state and action \mathbf{s} and \mathbf{a} . Based on this model, it is possible to forecast and maximize over all possible actions.

When a model is not available, there are several reinforcement learning approaches to learn effective decision and control policies to interact with the environment. Perhaps the most straightforward approach is to first learn a model of the environment using some data-driven active learning strategy, and then use the standard model-based approaches discussed earlier. However, this may be infeasible for very large or particularly unstructured systems.

Q -learning is a leading model-free alternative, which learns the Q function directly from experience, without requiring access to a model. Thus, it is possible to generalize many of the model-based optimization strategies above to more unstructured settings, where a model is unavailable. The Q function has the one-step look ahead implicitly built into its representation, without needing to explicitly refer to a model. From this learned Q function, the optimal policy and value function may be extracted as in (11.18).

Before discussing the mechanics of Q -learning in detail, it is helpful to introduce several concepts, including Monte Carlo based learning and temporal difference learning.

Monte Carlo learning

In the simplest approach to learning from experience, the value function V or quality function Q may be learned through a Monte Carlo random sampling of the state-action space through repeated evaluation of many policies. Monte Carlo approaches require that the RL task is *episodic*, meaning that the task has a defined start and terminates after a finite number of actions, resulting in a total cumulative reward at the end of the episode. Games are good examples of episodic RL tasks.

In Monte Carlo learning, the total cumulative reward at the end of the task is used to estimate either the value function V or the quality function Q by dividing the final reward equally among all of the intermediate states or state-action pairs, respectively. This is the simplest possible approach to deal with the credit assignment problem, as credit is shared equally among all intermediate steps. However, for this reason, Monte Carlo learning is typically quite sample inefficient, especially for problems with sparse rewards.

Consider the case of Monte Carlo learning of the value function. Given a new episode consisting of n steps, the cumulative discounted reward R_Σ is computed

$$R_\Sigma = \sum_{k=1}^n \gamma^k \mathbf{r}_k \quad (11.19)$$

and used to update the value function at every state s_k visited in this episode:

$$V^{\text{new}}(s_k) = V^{\text{old}}(s_k) + \frac{1}{n} (R_\Sigma - V^{\text{old}}(s_k)) \quad \forall k \in [1, \dots, n]. \quad (11.20)$$

This incremental update, weighted by $1/n$, is equivalent to waiting until the end of the episode and then updating the value function at all states along the trajectory with an equal share of the reward. Similarly, in the case of Monte Carlo learning of the Q function, the discounted reward R_Σ is used to update the Q function at every state-action pair (s_k, \mathbf{a}_k) visited in this episode:

$$Q^{\text{new}}(s_k, \mathbf{a}_k) = Q^{\text{old}}(s_k, \mathbf{a}_k) + \frac{1}{n} (R_\Sigma - Q^{\text{old}}(s_k, \mathbf{a}_k)) \quad \forall k \in [1, \dots, n]. \quad (11.21)$$

In the limit of infinite data and infinite exploration, this approach will eventually sample all possible state-action pairs and converge to the true quality function Q . However, in practice, this often amounts to an intractable brute-force search.

It is also possible to discount past experiences by introducing a learning rate $\alpha \in [0, 1]$ and using this to update the Q function:

$$Q^{\text{new}}(s_k, \mathbf{a}_k) = Q^{\text{old}}(s_k, \mathbf{a}_k) + \alpha (R_\Sigma - Q^{\text{old}}(s_k, \mathbf{a}_k)) \quad \forall k \in [1, \dots, n]. \quad (11.22)$$

Larger learning rates $\alpha > 1/n$ will favor more recent experience.

There is a question about how to initialize the many episodes required to learn with Monte Carlo. When possible, the episode will be initialized randomly at every initial state or state-action pair, providing a random sampling; however, this might not be possible for many learning tasks. Typically, Monte Carlo learning is performed *on-policy*, meaning that the optimal policy is enacted, based on the current value or quality function, and the information from this locally optimal policy is used for the update. It is also possible to promote exploration by adding a small probability of taking a random action, rather than the action dictated by the optimal policy. Finally, there are off-policy Monte Carlo methods, but in general, they are quite inefficient or unfeasible.

Temporal difference (TD) learning

Temporal different learning [694, 711, 202, 712, 105], known as TD learning, is another sample-based learning strategy. In contrast to Monte Carlo learning, TD learning is not restricted to episodic tasks, but instead learns continuously by bootstrapping based on current estimates of the value function V or quality function Q , as in dynamic programming (e.g., as in value iteration in (11.15)). TD learning is designed to mimic learning processes in animals, where time delayed rewards are often learned through environmental cues that act as secondary reinforcers preceding the delayed reward; this is most popularly understood through Pavlov's dog [563]. Thus, TD learning is typically more sample efficient than Monte Carlo learning, resulting in decreased variance, but at the cost of a bias in the learning due to the bootstrapping.

TD(0): 1-step look ahead

To understand TD learning, it is helpful to begin with the simplest algorithm: TD(0). In TD(0), the estimate of the one-step-ahead future reward is used to update the current value function.

Given a control trajectory generated through an optimal policy π , the value function at state \mathbf{s}_k is given by

$$V(\mathbf{s}_k) = \mathbb{E}(\mathbf{r}_k + \gamma V(\mathbf{s}_{k+1})). \quad (11.23)$$

Thus, in the language of Bayesian statistics, $\mathbf{r}_k + \gamma V(\mathbf{s}_{k+1})$ is an *unbiased estimator* for $V(\mathbf{s}_k)$.

For non-optimal policies π , this same idea may be used to update the value function based on the value function one step in the future:

$$V^{\text{new}}(\mathbf{s}_k) = V^{\text{old}}(\mathbf{s}_k) + \alpha \overbrace{\left(\underbrace{\mathbf{r}_k + \gamma V^{\text{old}}(\mathbf{s}_{k+1})}_{\text{TD target estimates } R_\Sigma} - V^{\text{old}}(\mathbf{s}_k) \right)}^{\text{TD error}}. \quad (11.24)$$

Instead of using a model to predict \mathbf{s}_{k+1} , which is required to evaluate $V(\mathbf{s}_{k+1})$, it is possible to wait until the next step is actually taken and retroactively adjust the value function. Notice that this is very similar to optimization of the Bellman equation using dynamic programming but with retroactive updates based on sampled data rather than proactive updates based on a model prediction.

In the TD(0) update above, the expression $R_\Sigma = \mathbf{r}_k + \gamma V(\mathbf{s}_{k+1})$ is known as the *TD target*, as it is the estimate for the future reward, analogous to R_Σ in Monte Carlo learning of the Q function in (11.22). The difference between this target and the previous estimate of the value function is the TD error, and it is used to update the value function, just as in Monte Carlo learning, with a learning rate α .

TD(n): n-step look ahead

Other temporal difference algorithms can be developed, based on multi-step look-aheads into the future. For example, TD(1) uses a TD target based on two steps into the future

$$\mathbf{r}_k + \gamma \mathbf{r}_{k+1} + \gamma^2 V(\mathbf{s}_{k+2}) \quad (11.25)$$

and, TD(n) uses a TD target based on $n + 1$ steps into the future

$$R_\Sigma^{(n)} = \mathbf{r}_k + \gamma \mathbf{r}_{k+1} + \gamma^2 \mathbf{r}_{k+2} + \cdots + \gamma^n \mathbf{r}_{k+n} + \gamma^{n+1} V(\mathbf{s}_{k+n+1}) \quad (11.26a)$$

$$= \sum_{j=0}^n \gamma^j \mathbf{r}_{k+j} + \gamma^{n+1} V(\mathbf{s}_{k+n+1}). \quad (11.26b)$$

Again, there does not need to be a model for these future states, but instead, the value function may be retroactively adjusted based on the actual sampled trajectory and rewards. Note that in the limit that an entire episode is used, TD(n) converges to the Monte Carlo learning approach.

TD- λ : Weighted look ahead

An important variant of the TD learning family is TD- λ , which was introduced by Sutton [694]. TD- λ creates a TD target R_Σ^λ that is a weighted average of the various TD(n) targets $R_\Sigma^{(n)}$. The weighting is given by:

$$R_\Sigma^\lambda = (1 - \lambda) \sum_{k=1}^{\infty} \lambda^{k-1} R_\Sigma^{(k)} \quad (11.27)$$

and the update equation is

$$V^{\text{new}}(\mathbf{s}_k) = V^{\text{old}}(\mathbf{s}_k) + \alpha (R_\Sigma^\lambda - V^{\text{old}}(\mathbf{s}_k)). \quad (11.28)$$

TD- λ was used for an impressive demonstration in the game of Backgammon by Tesauro in 1995 [712].

TD learning provides one of the strongest connections between reinforcement learning and learning in biological systems. These neural circuits are believed to estimate the future reward, and feedback is based on the difference between the expected reward and the actual reward, which is closely related to the TD error. In fact, there are specific neurotransmitter feedback loops that strengthen connections based on proximity of their firing to a dopamine reward signal [657]. The closer the proximity in time between an action and a reward, the stronger the feedback.

Bias-variance tradeoff

Monte Carlo learning and TD learning exemplify the *bias-variance tradeoff* in machine learning. Monte Carlo learning typically has high variance but no bias, while TD learning has lower variance but introduces a bias because of the bootstrapping. Although the *true* TD target $r_k + \gamma V(\mathbf{s}_{k+1})$ is an unbiased estimate of $V(\mathbf{s}_k)$ for an optimal policy π , the sampled TD target is a biased estimate, because it uses sub-optimal actions and the current imperfect estimate of the value function.

SARSA: State–action–reward–state–action learning

SARSA is a popular TD algorithm that is used to learn the Q function *on-policy*. The Q update equation in SARSA(0) is nearly identical to the V update equation

(11.24) in TD(0):

$$Q^{\text{new}}(\mathbf{s}_k, \mathbf{a}_k) = Q^{\text{old}}(\mathbf{s}_k, \mathbf{a}_k) + \alpha \left(\mathbf{r}_k + \gamma Q^{\text{old}}(\mathbf{s}_{k+1}, \mathbf{a}_{k+1}) - Q^{\text{old}}(\mathbf{s}_k, \mathbf{a}_k) \right). \quad (11.29)$$

There are SARSA variants for all of the TD(n) algorithms, based on the n step TD target:

$$R_{\Sigma}^{(n)} = \mathbf{r}_k + \gamma \mathbf{r}_{k+1} + \gamma^2 \mathbf{r}_{k+2} + \cdots + \gamma^n \mathbf{r}_{k+n} + \gamma^{n+1} Q(\mathbf{s}_{k+n+1}, \mathbf{a}_{k+n+1}) \quad (11.30a)$$

$$= \sum_{j=0}^n \gamma^j \mathbf{r}_{k+j} + \gamma^{n+1} Q(\mathbf{s}_{k+n+1}, \mathbf{a}_{k+n+1}). \quad (11.30b)$$

In this case, the SARSA(n) update equation is given by

$$Q^{\text{new}}(\mathbf{s}_k, \mathbf{a}_k) = Q^{\text{old}}(\mathbf{s}_k, \mathbf{a}_k) + \alpha \left(R_{\Sigma}^{(n)} - Q^{\text{old}}(\mathbf{s}_k, \mathbf{a}_k) \right). \quad (11.31)$$

Note that this is on-policy because the actual action sequence $\mathbf{a}_k, \mathbf{a}_{k+1}, \dots, \mathbf{a}_{k+n+1}$ has been used to receive the rewards \mathbf{r} and evaluate the $n + 1$ step Q function $Q(\mathbf{s}_{k+n+1}, \mathbf{a}_{k+n+1})$.

Q-Learning

We are now ready to discuss Q -learning [757, 734, 244], which is one of the most central approaches in model-free RL. Q -learning is essentially an off-policy TD learning scheme for the Q function. In Q -learning, the Q update equation is

$$Q^{\text{new}}(\mathbf{s}_k, \mathbf{a}_k) = Q^{\text{old}}(\mathbf{s}_k, \mathbf{a}_k) + \alpha \left(\mathbf{r}_k + \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{k+1}, \mathbf{a}) - Q^{\text{old}}(\mathbf{s}_k, \mathbf{a}_k) \right). \quad (11.32)$$

Notice that the only difference between Q -learning and SARSA(0) is that SARSA(0) uses $Q(\mathbf{s}_{k+1}, \mathbf{a}_{k+1})$ for the TD target, while Q -learning uses $\max_{\mathbf{a}} Q(\mathbf{s}_{k+1}, \mathbf{a})$ for the TD target. Thus, SARSA(0) is considered *on-policy* because it uses the action \mathbf{a}_{k+1} based on the actual policy: $\mathbf{a}_{k+1} = \pi(\mathbf{s}_{k+1})$. In contrast, Q -learning is *off-policy* because it uses the optimal \mathbf{a} for the update based on the current estimate for Q , while taking a different action \mathbf{a}_{k+1} based on a different behavior policy. Thus, Q -learning may take sub-optimal actions \mathbf{a}_{k+1} to explore, while still using the optimal action \mathbf{a} to update the Q function.

Generally, Q -learning will learn a more optimal solution faster than SARSA, but with more variance in the solution. However, SARSA will typically yield more cumulative rewards during the training process, since it is on-policy. In safety critical applications, such as self-driving cars or other applications where there can be catastrophic failure, SARSA will typically learn less optimal solutions, but with a better safety margin, since it is maximizing on-policy rewards.

Q -learning applies to discrete action spaces \mathcal{A} and state spaces \mathcal{S} governed by a finite MDP. The Q function is classically represented as a table of Q values that is updated through some iteration based on new information as a policy is tested and evaluated. However, this tabular approach doesn't scale well to large state spaces, and so typically function approximation is used to represent the Q function, such as a neural network in deep Q -learning. Even if the action and state spaces are continuous, as in the pendulum on a cart system, it is possible to discretize and then apply Q -learning. In addition to being model free, Q -learning is also referred to as *off-policy* RL, as it does not require that an optimal policy is enacted, as in policy iteration and value iteration. Off-policy learning is more realistic in real-world applications, enabling the RL agent to improve when its policy is sub-optimal and by watching and imitating other more skilled agents. Q -learning is especially good for games, such as backgammon, chess, and go. In particular, deep Q -learning, which approximates the Q function using a deep neural network, has been used to surpass the world champions in these challenging games.

Experience replay and imitation learning

Because Q -learning is off-policy, it is possible to learn from action-state sequences that do not use the current optimal policy. For example, it is possible to store past experiences, such as previously played games, and *replay* these experiences to further improve the Q function.

In an on-policy strategy, such as SARSA, using actions that are sub-optimal, based on the current optimal policy, will degrade the Q function, since the TD target will be a flawed estimate of future rewards based on a sub-optimal action. However, in Q -learning, since the action is optimized over the current Q function in the update, it is possible to learn from experience resulting from sub-optimal actions. This also makes it possible to learn from watching other, more experienced agents, which is related to imitation learning [637, 337, 351, 222].

Experience replay is deeply intuitive, as it is closely related to how we learn, through recalling past experiences in the light of new knowledge (i.e., an updated Q function). Similarly, imitation learning is perhaps one of the most fundamental first steps in biological learning.

Exploration vs exploitation: ϵ -greedy actions

It is important to introduce an element of random exploration into Q -learning, and there are several techniques. One approach is the ϵ -greedy algorithm to select the next action. In this approach, the agent takes the current optimal action $\mathbf{a}_k = \max_{\mathbf{a}} Q(\mathbf{s}_k, \mathbf{a})$, based on the current Q function, with probability $1 - \epsilon$, where $\epsilon \in [0, 1]$. With probability ϵ , the agent takes a random action.

Thus, the agent balances exploration with the random actions and exploitation with the optimal actions. Larger ϵ promote more random exploration.

Typically, the value of ϵ will be initialized to a large value, often $\epsilon = 1$. Throughout the course of training, ϵ decays so that as the Q function improves, the agent increasingly takes the current optimal action. This is closely related to simulated annealing from optimization, which mimics the process of metal finding a low-energy state through a specific cooling schedule.

Policy Gradient Optimization

Policy gradients [696, 377, 672] are one of the most common and powerful techniques to optimize a policy that is parameterized, as in (11.2). When the policy π is parameterized by θ , it is possible to use gradient optimization on the parameters to improve the policy much faster than through traditional iteration. The parameterization may be a multi-layer neural network, in which case this would be a *deep policy network*, although other representations and function approximations may be useful. In any case, instead of extracting the policy as the argument maximizing the value or quality functions, it is possible to directly optimize the parameters θ , for example through gradient descent or stochastic gradient descent. The value function $V_\pi(s)$, depending on a policy π then becomes $V(s, \theta)$ and a similar modification is possible for the quality function Q .

The total estimated reward is given by

$$R_{\Sigma, \theta} = \sum_{s \in \mathcal{S}} \mu_\theta(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) Q(s, a), \quad (11.33)$$

where μ_θ is the asymptotic steady state of the MDP given a policy π_θ parameterized by θ . It is then possible to compute the gradient of the total estimated reward with respect to θ

$$\nabla_\theta R_{\Sigma, \theta} = \sum_{s \in \mathcal{S}} \mu_\theta(s) \sum_{a \in \mathcal{A}} Q(s, a) \nabla_\theta \pi_\theta(s, a) \quad (11.34a)$$

$$= \sum_{s \in \mathcal{S}} \mu_\theta(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) Q(s, a) \frac{\nabla_\theta \pi_\theta(s, a)}{\pi_\theta(s, a)} \quad (11.34b)$$

$$= \sum_{s \in \mathcal{S}} \mu_\theta(s) \sum_{a \in \mathcal{A}} \pi_\theta(s, a) Q(s, a) \nabla_\theta \log(\pi_\theta(s, a)) \quad (11.34c)$$

$$= \mathbb{E}(Q(s, a) \nabla_\theta \log(\pi_\theta(s, a))). \quad (11.34d)$$

Then the policy parameters may be updated as

$$\theta^{\text{new}} = \theta^{\text{old}} + \alpha \nabla_\theta R_{\Sigma, \theta}, \quad (11.35)$$

where α is the learning weight; note that α may be replaced with a vector of learning weights for each component of θ . There are several approaches to approximating this gradient, including through finite differences, the REINFORCE algorithm [770], and natural policy gradients [377].

11.4 Deep Reinforcement Learning

Deep reinforcement learning is one of the most exciting areas of machine learning and of control theory, and it is one of the most promising avenues of research towards generalized artificial intelligence. Deep learning has revolutionized our ability to represent complex functions from data, providing a set of architectures for achieving human level performance in complex tasks such as image recognition and natural language processing. Classic reinforcement learning suffers from a representation problem, as many of the relevant functions, such as the policy π , the value function V , and the quality function Q , may be exceedingly complex functions defined over a very high dimensional state and action space. Indeed, even for simple games, such as the 1972 Atari game Pong, the black and white screen at standard resolution 336×240 has over $10^{24,000}$ possible discrete states, making it infeasible to represent any of these functions exactly without approximation. Thus, deep learning provides a powerful tool for improving these representations.

It is possible to use deep learning in several different ways to approximate the various functions used in RL, or to model the environment more generally. Typically the central challenge is in identifying and representing key features in a high-dimensional state space. For example, the policy $\pi(a, s)$ may now be approximated by

$$\pi(s, a) \approx \pi(s, a, \theta), \quad (11.36)$$

where θ represent the weights of a neural network.

This pairing of deep learning for representations with reinforcement learning for decision making and control has resulted in dramatic improvements to our capabilities of reinforcement learning. For example, Fig. 11.4 shows a simple policy network designed to play Pong, and Fig. 11.5 shows a more general deep convolutional neural network architecture used to develop a deep Q network to play Atari games [519].

Much of what is discussed in this section is also relevant for other function approximation techniques besides deep learning. For example, policy gradients may be computed and used for gradient-based optimization using other representations, and there is a long history before deep learning [696, 377]. That said, many of the most exciting and impressive recent demonstrations of RL leverage the full power of deep learning, and so we present these innovations in this context.

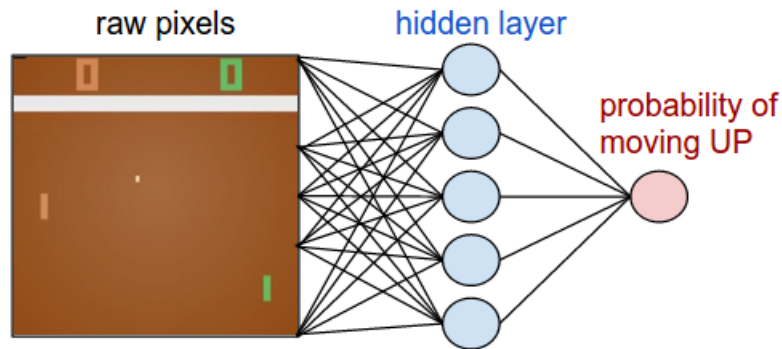


Figure 11.4: Deep policy network to encode the probability of moving up in the game of Pong. Reproduced with permission from Andrej Karpathy's Blog "Deep Reinforcement Learning: Pong from Pixels" at <http://karpathy.github.io/2016/05/31/r1/>.

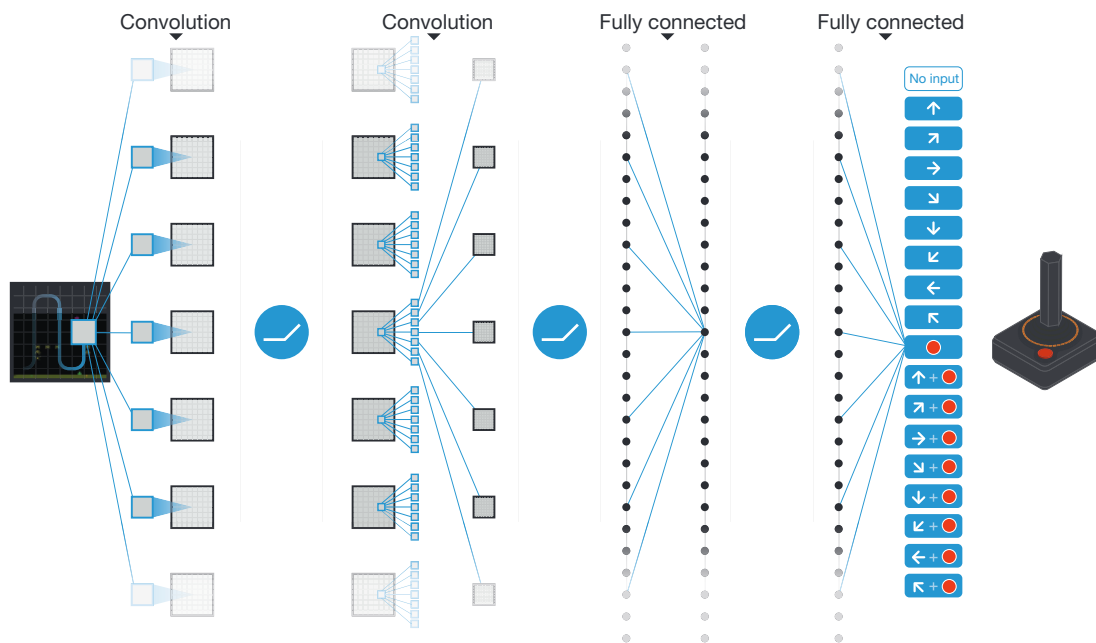


Figure 11.5: Convolutional structure of deep Q network used to play Atari games. Reproduced with permission from [519].

Deep Q-learning

Many of the most exciting advances in the past decade have involved some variation of deep Q -learning, which uses deep neural networks to represent the quality function Q . As with the policy in (11.36), it is possible to approximate

the Q function through some parameterization θ

$$Q(\mathbf{s}, \mathbf{a}) \approx Q(\mathbf{s}, \mathbf{a}, \theta), \quad (11.37)$$

where θ represents the weights of a deep neural network. In this representation, the training loss function is directly related to the standard Q -learning update in (11.32):

$$\mathcal{L} = \mathbb{E} \left[\left(\mathbf{r}_k + \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{k+1}, \mathbf{a}_{k+1}, \theta) - Q(\mathbf{s}_k, \mathbf{a}_k, \theta) \right)^2 \right]. \quad (11.38)$$

The first part of the loss function, $\mathbf{r}_k + \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{k+1}, \mathbf{a}_{k+1}, \theta)$, is the temporal difference target from before, and the second part, $Q(\mathbf{s}_k, \mathbf{a}_k, \theta)$, is the prediction.

Deep reinforcement learning based on a deep Q network (DQN) was introduced by Mnih et al. [519] to play Atari games. Specifically, this network used a deep convolutional neural network to represent the Q function, where the inputs were the Atari screen, as shown in Fig. 11.5. In this original paper, both the Q functions in (11.38) were represented by the same network weights θ . However, in a double DQN [742], different networks are used to represent the target and prediction Q functions, which reduces bias due to inaccuracies early in training. In double DQN, it may be necessary to fix the target network for multiple training iterations of the prediction network before updating to improve stability and convergence [264].

Experience replay is a critical component of training a DQN, which is possible because it is an off-policy RL algorithm. Short segments of past experiences are used in batches for the stochastic gradient descent during training. Moreover, to place more importance on experiences with large model mismatch, it is possible to weight past experiences by the magnitude of the TD error. This process is known as prioritized experience replay [642].

Dueling deep Q networks (DDQNs) [756] are another important deep Q learning architecture that are used to improve training when actions have a marginal affect on the quality function. In particular, a DDQN splits the quality function into the sum of a value function and an *advantage* function $A(\mathbf{s}, \mathbf{a})$, which quantifies the additional benefit of a particular action over the value of being in that state:

$$Q(\mathbf{s}, \mathbf{a}, \theta) = V(\mathbf{s}, \theta_1) + A(\mathbf{s}, \mathbf{a}, \theta_2). \quad (11.39)$$

The value and advantage networks have separate networks that are combined to estimate the Q function.

There are a variety of other useful architectures for deep Q learning, with more introduced regularly. For example, deep recurrent Q networks are promising for dynamic problems [323]. Advantage actor-critic networks, discussed in the next section, combine the DDQN with deep policy networks.

Actor-critic networks

Actor-critic methods in reinforcement learning simultaneously learn a policy function and a value function, with the goal of taking the best of both value-based and policy-based learning. The basic idea is to have an actor, which is policy-based, and a critic, which is value-based, and to use the temporal difference signal from the critic to update the policy parameters. There are many actor-critic methods that predate deep learning. For example, a simple actor-critic approach would update the policy parameters θ in (11.36) using the temporal difference error $r_k + \gamma V(\mathbf{s}_{k+1}) - V(\mathbf{s}_k)$:

$$\theta_{k+1} = \theta_k + \alpha (r_k + \gamma V(\mathbf{s}_{k+1}) - V(\mathbf{s}_k)). \quad (11.40)$$

It is rather straightforward to incorporate deep learning into an actor-critic framework. For example, in the advantage actor critic (A2C) network, the actor is a deep policy network, and the critic is a DDQN. In this case, the update is given by

$$\theta_{k+1} = \theta_k + \alpha \nabla_{\theta} ((\log \pi(\mathbf{s}_k, \mathbf{a}_k, \theta)) Q(\mathbf{s}_k, \mathbf{a}_k, \theta_2)). \quad (11.41)$$

Challenges and Additional Techniques

There are several important innovations that are necessary to make reinforcement learning tractable for even moderately challenging tasks. Two of the biggest challenges in RL are: 1) high-dimensional state and action spaces, and 2) sparse and delayed rewards.

Many games, such as chess and go, have exceedingly large state spaces. For example, Claude Shannon estimated the number of games of chess, known as the Shannon number, at around 10^{120} in his famous paper “Programming a computer for playing chess” [666]; this paper was a major inspiration for modern dynamic programming and reinforcement learning. Representing a value or quality function, let alone sampling over these states, is beyond astronomically difficult. Thus, approximate representations of the value or quality functions using approximation theory, such as deep neural networks, are necessary.

Sparse and delayed rewards represent the central challenge of reinforcement learning, leading to the well-known credit assignment problem, which we have seen multiple times at this point. The following techniques, including reward shaping and hindsight experience replay, are leading techniques to overcome the credit assignment problem.

Reward shaping

Perhaps the most standard approach for systems with sparse rewards is a technique called reward shaping. This involves designing customized proxy fea-

tures that are indicative of a future reward and that may be used as an intermediate reward signal. For example, in the game of chess, the relative point count, where each piece is assigned a numeric value and added up (e.g., a queen is worth 10 points, rooks are worth 5, knights and bishops are worth 3, and pawns are worth 1 point), is an example of a shaped reward that gives an intermediate reward signal each time a piece is taken.

Reward shaping is quite common and can be very effective. However, these rewards require expert human guidance to design, and this requires customized effort for each new task. Thus, reward shaping is not a viable strategy for a generalized artificial intelligence agent capable of learning multiple games or tasks. In addition, reward shaping generally limits the upper end of the agent's performance to that of the human expert.

Hindsight experience replay

In many tasks, such as robotic manipulation, the goal is to move the robot or an object from one location to another. For example, consider a robot arm that is required to slide an object on a table from point *A* to point *B*. Without a detailed physical model, or other prior knowledge, it is extremely unlikely that a random control policy will result in the object actually reaching the desired destination, so the rewards may be very sparse. It is possible to shape a reward based on the distance of the object to the goal state, although this is not a general strategy and suffers from the limitations discussed above.

Hindsight experience replay (HER) [22, 438] is a strategy that enriches the reward signal by taking failed trials and pretending that they were successful at a different task. This approach makes the reward structure much more dense, and has the benefit of enabling the simultaneous learning of a whole family of motion tasks.

HER is quite intuitive in the context of human learning, for example in the case of tennis. Initially, it is difficult to aim the ball, shots often go wild when learning. However, this provides valuable information about those muscle actions, which might be useful for future tasks. After lots of practice, it then becomes possible to pick from different shots and place the ball more deliberately.

Curiosity driven exploration

Another challenge with RL for large open-world environments is that the agent may easily get stuck in a local minima, where it over-optimizes for a small region of state space. One approach to this problem is to augment the reward signal with a *novelty* reward that is large in regions of state space that are not well modeled. This is known as curiosity driven exploration [562], and it involves an intrinsic curiosity module (ICM), which compares a forward model

of the evolution of the state, or a latent representation of the state, with the actual observed evolution. The discrepancy between the model and the actual dynamics is the novelty reward. When this difference is large, the agent becomes *curious* and explores this region more. There are similarities between this approach and TD learning, and in fact, many of the same variations may be implemented for curiosity driven exploration. The main difference is that in TD learning, the reward discrepancy is used as feedback to improve the value or quality function, while in curiosity driven exploration the discrepancy is explicitly used as an additional reward signal. This is a clever approach to embedding this fundamental behavior of intelligent biological learning systems, to be curious and explore.

There are challenges when using this novelty reward for chaotic and stochastically driven systems, where there are aspects of the state evolution that are fundamentally unpredictable. A naive novelty reward would constantly provide positive incentive to explore these regions, since the forward model will not improve. Instead, the authors in [562] overcome this challenge by predicating novelty on the predictability of an outcome given the action using latent features in an autoencoder, so only aspects of the future state that can be affected by the agent's actions are included in the novelty signal.

11.5 Applications and Environments

Here we provide a brief overview of some of the modern applications and success stories of RL, along with some common environments.

OpenAI Gym

The OpenAI Gym is an incredible open source resource to develop and test reinforcement learning algorithms in a wide range of environments. Fig. 11.6 shows a small selection of these systems. Example environments include

- Classic Atari video games: over 100 tasks on Atari 2600 games, including asteroids, breakout, space invaders, and many others.
- Classic control benchmarks: tasks include balancing an inverted pendulum on a cart; swing-up of a pendulum; swing-up of a double pendulum; and driving up a hill with an underactuated system.
- Goal-based robotics [780]: tasks include pushing or fetching a block to a goal position with a robot arm, with and without sliding after loss of contact; robotic hand manipulation for reaching a pose or orienting various objects.

- MuJoCo [718]: tasks include multi-legged locomotion, running, hopping, swimming, etc. within a fast physics simulator environment.

This wide range of environments and tasks provides an invaluable resource for RL researchers, dramatically lowering the barrier to entry and facilitating the benchmarking and comparison of innovations.


Classic board games

As discussed throughout this chapter, RL has developed tremendously over the past half century, from a biologically inspired idea to a major field pushing the forefront of efforts in generalized artificial intelligence. This progress can be largely traced through the success of RL on increasingly challenging games, where RL has learned to interact with and mimic humans, and eventually to defeat our greatest grandmasters.

Many of the most fundamental advances in RL were either developed for the purpose of playing games, or demonstrated on the most challenging games of the time. These simple board games also make the struggles of machine learning and artificial intelligence more relatable to humans², as we can reflect on our own experiences learning first how to play tic-tac-toe, then checkers, and then eventually “real” games, such as backgammon, chess, and go. The progression of RL capabilities roughly follows this progression of complexity, with tic-tac-toe being essentially a homework exercise, checkers being the earliest real demonstration of RL by Arthur Samuel [628], and more complex games such as backgammon [712] and eventually chess and go [670, 673] following. Interestingly, about three decades passed between each of these definitive landmarks. One of the next major landmarks is a generalist RL agent that can learn to play multiple games [671], rather than specializing in only one task.

The success of DeepMind’s AlphaGo and AlphaGo Zero, depicted in Fig. 11.7, demonstrates the remarkable power of modern RL. This system was a major breakthrough in RL research, learning to beat the Grandmaster Lee Sedol 4-1 in 2016. However, AlphaGo relied heavily on reward shaping and expert guidance, making it a custom solution, rather than a generalized learner. Its successor, AlphaGo Zero, relied entirely on self-play, and was able to eventually defeat the original AlphaGo decisively. AlphaGo was based largely on CNNs, while AlphaGo Zero used a residual network (ResNet). ResNets are easier to train, and AlphaGo Zero was one of the first concrete success stories that cemented ResNets as a competitive architecture. AlphaGo Zero was trained in 40 days on 4 tensor processing units, in contrast to many advanced ML algorithms that are trained for months on thousands of GPUs. Both AlphaGo and

²“A strange game. The only winning move is not to play. How about a nice game of chess?”
– WarGames, 1983


Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

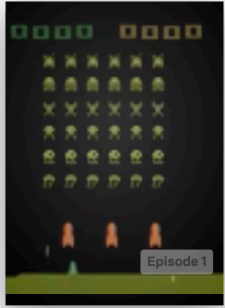

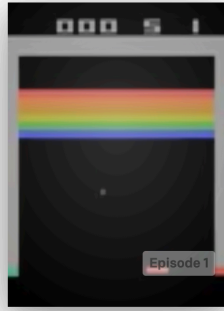
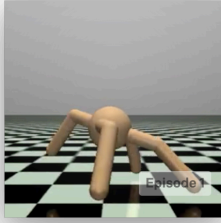
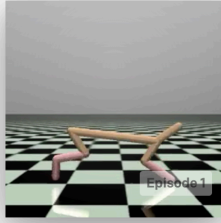
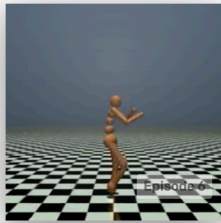
Algorithms	Atari Reach high scores in Atari 2600 games.		
Atari			
Box2D	<div style="display: flex; justify-content: space-around;"> <div style="width: 30%; text-align: center;"> <p>SpaceInvaders-ram-v0 Maximize score in the game SpaceInvaders, with RAM as input</p> </div> <div style="width: 30%; text-align: center;"> <p>Qbert-v0 Maximize score in the game Qbert, with screen images as input</p> </div> <div style="width: 30%; text-align: center;"> <p>Breakout-ram-v0 Maximize score in the game Breakout, with RAM as input</p> </div> </div>		
Classic control	MuJoCo Continuous control tasks, running in a fast physics simulator.		
MuJoCo			
Robotics	Robotics Simulated goal-based tasks for the Fetch and ShadowHand robots.		
Toy text EASY	<div style="display: flex; justify-content: space-around;"> <div style="width: 30%; text-align: center;"> <p>Ant-v2 Make a 3D four-legged robot walk.</p> </div> <div style="width: 30%; text-align: center;"> <p>HalfCheetah-v2 Make a 2D cheetah robot run.</p> </div> <div style="width: 30%; text-align: center;"> <p>Humanoid-v2 Make a 3D two-legged robot walk.</p> </div> </div>		
Third party environments ↗	<div style="display: flex; justify-content: space-around;"> <div style="width: 30%; text-align: center;"> <p>FetchPickAndPlace-v1 Lift a block into the air.</p> </div> <div style="width: 30%; text-align: center;"> <p>FetchPush-v1 Push a block to a goal position.</p> </div> <div style="width: 30%; text-align: center;"> <p>FetchReach-v1 Move Fetch to a goal position.</p> </div> </div>		

Figure 11.6: The OpenAI Gym [121] (gym.openai.com) provides a flexible simulation environment to test learning strategies. Examples include classic Atari 2600 video games and simulated rule-based control environments, including open world physics [718], and robotics [780]. Other examples include classic control benchmarks.



Figure 11.7: Reinforcement learning has demonstrated incredible performance in recent expert tasks, such as AlphaGo defeating world champion Lee Sedol in the game of Go [671] on March 19, 2016.

AlphaGo Zero are based on using deep learning to improve a Monte Carlo tree search.

Video games

Some of the most impressive recent innovations in RL have involved scaling up to larger input spaces, which are well-exemplified by the ability of RL to master classic Atari video games [519]. In the case of Atari games, the pixel space is processed using a CNN architecture, with human-level performance being achieved mere years after the birth of modern deep learning for image classification [423]. More recently, RL has been demonstrated on more sophisticated games, such as StarCraft [747], which is a real-time strategy game; DeepMind's AlphaStar became a Grandmaster in 2019.

General artificial intelligence is one of the grand challenge problems in modern machine learning, whereby a learning agent is able to excel at multiple tasks, as in biological systems. What is perhaps most impressive about recent RL agents that learn video games is that the learning approach is *general*, so that the same RL framework can be used to learn multiple tasks. There is evidence that video games may improve performance in human surgeons [619, 478], and it may be that future RL agents will master both robotic manipulation and video games in a next stage of generalized AI.

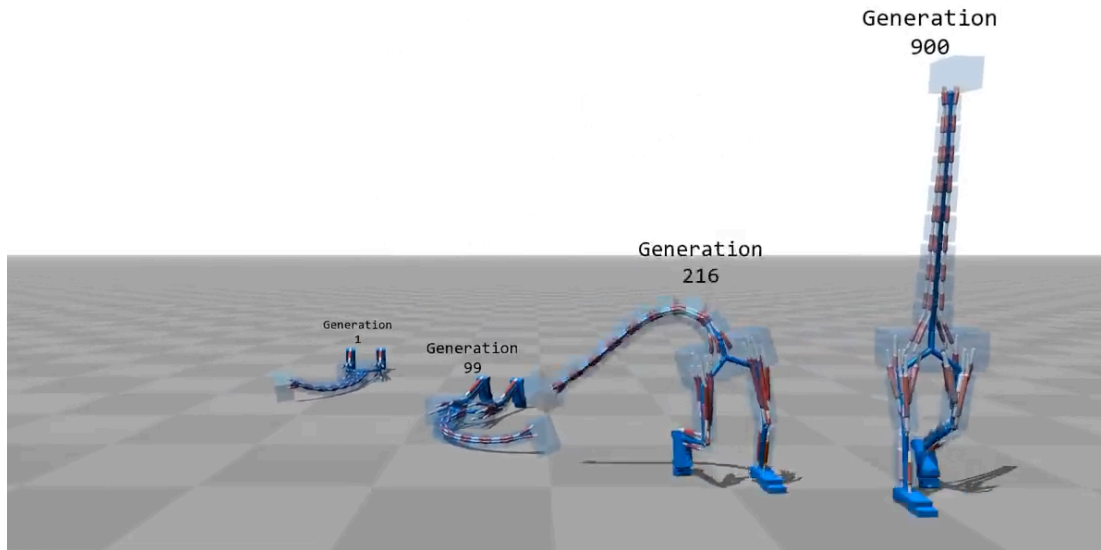


Figure 11.8: Illustration of improved bipedal locomotion performance with more generations of learning. *Reproduced from Geijtenbeek et al. [277].*

Physical systems

Although much of RL has been developed for board games and video games, it is increasingly being used for various advanced modeling and control tasks in physical systems. Physical systems, such as lasers [690] and fluids [595], often require additional considerations, such as continuous state and action spaces [601], and the need for certifiable solutions, such as trust regions [656], for safety critical applications (e.g., transportation, autonomous flight, etc.).

There has been considerable work applying RL in the field of fluid dynamics [132] for fluid flow control [308, 577, 594, 595], for example for bluff body control [247] and controlling Rayleigh-Bénard convection [67]. RL has also been applied to the related problem of navigation in a fluid environment [184, 86, 310], and more recently for turbulence modeling [543].

In addition to studying fluids, there is an extensive literature using RL to develop control policies for real and simulated robotic systems that operate primarily in a fluid environment, for example to learn how to fly and swim. For example, some of the earliest work has involved optimizing the flight of uninhabited aerial vehicles [395, 2, 710, 1, 785, 551, 603] with especially impressive helicopter aerobatics [2]. Controlling the motion of fish [274, 275, 545, 745] is another major area of development, including individual [274] and collective motion [275, 545, 745]. Gliding and perching is another large area of development [602, 603, 544].

Robotics and Autonomy

Robotics [404, 306] and autonomy [664, 627, 558, 604] are two of the largest areas of current research in RL. These both count as *physical systems*, as in the section above, but deserve their own treatment, as these are major areas of innovation. In fact, both robotics and autonomy may be viewed as two of the most pressing societal applications of machine learning in general, and reinforcement learning in particular, with self driving cars alone promising to remake the modern transportation and energy landscape. As with the discussion of physical systems above, these are typically safety critical applications with physical constraints [440, 708]. Figure 11.8 shows a virtual locomotion task that involves learning physics in a robot walker.

11.6 Optimal Nonlinear Control

Reinforcement learning has considerable overlap with optimal nonlinear control, and historically they were developed in parallel under the same optimization framework. Here we provide a brief overview of optimal nonlinear control theory, which will provide a connection between the classic linear control theory from Chapter 8 and dynamic programming to solve Bellman's equations used in this chapter. We have already seen optimal control in context of linear dynamics and quadratic cost functions in Section 8.4, resulting in the linear quadratic regulator (LQR). Similarly, we have used Bellman's equations to find optimal policies in RL for systems governed by MDPs. A major goal of this section is to provide a more general mathematical treatment of Bellman's equations, extending these approaches to fully nonlinear optimal control problems. However, this section is very technical and departs from the MDP notation used throughout the rest of the chapter; it may be omitted on a first reading. For more details, see the excellent text by Stengel [687].

Hamilton-Jacobi-Bellman equation

In optimal control, the goal is often to find a control input $\mathbf{u}(t)$ to drive a dynamical system

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) dt \quad (11.42)$$

to follow a trajectory $\mathbf{x}(t)$ that minimizes a cost function

$$J(\mathbf{x}(t), \mathbf{u}(t), t_0, t_f) = Q(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau. \quad (11.43)$$

Note that this formulation in (11.43) generalizes the LQR cost function in (8.47); now the immediate cost function $\mathcal{L}(\mathbf{x}, \mathbf{u})$ and the terminal cost $Q(\mathbf{x}(t_f), t_f)$ may be non-quadratic functions. Often there are also constraints on the state \mathbf{x} and control \mathbf{u} , which determine what solutions are admissible.

Given an initial state $\mathbf{x}_0 = \mathbf{x}(t_0)$ at t_0 , an optimal control $\mathbf{u}(t)$ will result in an optimal cost function J . We may define a *value* function $V(\mathbf{x}, t_0, t_f)$ that describes the total integrated cost starting at this position \mathbf{x} assuming the control law is optimal:

$$V(\mathbf{x}(t_0), t_0, t_f) = \min_{\mathbf{u}(t)} J(\mathbf{x}(t), \mathbf{u}(t), t_0, t_f), \quad (11.44)$$

where $\mathbf{x}(t)$ is the solution to (11.42) for the optimal $\mathbf{u}(t)$. Notice that the value function is no longer a function of the control $\mathbf{u}(t)$, as this has been optimized over, and it is also not a function of a trajectory $\mathbf{x}(t)$, but rather of an initial state \mathbf{x}_0 , as the remainder of the trajectory is entirely specified by the dynamics and the optimal control law. The value function is often called the *cost-to-go* in control theory, as the value function evaluated at any point $\mathbf{x}(t)$ on an optimal trajectory will represent the remaining cost associated with continuing to enact this optimal policy until the final time t_f . In fact, this is a statement of Bellman's optimality principle, that the value function V remains optimal starting with any point on an optimal trajectory.

The Hamilton-Jacobi-Bellman³ (HJB) equation establishes a partial differential equation that must be satisfied by the value function $V(\mathbf{x}(t), t, t_f)$ at every intermediate time $t \in [t_0, t_f]$:

$$-\frac{\partial V}{\partial t} = \min_{\mathbf{u}(t)} \left(\left(\frac{\partial V}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t)) \right). \quad (11.45)$$

To derive the HJB equation, we may compute the total time derivative of

³Kalman recognized that the Bellman optimal control formulation was a generalization of the Hamilton-Jacobi equation from classical mechanics to handle stochastic input-output systems. These formulations all involve the calculus of variations, which traces its roots back to the Brachistichrone problem of Johann Bernoulli.

the value function $V(\mathbf{x}(t), t, t_f)$ at some intermediate time t :

$$\frac{d}{dt}V(\mathbf{x}(t), t, t_f) = \frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial \mathbf{x}}\right)^T \frac{d\mathbf{x}}{dt} \quad (11.46a)$$

$$= \min_{\mathbf{u}(t)} \frac{d}{dt} \left(\int_0^{t_f} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau + Q(\mathbf{x}(t_f), t_f) \right) \quad (11.46b)$$

$$= \min_{\mathbf{u}(t)} \left(\underbrace{\frac{d}{dt} \int_0^{t_f} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau}_{-\mathcal{L}(\mathbf{x}(t), \mathbf{u}(t))} \right) \quad (11.46c)$$

$$\implies -\frac{\partial V}{\partial t} = \min_{\mathbf{u}(t)} \left(\left(\frac{\partial V}{\partial \mathbf{x}}\right)^T \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathcal{L}(\mathbf{x}, \mathbf{u}) \right). \quad (11.46d)$$

Note that the terminal cost does not vary with t , so it has zero time derivative. The derivative of the integral of the instantaneous cost $\int_t^{t_f} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau$ is equal to $-\mathcal{L}(\mathbf{x}(t), \mathbf{u}(t))$ by the first fundamental theorem of calculus. Finally, the term $(\partial V/\partial \mathbf{x})^T \mathbf{f}(\mathbf{x}, \mathbf{u})$ may be brought into the minimization argument, since V is already defined as the optimal cost over \mathbf{u} . The LQR optimal Riccati equation is a special case of the HJB equation, and the vector of partial derivatives in $(\partial J/\partial \mathbf{x})$ serves the same role of the Lagrange multiplier co-state $\boldsymbol{\lambda}$. The HJB equation may also be more intuitive in vector calculus notation

$$-\frac{\partial V}{\partial t} = \min_{\mathbf{u}(t)} (\nabla V \cdot \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t))). \quad (11.47)$$

The HJB formulation above relies implicitly on Bellman's principle of optimality, that for any point on an optimal trajectory $\mathbf{x}(t)$, the value function V is still optimal for the remainder of the trajectory:

$$V(\mathbf{x}(t), t, t_f) = \min_{\mathbf{u}} \left(\int_t^{t_f} \mathcal{L}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau + Q(\mathbf{x}(t_f), t_f) \right). \quad (11.48)$$

One outcome is that the value function can be decomposed as:

$$V(\mathbf{x}(t_0), t_0, t_f) = V(\mathbf{x}(t_0), t_0, t) + V(\mathbf{x}(t), t, t_f). \quad (11.49)$$

This makes it possible to take the total time derivative above. A more rigorous derivation is possible using the calculus of variations.

The HJB equation is incredibly powerful, providing a PDE for the optimal solution of general nonlinear control problems. Typically, the HJB equation is solved numerically as a two-point boundary value problem, with boundary conditions $\mathbf{x}(0) = \mathbf{x}_0$ and $V(\mathbf{x}(t_f), t_f) = Q(\mathbf{x}(t_f), t_f)$, for example using a shooting method. However, a nonlinear control problem with a three-dimensional

state vector $\mathbf{x} \in \mathbb{R}^3$ will result in a three-dimensional PDE. Thus, optimal nonlinear control based on the HJB equation typically suffers from the curse of dimensionality. Phase-space clustering techniques have shown great promise in reducing the effective state-space dimension for systems that evolve on a low-dimensional attractor [376].

Discrete-time HJB and the Bellman equation

Bellman's optimal control is especially intuitive for discrete-time systems, where instead of optimizing over a function, we optimize over a discrete control sequence. Consider a discrete-time dynamical system

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k). \quad (11.50)$$

The cost is now given by

$$J(\mathbf{x}_0, \{\mathbf{u}_k\}_{k=0}^n, n) = \sum_{k=0}^n \mathcal{L}(\mathbf{x}_k, \mathbf{u}_k) + Q(\mathbf{x}_n, t_n). \quad (11.51)$$

Similarly, the value function is defined as the value of the cumulative cost function, starting at a point \mathbf{x}_0 assuming an optimal control policy \mathbf{u} :

$$V(\mathbf{x}_0, n) = \min_{\{\mathbf{u}_k\}_{k=0}^n} J(\mathbf{x}_0, \{\mathbf{u}_k\}_{k=0}^n, n). \quad (11.52)$$

Again, Bellman's principle of optimality states that an optimal control policy has the property that at any point along the optimal trajectory $\mathbf{x}(t)$, the remaining control policy is optimal with respect to this new initial state. Mathematically,

$$V(\mathbf{x}_0, n) = V(\mathbf{x}_0, k) + V(\mathbf{x}_k, n) \quad \forall k \in (0, n). \quad (11.53)$$

Thus, the value at an intermediate time step k may be written as

$$V(\mathbf{x}_k, n) = \left(\min_{\mathbf{u}_k} \mathcal{L}(\mathbf{x}_k, \mathbf{u}_k) \right) + \underbrace{V(\mathbf{x}_{k+1}, n)}_{\text{s.t. } \mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k)} \quad (11.54a)$$

$$= \min_{\mathbf{u}_k} (\mathcal{L}(\mathbf{x}_k, \mathbf{u}_k) + V(\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k), n)). \quad (11.54b)$$

It is also possible, given a value function $V(\mathbf{x}_k, n)$, to determine the next optimal control action \mathbf{u}_k by returning the \mathbf{u}_k that minimizes the above expression. This defines an *optimal policy* $\mathbf{u} = \boldsymbol{\pi}(\mathbf{x})$. Dropping the functional dependence of V on the end time, we then have

$$V(\mathbf{x}) = \min_{\mathbf{u}} (\mathcal{L}(\mathbf{x}, \mathbf{u}) + V(\mathbf{F}(\mathbf{x}, \mathbf{u}))) \quad (11.55a)$$

$$\boldsymbol{\pi}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{u}} (\mathcal{L}(\mathbf{x}, \mathbf{u}) + V(\mathbf{F}(\mathbf{x}, \mathbf{u}))). \quad (11.55b)$$

These form the Bellman equations.

Note that we have explicitly include the terminal time t_f in the terminal cost $Q(\mathbf{x}_n, t_n)$ and $Q(\mathbf{x}(t_f), t_f)$, as it there are situations when the arrival time should be minimized. However, it is also possible to include the time explicitly in the immediate cost $\mathcal{L}(\mathbf{x}, \mathbf{u}, t)$, for example to include a discount function $e^{-\gamma t}$ for future costs or rewards.

Suggested reading

Texts

- (1) **Reinforcement learning: An introduction**, by R. S. Sutton and A. G. Barto, 1998 [695].

Papers and reviews

- (1) **Q-learning**, by C. Watkins and P. Dayan, *Machine Learning*, 1992 [757].
- (2) **TD (λ) converges with probability 1**, by P. Dayan and T. J. Sejnowski, *Machine Learning*, 1994 [202].
- (3) **Human-level control through deep reinforcement learning**, by V. Mnih et al., *Nature*, 2015 [519].
- (4) **Mastering the game of go without human knowledge**, by D. Silver et al., *Nature*, 2017 [673].
- (5) **A tour of reinforcement learning: The view from continuous control**, by B. Recht, *Annual Review of Control, Robotics, and Autonomous Systems*, 2019 [601].

Blogs and lectures

- (1) **Deep Reinforcement Learning: Pong from Pixels**, by A. Karpathy, <http://karpathy.github.io/2016/05/31/r1/>.
- (2) **Introduction to Reinforcement Learning with David Silver**, by D. Silver, https://www.youtube.com/playlist?list=PLqYmG7hTraZBiG_XpjnPrSNw-1XQaM_gB

Homework

Exercise RL-1. This example will explore reinforcement learning on the game of tic-tac-toe. First, describe the states, actions, and rewards.

Next, design a policy iteration algorithm to optimize the policy π . Begin with a randomly chosen policy. Plot the value function on the board and describe the optimal policy.

How many policy iterations are required before the policy and value function converge? How many games were played at each policy iteration? Is this consistent with what you would expect a human learning would do?

Is there any structure or symmetry in the game that could be used to improve the learning rate? Implement a policy iteration that exploits this structure, and determine how many policy iterations are required before converging and how many games played per policy iteration.

Exercise RL-2. Repeat the above example using value iteration instead of policy iteration. Compare the number of iterations in both methods, along with the total training time.

Exercise RL-3. This exercise will develop a reinforcement learning controller for the fluid flow past a cylinder. There are several open-source codes that can be used to simulate simple fluid flows, such as the IBPM code at <https://github.com/cwrowley/ibpm/>.

Use reinforcement learning to develop a control law to force the cylinder wake to be symmetric. Describe the reward structure and what learning framework you chose. Also plot your results, including learning rates, performance, etc. How long did it take to train this controller (i.e., how many computational iterations, how much CPU time, etc.)?

Now, assume that the RL agent only has access to the lift and drag coefficients, C_L and C_D . Design an RL scheme to track a given reference lift value, say $C_L = 1$ or $C_L = -1$. See if you can make your controller track a reference that switches between these values. What if the reference lift is much larger, say $C_L = 2$ or $C_L = 5$?

Exercise RL-4. Install the AI Gym API and develop an RL controller for the classic control example of a pendulum on a cart. Explore different RL strategies.

Bibliography

- [1] P. Abbeel, A. Coates, and A. Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.
- [2] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Advances in neural information processing systems*, pages 1–8, 2007.
- [3] R. Abraham and J. E. Marsden. *Foundations of mechanics*, volume 36. Benjamin/Cummings Publishing Company Reading, Massachusetts, 1978.
- [4] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer-Verlag, 1988.
- [5] M. Agrawal, S. Vidyashankar, and K. Huang. On-chip implementation of ECoG signal data decoding in brain-computer interface. In *Mixed-Signal Testing Workshop (IMSTW), 2016 IEEE 21st International*, pages 1–6. IEEE, 2016.
- [6] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.
- [7] H.-S. Ahn, Y. Chen, and K. L. Moore. Iterative learning control: Brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007.
- [8] H. Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247, 1969.
- [9] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.
- [10] A. Alla and J. N. Kutz. Nonlinear model order reduction via dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 39(5):B778–B796, 2017.
- [11] A. Alla and J. N. Kutz. Randomized model order reduction. *Advances in Computational Mathematics*, 45(3):1251–1271, 2019.
- [12] E. P. Alves and F. Fiuza. Data-driven discovery of reduced plasma physics models from fully-kinetic simulations. *arXiv preprint arXiv:2011.01927*, 2020.
- [13] B. Amos, I. D. J. Rodriguez, J. Sacks, B. Boots, and J. Z. Kolter. Differentiable mpc for end-to-end planning and control. *arXiv preprint arXiv:1810.13400*, 2018.
- [14] W. Amrein and A.-M. Berthier. On support properties of L_p -functions and their Fourier transforms. *Journal of Functional Analysis*, 24(3):258–267, 1977.
- [15] D. Amsallem, J. Cortial, and C. Farhat. On-demand cfd-based aeroelastic predictions using a database of reduced-order bases and models. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, page 800, 2009.
- [16] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.
- [17] D. Amsallem, M. J. Zahr, and K. Washabaugh. Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. *Advances in Computational Mathematics*, 41(5):1187–1230, 2015.

- [18] J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.
- [19] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al. *LAPACK Users' guide*, volume 9. Siam, 1999.
- [20] J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly weather review*, 129(12):2884–2903, 2001.
- [21] C. A. Andersson and R. Bro. The n-way toolbox for matlab. *Chemometrics and intelligent laboratory systems*, 52(1):1–4, 2000.
- [22] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.
- [23] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.
- [24] A. C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, 2005.
- [25] H. Arbabi and I. Mezić. Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator. *SIAM J. Appl. Dyn. Syst.*, 16(4):2096–2126, 2017.
- [26] K. B. Ariyur and M. Krstić. *Real-Time Optimization by Extremum-Seeking Control*. Wiley, Hoboken, New Jersey, 2003.
- [27] T. Askham and J. N. Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.*, 17(1):380–416, 2018.
- [28] T. Askham, P. Zheng, A. Aravkin, and J. N. Kutz. Robust and scalable methods for the dynamic mode decomposition. *arXiv preprint arXiv:1712.01883*, 2017.
- [29] P. Astrid. Fast reduced order modeling technique for large scale LTV systems. In *American Control Conference, 2004. Proceedings of the 2004*, volume 1, pages 762–767. IEEE, 2004.
- [30] K. J. Aström and R. M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.
- [31] M. Azeez and A. Vakakis. Proper orthogonal decomposition (POD) of a class of vibroimpact oscillations. *Journal of Sound and vibration*, 240(5):859–889, 2001.
- [32] O. Azencot, W. Yin, and A. Bertozzi. Consistent dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(3):1565–1585, 2019.
- [33] K. Bache and M. Lichman. UCI machine learning repository, 2013.
- [34] P. J. Baddoo, B. Herrmann, B. J. McKeon, and S. L. Brunton. Kernel learning for robust dynamic mode decomposition: Linear and nonlinear disambiguation optimization (lando). *arXiv preprint arXiv:2106.01510*, 2021.
- [35] B. W. Bader and T. G. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, Dec. 2007.
- [36] S. Bagheri. Koopman-mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.
- [37] S. Bagheri. Effects of weak noise on oscillating flows: Linking quality factor, Floquet modes, and Koopman spectrum. *Physics of Fluids*, 26(9):094104, 2014.
- [38] S. Bagheri. Effects of weak noise on oscillating flows: Linking quality factor, Floquet modes, and Koopman spectrum. *Physics of Fluids*, 26(9):094104, 2014.
- [39] S. Bagheri, L. Brandt, and D. Henningson. Input-output analysis, model reduction and control of the flat-plate boundary layer. *J. Fluid Mechanics*, 620:263–298, 2009.
- [40] S. Bagheri, J. Hoepffner, P. J. Schmid, and D. S. Henningson. Input-output analysis and control design applied to a linear model of spatially developing flows. *Appl. Mech. Rev.*, 62(2):020803–1..27, 2009.

- [41] Z. Bai, S. L. Brunton, B. W. Brunton, J. N. Kutz, E. Kaiser, A. Spohn, and B. R. Noack. Data-driven methods in fluid dynamics: Sparse classification from experimental data. In *invited chapter for Whither Turbulence and Big Data in the 21st Century*, 2015.
- [42] Z. Bai, E. Kaiser, J. L. Proctor, J. N. Kutz, and S. L. Brunton. Dynamic mode decomposition for compressive system identification. *arXiv preprint arXiv:1710.07737*, 2017.
- [43] Z. Bai, T. Wimalajeewa, Z. Berger, G. Wang, M. Glauser, and P. K. Varshney. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA Journal*, 53(4):920–933, 2014.
- [44] O. Balabanov and A. Nouy. Randomized linear algebra for model reduction. part i: Galerkin methods and error estimation. *Advances in Computational Mathematics*, 45(5):2969–3019, 2019.
- [45] M. J. Balajewicz, E. H. Dowell, and B. R. Noack. Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation. *Journal of Fluid Mechanics*, 729:285–308, 2013.
- [46] M. Balasubramanian, S. Zabic, C. Bowd, H. W. Thompson, P. Wolenski, S. S. Iyengar, B. B. Karki, and L. M. Zangwill. A framework for detecting glaucomatous progression in the optic nerve head of an eye using proper orthogonal decomposition. *IEEE Transactions on Information Technology in Biomedicine*, 13(5):781–793, 2009.
- [47] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.
- [48] B. Bamieh and L. Giarré. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12:841–853, 2002.
- [49] A. Banaszuk, K. B. Ariyur, M. Krstić, and C. A. Jacobson. An adaptive algorithm for control of combustion instability. *Automatica*, 40(11):1965–1972, 2004.
- [50] A. Banaszuk, S. Narayanan, and Y. Zhang. Adaptive control of flow separation in a planar diffuser. *AIAA paper*, 617:2003, 2003.
- [51] A. Banaszuk, Y. Zhang, and C. A. Jacobson. Adaptive control of combustion instability using extremum-seeking. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 416–422. IEEE, 2000.
- [52] S. Banks. Infinite-dimensional Carleman linearization, the Lie series and optimal control of non-linear partial differential equations. *International journal of systems science*, 23(5):663–675, 1992.
- [53] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.
- [54] R. G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–120, 2007.
- [55] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.
- [56] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667–672, 2004.
- [57] J. Basley, L. R. Pastur, N. Delprat, and F. Lusseyran. Space-time aspects of a three-dimensional multi-modulated open cavity flow. *Physics of Fluids (1994-present)*, 25(6):064105, 2013.
- [58] J. Basley, L. R. Pastur, F. Lusseyran, T. M. Faure, and N. Delprat. Experimental investigation of global structures in an incompressible cavity flow using time-resolved PIV. *Experiments in Fluids*, 50(4):905–918, 2011.
- [59] T. Baumeister, S. L. Brunton, and J. N. Kutz. Deep learning and model predictive control

- for self-tuning mode-locked lasers. *JOSA B*, 35(3):617–626, 2018.
- [60] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.
- [61] P. W. Bearman. On vortex shedding from a circular cylinder in the critical reynolds number regime. *Journal of Fluid Mechanics*, 37(3):577–585, 1969.
- [62] J. F. Beaudoin, O. Cadot, J. L. Aider, and J. E. Wesfreid. Bluff-body drag reduction by extremum-seeking control. *Journal of Fluids and Structures*, 22:973–978, 2006.
- [63] J.-F. Beaudoin, O. Cadot, J.-L. Aider, and J.-E. Wesfreid. Drag reduction of a bluff body using adaptive control methods. *Physics of Fluids*, 18(8):085107, 2006.
- [64] R. Becker, R. King, R. Petz, and W. Nitsche. Adaptive closed-loop control on a high-lift configuration using extremum seeking. *AIAA Journal*, 45(6):1382–92, 2007.
- [65] S. Beetham and J. Capecelatro. Formulating turbulence closures using sparse regression with embedded form invariance. *Physical Review Fluids*, 5(8):084611, 2020.
- [66] S. Beetham, R. O. Fox, and J. Capecelatro. Sparse identification of multiphase turbulence closures for coupled fluid–particle flows. *Journal of Fluid Mechanics*, 914, 2021.
- [67] G. Beintema, A. Corbetta, L. Biferale, and F. Toschi. Controlling rayleigh-bénard convection via reinforcement learning. *arXiv preprint arXiv:2003.14358*, 2020.
- [68] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):711–720, 1997.
- [69] G. Bellani. Experimental studies of complex flows through image-based techniques. 2011.
- [70] R. Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952.
- [71] R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.
- [72] B. A. Belson, J. H. Tu, and C. W. Rowley. Algorithm 945: modred—a parallelized model reduction library. *ACM Transactions on Mathematical Software*, 40(4):30, 2014.
- [73] M. Benedicks. On Fourier transforms of functions supported on sets of finite Lebesgue measure. *Journal of mathematical analysis and applications*, 106(1):180–183, 1985.
- [74] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.
- [75] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox. *Model Reduction and Approximation: Theory and Algorithms*, volume 15. SIAM, 2017.
- [76] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.*, 57(4):483–531, 2015.
- [77] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.
- [78] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. Slicot—a subroutine library in systems and control theory. In *Applied and computational control, signals, and circuits*, pages 499–539. Springer, 1999.
- [79] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. B. Amor. Estimation of perturbations in robotic behavior using dynamic mode decomposition. *Journal of Advanced Robotics*, 29(5):331–343, 2015.
- [80] G. Berkooz, P. Holmes, and J. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25:539–575, 1993.
- [81] D. P. Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.
- [82] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press,

- 2014.
- [83] D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE, 1995.
- [84] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms i. *Communications on pure and applied mathematics*, 44(2):141–183, 1991.
- [85] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and Computational Fluid Dynamics*, pages 1–15, 2020.
- [86] L. Biferale, F. Bonaccorso, M. Buzzicotti, P. Clark Di Leoni, and K. Gustavsson. Zermelo’s problem: Optimal point-to-point navigation in 2d turbulent flows using reinforcement learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103138, 2019.
- [87] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.
- [88] P. Binetti, K. B. Ariyur, M. Krstić, and F. Bernelli. Formation flight optimization using extremum seeking feedback. *Journal of Guidance, Control, and Dynamics*, 26(1):132–142, 2003.
- [89] G. D. Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences*, 17(12):656–660, 1931.
- [90] G. D. Birkhoff and B. O. Koopman. Recent contributions to the ergodic theory. *Proceedings of the National Academy of Sciences*, 18(3):279–282, 1932.
- [91] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [92] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.
- [93] D. Bistrian and I. Navon. Randomized dynamic mode decomposition for non-intrusive reduced order modelling. *International Journal for Numerical Methods in Engineering*, 2016.
- [94] D. A. Bistrian and I. M. Navon. An improved algorithm for the shallow water equations model reduction: Dynamic mode decomposition vs POD. *International Journal for Numerical Methods in Fluids*, 2015.
- [95] D. A. Bistrian and I. M. Navon. Randomized dynamic mode decomposition for nonintrusive reduced order modelling. *International Journal for Numerical Methods in Engineering*, 112(1):3–25, 2017.
- [96] P. Bondi, G. Casalino, and L. Gambardella. On the iterative learning control theory for robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(1):14–22, 1988.
- [97] J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.
- [98] L. Boninsegna, F. Nüske, and C. Clementi. Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, 148(24):241723, 2018.
- [99] J. L. Borges. The library of Babel. *Collected fictions*, 1998.
- [100] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [101] H. Boustani and Y. Kamp. Autoassociative memory by multilayer perceptron and singular values decomposition. *Biol Cybern*, 59:291–294, 1989.
- [102] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [103] S. Boyd, L. O. Chua, and C. A. Desoer. Analytical foundations of volterra series. *IMA Journal of Mathematical Control and Information*, 1(3):243–282, 1984.
- [104] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009.
- [105] S. J. Bradtko and A. G. Barto. Linear least-squares algorithms for temporal difference

- learning. *Machine learning*, 22(1):33–57, 1996.
- [106] J. J. Bramburger and J. N. Kutz. Poincaré maps for multiscale physics discovery and nonlinear floquet theory. *Physica D: Nonlinear Phenomena*, 408:132479, 2020.
- [107] J. J. Bramburger, J. N. Kutz, and S. L. Brunton. Data-driven stabilization of periodic orbits. *IEEE Access*, 9:43504–43521, 2021.
- [108] A. I. Bratcu, I. Munteanu, S. Bacha, and B. Raison. Maximum power point tracking of grid-connected photovoltaic arrays by using extremum seeking control. *CEAI*, 10(4):3–12, 2008.
- [109] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.
- [110] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [111] L. Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.
- [112] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
- [113] M. Brenner, J. Eldredge, and J. Freund. Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids*, 4(10):100501, 2019.
- [114] I. Bright, G. Lin, and J. N. Kutz. Compressive sensing and machine learning strategies for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids*, 25(127102):1–15, 2013.
- [115] I. Bright, G. Lin, and J. N. Kutz. Classification of spatio-temporal data via asynchronous sparse sampling: Application to flow around a cylinder. *SIAM Multiscale modeling and simulation*, 14(2):823–838, 2016.
- [116] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.
- [117] D. Bristow, M. Tharayil, A. G. Alleyne, et al. A survey of iterative learning control. *Control Systems, IEEE*, 26(3):96–114, 2006.
- [118] R. Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171, 1997.
- [119] A. Broad, T. Murphey, and B. Argall. Learning models for shared control of human-machine systems with unknown dynamics. *Robotics: Science and Systems Proceedings*, 2017.
- [120] R. W. Brockett. Volterra series and geometric control theory. *Automatica*, 12(2):167–176, 1976.
- [121] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [122] D. Broomhead and R. Jones. Time-series analysis. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 423, pages 103–121. The Royal Society, 1989.
- [123] D. S. Broomhead and D. Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- [124] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Sparse sensor placement optimization for classification. *SIAM Journal on Applied Mathematics*, 76(5):2099–2122, 2016.
- [125] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.
- [126] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermit-

- tently forced linear system. *Nature Communications*, 8(19):1–9, 2017.
- [127] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE*, 11(2):e0150171, 2016.
- [128] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz. Modern Koopman theory for dynamical systems. *arXiv preprint arXiv:2102.12086*, 2021.
- [129] S. L. Brunton, X. Fu, and J. N. Kutz. Extremum-seeking control of a mode-locked laser. *IEEE Journal of Quantum Electronics*, 49(10):852–861, 2013.
- [130] S. L. Brunton, X. Fu, and J. N. Kutz. Self-tuning fiber lasers. *IEEE Journal of Selected Topics in Quantum Electronics*, 20(5), 2014.
- [131] S. L. Brunton and B. R. Noack. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67:050801–1–050801–48, 2015.
- [132] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.
- [133] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
- [134] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC NOLCOS*, 49(18):710–715, 2016.
- [135] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, 2015.
- [136] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165, 2015.
- [137] S. L. Brunton and C. W. Rowley. Maximum power point tracking for photovoltaic optimization using ripple-based extremum seeking control. *IEEE Transactions on Power Electronics*, 25(10):2531–2540, 2010.
- [138] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1716–1732, 2014.
- [139] D. Buche, P. Stoll, R. Dornberger, and P. Koumoutsakos. Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(4):460–473, 2002.
- [140] M. Budišić and I. Mezić. An approximate parametrization of the ergodic partition using time averaged observables. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3162–3168. IEEE, 2009.
- [141] M. Budišić and I. Mezić. Geometry of the ergodic quotient reveals coherent structures in flows. *Physica D: Nonlinear Phenomena*, 241(15):1255–1269, 2012.
- [142] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism a). *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.
- [143] A. Buhr and K. Smetana. Randomized local model order reduction. *SIAM journal on scientific computing*, 40(4):A2120–A2151, 2018.
- [144] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.
- [145] D. Burov, D. Giannakis, K. Manohar, and A. Stuart. Kernel analog forecasting: Multiscale test problems. *arXiv preprint arXiv:2005.06623*, 2020.
- [146] P. A. Businger and G. H. Golub. Algorithm 358: Singular value decomposition of a complex matrix [f1, 4, 5]. *Communications of the ACM*, 12(10):564–565, 1969.
- [147] J. Callahan, K. Maeda, and S. L. Brunton. Robust reconstruction of flow fields from

- limited measurements. *Physical Review Fluids*, 4(103907), 2019.
- [148] J. L. Callahan, S. L. Brunton, and J.-C. Loiseau. On the role of nonlinear correlations in reduced-order modeling. *arXiv preprint arXiv:2106.02409*, 2021.
- [149] J. L. Callahan, J.-C. Loiseau, G. Rigas, and S. L. Brunton. Nonlinear stochastic modelling with langevin regression. *Proceedings of the Royal Society A*, 477(2250):20210092, 2021.
- [150] J. L. Callahan, G. Rigas, J.-C. Loiseau, and S. L. Brunton. An empirical mean-field model of symmetry-breaking in a turbulent wake. *arXiv preprint arXiv:2105.13990*, 2021.
- [151] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.
- [152] E. Cambria, G.-B. Huang, L. L. C. Kasun, H. Zhou, C. M. Vong, J. Lin, J. Yin, Z. Cai, Q. Liu, K. Li, et al. Extreme learning machines [trends & controversies]. *IEEE Intelligent Systems*, 28(6):30–59, 2013.
- [153] E. J. Candès. Compressive sensing. *Proceedings of the International Congress of Mathematics*, 2006.
- [154] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11–1–11–37, 2011.
- [155] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [156] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications in Pure and Applied Mathematics*, 8(1207–1223), 59.
- [157] E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.
- [158] E. J. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.
- [159] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, pages 21–30, 2008.
- [160] Y. Cao, J. Zhu, Z. Luo, and I. Navon. Reduced-order modeling of the upper tropical pacific ocean model using proper orthogonal decomposition. *Computers & Mathematics with Applications*, 52(8):1373–1386, 2006.
- [161] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo. A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 53(10):1571–1583, 2007.
- [162] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.
- [163] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.
- [164] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.
- [165] T. Carleman. Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Mathematica*, 59(1):63–87, 1932.
- [166] T. Carleman. Sur la théorie de l'équation intégrodifférentielle de boltzmann. *Acta Mathematica*, 60(1):91–146, 1933.
- [167] T. Carleman. Sur les systemes lineaires aux dérivées partielles du premier ordrea deux variables. *CR Acad. Sci. Paris*, 197:471–474, 1933.

- [168] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition. *Psychometrika*, 35:283–319, 1970.
- [169] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.
- [170] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.
- [171] K. P. Champion, S. L. Brunton, and J. N. Kutz. Discovery of nonlinear multiscale systems: Sampling strategies and embeddings. *SIAM Journal on Applied Dynamical Systems*, 18(1):312–333, 2019.
- [172] R. Chartrand. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011, 2011.
- [173] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, 78(7):808–817, 2000.
- [174] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [175] K. K. Chen and C. W. Rowley. Normalized coprime robust stability and performance guarantees for reduced-order controllers. *IEEE Transactions on Automatic Control*, 58(4):1068–1073, 2013.
- [176] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.
- [177] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.
- [178] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.
- [179] Y. Chen, K. L. Moore, and H.-S. Ahn. Iterative learning control. In *Encyclopedia of the Sciences of Learning*, pages 1648–1652. Springer, 2012.
- [180] S. Cherry. Singular value decomposition analysis and canonical correlation analysis. *Journal of Climate*, 9(9):2003–2009, 1996.
- [181] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [182] J. Choi, M. Krstić, K. Ariyur, and J. Lee. Extremum seeking control for discrete-time systems. *IEEE Transactions on Automatic Control*, 47(2):318–323, FEB 2002.
- [183] Y. Choi, D. Amsallem, and C. Farhat. Gradient-based constrained optimization using a database of linear reduced-order models. *arXiv preprint arXiv:1506.07849*, 2015.
- [184] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale. Flow navigation by smart microswimmers via reinforcement learning. *Physical review letters*, 118(15):158004, 2017.
- [185] T. Colonius and K. Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197:2131–2146, 2008.
- [186] J. W. Cooley, P. A. Lewis, and P. D. Welch. Historical notes on the fast Fourier transform. *Proceedings of the IEEE*, 55(10):1675–1677, 1967.
- [187] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex

- Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- [188] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [189] M. C. Cross and P. C. Hohenberg. Pattern formation outside of equilibrium. *Reviews of modern physics*, 65(3):851, 1993.
- [190] J. P. Crutchfield and B. S. McNamara. Equations of motion from a data series. *Complex systems*, 1:417–452, 1987.
- [191] M. Dam, M. Brøns, J. Juul Rasmussen, V. Naulin, and J. S. Hesthaven. Sparse identification of a predator-prey system from simulation data of a convection model. *Physics of Plasmas*, 24(2):022310, 2017.
- [192] B. C. Daniels and I. Nemenman. Automated adaptive inference of phenomenological dynamical models. *Nature communications*, 6, 2015.
- [193] B. C. Daniels and I. Nemenman. Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PloS one*, 10(3):e0119821, 2015.
- [194] S. Das and D. Giannakis. Delay-coordinate maps and the spectra of Koopman operators. *arXiv preprint arXiv:1706.08544*, 2017.
- [195] S. Das and D. Giannakis. Delay-coordinate maps and the spectra of Koopman operators. *Journal of Statistical Physics*, 175(6):1107–1145, 2019.
- [196] S. Das and D. Giannakis. Koopman spectra in reproducing kernel Hilbert spaces. *Applied and Computational Harmonic Analysis*, 49(2):573–607, 2020.
- [197] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE transactions on information theory*, 36(5):961–1005, 1990.
- [198] L. Davis et al. *Handbook of genetic algorithms*, volume 115. Van Nostrand Reinhold New York, 1991.
- [199] N. D. Daw, J. P. O’doherly, P. Dayan, B. Seymour, and R. J. Dolan. Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095):876–879, 2006.
- [200] S. T. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):1–19, 2016.
- [201] P. Dayan and L. F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.
- [202] P. Dayan and T. J. Sejnowski. Td (λ) converges with probability 1. *Machine Learning*, 14(3):295–301, 1994.
- [203] B. de Silva, D. M. Higdon, S. L. Brunton, and J. N. Kutz. Discovery of physics from data: Universal laws and discrepancy models. *arXiv preprint arXiv:1906.07906*, 2019.
- [204] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton. PySINDy: a Python package for the sparse identification of nonlinear dynamics from data. *Journal of Open Source Software*, 5(49):2104, 2020.
- [205] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [206] N. Deng, B. R. Noack, M. Morzyński, and L. R. Pastur. Galerkin force model for transient and post-transient dynamics of the fluidic pinball. *Journal of Fluid Mechanics*, 918, 2021.
- [207] Z. Deng, C. He, Y. Liu, and K. C. Kim. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Physics of Fluids*, 31(12):125111, 2019.
- [208] S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *Automatic Control, IEEE Transactions on*, 41(7):930–942, 1996.
- [209] D. Donoho. 50 years of data science. In *Based on a Presentation at the Tukey Centennial*

- Workshop*. NJ Princeton, 2015.
- [210] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [211] D. L. Donoho and M. Gavish. Code supplement to “The optimal hard threshold for singular values is $4/\sqrt{3}$ ”. <http://purl.stanford.edu/vg705qn9070>, 2014.
- [212] D. L. Donoho, I. M. Johnstone, J. C. Hoch, and A. S. Stern. Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 41–81, 1992.
- [213] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [214] J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.
- [215] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.
- [216] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.
- [217] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg. Vision-based high-speed driving with a deep dynamic observer. *IEEE Robotics and Automation Letters*, 4(2):1564–1571, 2019.
- [218] J. Drgona, K. Kis, A. Tuor, D. Vrabie, and M. Klauco. Differentiable predictive control: An mpc alternative for unknown nonlinear systems using constrained deep learning. *arXiv preprint arXiv:2011.03699*, 2020.
- [219] P. Drineas and M. W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. *Linear algebra and its applications*, 420(2-3):553–571, 2007.
- [220] Z. Drmac and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016.
- [221] Q. Du and M. Gunzburger. Model reduction by proper orthogonal decomposition coupled with centroidal voronoi tessellations (keynote). In *ASME 2002 Joint US-European Fluids Engineering Division Conference*, pages 1401–1406. American Society of Mechanical Engineers, 2002.
- [222] Y. Duan, M. Andrychowicz, B. C. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*, 2017.
- [223] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.
- [224] J. A. Duersch and M. Gu. Randomized QR with column pivoting. *SIAM Journal on Scientific Computing*, 39(4):C263–C291, 2017.
- [225] D. Duke, D. Honnery, and J. Soria. Experimental investigation of nonlinear instabilities in annular liquid sheets. *Journal of Fluid Mechanics*, 691:594–604, 2012.
- [226] D. Duke, J. Soria, and D. Honnery. An error analysis of the dynamic mode decomposition. *Experiments in fluids*, 52(2):529–542, 2012.
- [227] G. E. Dullerud and F. Paganini. *A course in robust control theory: A convex approach*. Texts in Applied Mathematics. Springer, Berlin, Heidelberg, 2000.
- [228] R. Dunne and B. J. McKeon. Dynamic stall on a pitching and surging airfoil. *Experiments in Fluids*, 56(8):1–15, 2015.
- [229] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual Reviews of Fluid Mechanics*, 51:357–377, 2019.
- [230] T. Duriez, S. L. Brunton, and B. R. Noack. *Machine Learning Control: Taming Nonlinear*

- Dynamics and Turbulence*. Springer, 2016.
- [231] T. Duriez, V. Parezanović, L. Cordier, B. R. Noack, J. Delville, J.-P. Bonnet, M. Segond, and M. Abel. Closed-loop turbulence control using machine learning. *arXiv preprint arXiv:1404.4589*, 2014.
- [232] T. Duriez, V. Parezanovic, J.-C. Laurentie, C. Fourment, J. Delville, J.-P. Bonnet, L. Cordier, B. R. Noack, M. Segond, M. Abel, N. Gautier, J.-L. Aider, C. Raibaud, C. Cuvier, M. Stanislas, and S. L. Brunton. Closed-loop control of experimental shear flows using machine learning. AIAA Paper 2014-2219, 7th Flow Control Conference, 2014.
- [233] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [234] J. L. Eftang, A. T. Patera, and E. M. Rønquist. An “hp” certified reduced basis method for parametrized elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 32(6):3170–3200, 2010.
- [235] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- [236] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017.
- [237] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for real-time object detection. *Journal of Real-Time Image Processing*, 2016.
- [238] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Randomized dynamic mode decomposition. *arXiv preprint arXiv:1702.02912*, 2017.
- [239] N. B. Erichson, K. Manohar, S. L. Brunton, and J. N. Kutz. Randomized CP tensor decomposition. *arXiv preprint arXiv:1703.09074*.
- [240] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton. Randomized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(4):1867–1891, 2019.
- [241] N. B. Erichson, L. Mathelin, Z. Yao, S. L. Brunton, M. W. Mahoney, and J. N. Kutz. Shallow neural networks for fluid flow reconstruction with limited sensors. *Proceedings of the Royal Society A*, 476(2238):20200097, 2020.
- [242] N. B. Erichson, S. Voronin, S. L. Brunton, and J. N. Kutz. Randomized matrix decompositions using R. *arXiv preprint arXiv:1608.02148*, 2016.
- [243] T. Esumi, J. W. Kimball, P. T. Krein, P. L. Chapman, and P. Midya. Dynamic maximum power point tracking of photovoltaic arrays using ripple correlation control. *Ieee Transactions On Power Electronics*, 21(5):1282–1291, Sept. 2006.
- [244] E. Even-Dar, Y. Mansour, and P. Bartlett. Learning rates for q-learning. *Journal of machine learning Research*, 5(1), 2003.
- [245] R. Everson and L. Sirovich. Karhunen–Loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.
- [246] N. Fabbiane, O. Semeraro, S. Bagheri, and D. S. Henningson. Adaptive and model-based control theory applied to convectively unstable flows. *Appl. Mech. Rev.*, 66(6):060801–1–20, 2014.
- [247] D. Fan, L. Yang, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis. Reinforcement learning for bluff body active flow control in experiments and simulations. *Proceedings of the National Academy of Sciences*, 117(42):26091–26098, 2020.
- [248] D. D. Fan, A.-a. Agha-mohammadi, and E. A. Theodorou. Deep learning tubes for tube mpc. *arXiv preprint arXiv:2002.01587*, 2020.
- [249] B. Feeny. On proper orthogonal co-ordinates as indicators of modal activity. *Journal of Sound and Vibration*, 255(5):805–817, 2002.
- [250] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical*

- Character*, 222:309–368, 1922.
- [251] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.
- [252] P. J. Fleming and R. C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10:1223–1241, 2002.
- [253] N. Fonzi, S. L. Brunton, and U. Fasel. Data-driven nonlinear aeroelastic models of morphing wings for control. *Proceedings of the Royal Society A*, 476(2239):20200079, 2020.
- [254] J. Fourier. *Theorie analytique de la chaleur, par M. Fourier*. Chez Firmin Didot, père et fils, 1822.
- [255] J. B. J. Fourier. *The analytical theory of heat*. The University Press, 1878.
- [256] J. E. Fowler. Compressive-projection principal component analysis. *IEEE Transactions on Image Processing*, 18(10):2230–2242, 2009.
- [257] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [258] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [259] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.
- [260] G. Froyland, G. A. Gottwald, and A. Hammerlindl. A computational method to extract macroscopic variables and their dynamics in multiscale systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1816–1846, 2014.
- [261] G. Froyland, G. A. Gottwald, and A. Hammerlindl. A trajectory-free framework for analysing multiscale systems. *Physica D: Nonlinear Phenomena*, 328:34–43, 2016.
- [262] X. Fu, S. L. Brunton, and J. Nathan Kutz. Classification of birefringence in mode-locked fiber lasers using machine learning and sparse representation. *Optics express*, 22(7):8585–8597, 2014.
- [263] K. Fujii and Y. Kawahara. Dynamic mode decomposition in vector-valued reproducing kernel hilbert spaces for extracting dynamical structure among observables. *Neural Networks*, 117:94–103, 2019.
- [264] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.
- [265] K. Fukagata, S. Kern, P. Chatelain, P. Koumoutsakos, and N. Kasagi. Evolutionary optimization of an anisotropic compliant surface for turbulent friction drag reduction. *Journal of Turbulence*, 9(35):1–17, 2008.
- [266] F. Fukushima. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetic*, 36:193–202, 1980.
- [267] H. Gao, J. Lam, C. Wang, and Y. Wang. Delay-dependent output-feedback stabilisation of discrete-time systems with time-varying state delay. *IEE Proceedings-Control Theory and Applications*, 151(6):691–698, 2004.
- [268] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.
- [269] J. L. Garriga and M. Soroush. Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, 49(8):3505–3515, 2010.
- [270] C. Gauss. *Nachlass: Theoria interpolationis methodo nova tractata*, volume werke. *Königliche Gesellschaft der Wissenschaften, Göttingen*, 1866.
- [271] C.-F. Gauss. *Theoria combinationis observationum erroribus minimis obnoxiae*, volume 1. Henricus Dieterich, 1823.
- [272] N. Gautier, J.-L. Aider, T. Duriez, B. Noack, M. Segond, and M. Abel. Closed-loop sepa-

- ration control using machine learning. *Journal of Fluid Mechanics*, 770:442–457, 2015.
- [273] M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.
- [274] M. Gazzola, B. Hejazialhosseini, and P. Koumoutsakos. Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers. *SIAM Journal on Scientific Computing*, 36(3):B622–B639, 2014.
- [275] M. Gazzola, A. Tchieu, D. Alexeev, A. De Brauer, and P. Koumoutsakos. Learning to school in the presence of hydrodynamic interactions. *J. Fluid Mech.*, 789, 2016.
- [276] M. Gazzola, O. V. Vasilyev, and P. Koumoutsakos. Shape optimization for drag reduction in linked bodies using evolution strategies. *Computers & Structures*, 89(11):1224–1231, 2011.
- [277] T. Geijtenbeek, M. Van De Panne, and A. F. Van Der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32(6):1–11, 2013.
- [278] G. Gelbert, J. P. Moeck, C. O. Paschereit, and R. King. Advanced algorithms for gradient estimation in one-and two-parameter extremum seeking controllers. *Journal of Process Control*, 22(4):700–709, 2012.
- [279] P. Geiß, S. Klus, J. Eisert, and C. Schütte. Multidimensional approximation of nonlinear dynamical systems. *Journal of Computational and Nonlinear Dynamics*, 14(6), 2019.
- [280] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6):643–660, 2001.
- [281] J. J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern recognition*, 14(1):375–381, 1981.
- [282] A. C. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.
- [283] A. C. Gilbert, J. Y. Park, and M. B. Wakin. Sketched SVD: Recovering spectral features from compressive measurements. *ArXiv e-prints*, 2012.
- [284] A. C. Gilbert, M. J. Strauss, and J. A. Tropp. A tutorial on fast Fourier sampling. *IEEE Signal Processing Magazine*, pages 57–66, 2008.
- [285] C. Gin, B. Lusch, S. L. Brunton, and J. N. Kutz. Deep learning models for global coordinate transformations that linearise PDEs. *European Journal of Applied Mathematics*, pages 1–25, 2020.
- [286] C. Gin, B. Lusch, S. L. Brunton, and J. N. Kutz. Deep learning models for global coordinate transformations that linearise pdes. *European Journal of Applied Mathematics*, 32(3):515–539, 2021.
- [287] C. R. Gin, D. E. Shea, S. L. Brunton, and J. N. Kutz. Deepgreen: Deep learning of green’s functions for nonlinear boundary value problems. *arXiv preprint arXiv:2101.07206*, 2020.
- [288] B. Glaz, L. Liu, and P. P. Friedmann. Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework. *AIAA journal*, 48(10):2418–2429, 2010.
- [289] P. J. Goddard and K. Glover. Controller approximation: approaches for preserving H_∞ performance. *IEEE Transactions on Automatic Control*, 43(7):858–871, 1998.
- [290] D. E. Goldberg. *Genetic algorithms*. Pearson Education India, 2006.
- [291] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.
- [292] G. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem $ax + xb = c$. *IEEE Transactions on Automatic Control*, 24(6):909–913, 1979.
- [293] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions.

- Numerical Mathematics*, 14:403–420, 1970.
- [294] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [295] R. González-García, R. Rico-Martínez, and I. G. Kevrekidis. Identification of distributed parameter systems: A neural net based approach. *Computers & chemical engineering*, 22:S965–S968, 1998.
- [296] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>
- [297] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [298] A. Goza and T. Colonius. Modal decomposition of fluid–structure interaction with application to flag flapping. *Journal of Fluids and Structures*, 81:728–737, 2018.
- [299] M. Grant, S. Boyd, and Y. Ye. *Cvx: Matlab software for disciplined convex programming*, 2008.
- [300] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.
- [301] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997.
- [302] M. S. Grewal. Kalman filtering. In *International Encyclopedia of Statistical Science*, pages 705–708. Springer, 2011.
- [303] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams. Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction. *Journal of Fluid Mechanics*, 700:16–28, 2012.
- [304] J. Grosek and J. N. Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.
- [305] M. Gu. Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):1139–1173, 2015.
- [306] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.
- [307] Y. Guan, S. L. Brunton, and I. Novosselov. Sparse nonlinear models of chaotic electroconvection. *Royal Society Open Science*, 8(8):202367, 2021.
- [308] F. Guéniat, L. Mathelin, and M. Y. Hussaini. A statistical learning strategy for closed-loop control of fluid flows. *Theor. Comp. Fluid Dyn.*, 30(6):497–510, 2016.
- [309] F. Guéniat, L. Mathelin, and L. Pastur. A dynamic mode decomposition approach for large and arbitrarily sampled systems. *Physics of Fluids*, 27(2):025113, 2015.
- [310] P. Gunnarson, I. Mandralis, G. Novati, P. Koumoutsakos, and J. O. Dabiri. Learning efficient navigation in vortical flow fields. *arXiv preprint arXiv:2102.10536*, 2021.
- [311] D. R. Gurevich, P. A. Reinbold, and R. O. Grigoriev. Robust and optimal sparse regression for nonlinear pde models. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(10):103113, 2019.
- [312] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forsell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.
- [313] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [314] N. Halko, P.-G. Martinsson, Y. Shkolnisky, and M. Tygert. An algorithm for the principal component analysis of large data sets. *SIAM Journal on Scientific Computing*, 33:2580–2594, 2011.
- [315] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Prob-

- abilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.
- [316] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- [317] S. J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA Journal of Numerical Analysis*, 2(3):303–323, 1982.
- [318] S. Han and B. Feeny. Application of proper orthogonal decomposition to structural vibration analysis. *Mechanical Systems and Signal Processing*, 17(5):989–1001, 2003.
- [319] N. Hansen, A. S. Niederberger, L. Guzzella, and P. Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197, 2009.
- [320] D. Harrison Jr and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.
- [321] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970. Available at <http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf>
- [322] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.
- [323] M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*, 2015.
- [324] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [325] M. Heath, A. Laub, C. Paige, and R. Ward. Computing the singular value decomposition of a product of two matrices. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1147–1159, 1986.
- [326] M. Heideman, D. Johnson, and C. Burrus. Gauss and the history of the fast Fourier transform. *IEEE ASSP Magazine*, 1(4):14–21, 1984.
- [327] W. Heisenberg. Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik. In *Original Scientific Papers Wissenschaftliche Originalarbeiten*, pages 478–504. Springer, 1985.
- [328] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. De-biasing the dynamic mode decomposition for applied Koopman spectral analysis. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, 2017.
- [329] M. S. Hemati, M. O. Williams, and C. W. Rowley. Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids (1994-present)*, 26(11):111701, 2014.
- [330] K. K. Herrity, A. C. Gilbert, and J. A. Tropp. Sparse approximation via iterative thresholding. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, pages III–III. IEEE, 2006.
- [331] B. Herrmann, P. J. Baddoo, R. Semaan, S. L. Brunton, and B. J. McKeon. Data-driven resolvent analysis. *arXiv preprint arXiv:2010.02181*, 2020.
- [332] J. S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2015.
- [333] T. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.
- [334] G. E. Hinton and T. J. Sejnowski. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986.

- [335] S. M. Hirsh, S. M. Ichinaga, S. L. Brunton, J. N. Kutz, and B. W. Brunton. Structured time-delay models for dynamical systems with connections to frenet-serret frame. *arXiv preprint arXiv:2101.08344*, 2021.
- [336] B. L. Ho and R. E. Kalman. Effective construction of linear state-variable models from input/output data. In *Proceedings of the 3rd Annual Allerton Conference on Circuit and System Theory*, pages 449–459, 1965.
- [337] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.
- [338] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [339] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [340] J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [341] P. Holmes and J. Guckenheimer. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, Heidelberg, 1983.
- [342] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 2nd paperback edition, 2012.
- [343] E. Hopf. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Communications on Pure and Applied mathematics*, 3(3):201–230, 1950.
- [344] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- [345] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [346] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, Sept. 1933.
- [347] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520, Oct. 1933.
- [348] C. Huang, W. E. Anderson, M. E. Harvazinski, and V. Sankaran. Analysis of self-excited combustion instabilities using decomposition techniques. In *51st AIAA Aerospace Sciences Meeting*, pages 1–18, 2013.
- [349] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *Journal of Physiology*, 160:106–154, 1962.
- [350] P. J. Huber. Robust statistics. In *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer, 2011.
- [351] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.
- [352] S. J. Illingworth, A. S. Morgans, and C. W. Rowley. Feedback control of flow resonances using balanced reduced-order models. *Journal of Sound and Vibration*, 330(8):1567–1581, 2010.
- [353] E. Jacobsen and R. Lyons. The sliding DFT. *IEEE Signal Processing Magazine*, 20(2):74–80, 2003.
- [354] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.
- [355] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*. Springer, 2013.
- [356] M. C. Johnson, S. L. Brunton, N. B. Kundtz, and J. N. Kutz. Extremum-seeking control

- of a beam pattern of a reconfigurable holographic metamaterial antenna. *Journal of the Optical Society of America A*, 33(1):59–68, 2016.
- [357] R. A. Johnson and D. Wichern. *Multivariate analysis*. Wiley Online Library, 2002.
- [358] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
- [359] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
- [360] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.
- [361] M. R. Jovanović. From bypass transition to flow control and data-driven turbulence modeling: An input-output viewpoint. *Annu. Rev. Fluid. Mech.*, 53(1), 2021.
- [362] M. R. Jovanović and B. Bamieh. Componentwise energy amplification in channel flows. *J. Fluid Mech.*, 534:145–183, 2005.
- [363] M. R. Jovanović, P. J. Schmid, and J. W. Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2):024103, 2014.
- [364] J. N. Juang. *Applied System Identification*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1994.
- [365] J. N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.
- [366] J. N. Juang, M. Phan, L. G. Horta, and R. W. Longman. Identification of observer/Kalman filter Markov parameters: Theory and experiments. Technical Memorandum 104069, NASA, 1991.
- [367] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages 182–193. Orlando, FL, 1997.
- [368] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- [369] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- [370] K. Kaheman, E. Kaiser, B. Strom, J. N. Kutz, and S. L. Brunton. Learning discrepancy models from experimental data. *CDC [arXiv preprint arXiv:1909.08574]*, 2019.
- [371] K. Kaheman, J. N. Kutz, and S. L. Brunton. Sindy-pi: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, 476(2242):20200279, 2020.
- [372] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of Koopman eigenfunctions for control. *arXiv preprint arXiv:1707.01146*, 2017.
- [373] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London A*, 474(2219), 2018.
- [374] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London A*, 474(2219), 2018.
- [375] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Osth, S. Krajnovic, and R. K. Niven. Cluster-based reduced-order modelling of a mixing layer. *J. Fluid Mech.*, 754:365–414, 2014.
- [376] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajnović, and R. K. Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.
- [377] S. M. Kakade. A natural policy gradient. *Advances in neural information processing systems*,

- 14, 2001.
- [378] M. Kalia, S. L. Brunton, H. G. Meijer, C. Brune, and J. N. Kutz. Learning normal form autoencoders for data-driven discovery of universal, parameter-dependent governing equations. *arXiv preprint arXiv:2106.05102*, 2021.
- [379] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [380] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz. Time-delay observables for Koopman: Theory and applications. *SIAM J. Appl. Dyn. Syst.*, 19(2):886–917, 2020.
- [381] A. A. Kaptanoglu, J. L. Callahan, C. J. Hansen, A. Aravkin, and S. L. Brunton. Promoting global stability in data-driven models of quadratic nonlinear dynamics. *arXiv preprint arXiv:2105.01843*, 2021.
- [382] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton. Physics-constrained, low-dimensional models for mhd: First-principles and data-driven approaches. *Physical Review E*, 104(015206), 2021.
- [383] K. Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung, vol. 37. *Annales Academiae Scientiarum Fennicae, Ser. A. I*, 1947.
- [384] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- [385] K. Kasper, L. Mathelin, and H. Abou-Kandil. A machine learning approach for constrained sensor placement. In *American Control Conference (ACC), 2015*, pages 4479–4484. IEEE, 2015.
- [386] A. K. Kassam and L. N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.
- [387] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.
- [388] A. R. Kellems, S. Chaturantabut, D. C. Sorensen, and S. J. Cox. Morphologically accurate reduced order modeling of spiking neurons. *Journal of computational neuroscience*, 28(3):477–494, 2010.
- [389] J. Kepler. *Tabulae Rudolphinae, quibus Astronomicae scientiae, temporum longinquitate collapsae Restauratio continetur*. Ulm: Jonas Saur, 1627.
- [390] G. Kerschen and J.-C. Golinval. Physical interpretation of the proper orthogonal modes using the singular value decomposition. *Journal of Sound and Vibration*, 249(5):849–865, 2002.
- [391] G. Kerschen, J.-c. Golinval, A. F. Vakakis, and L. A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear dynamics*, 41(1-3):147–169, 2005.
- [392] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, C. Theodoropoulos, and others. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.
- [393] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Science*, 1(4):715–762, 2003.
- [394] N. J. Killingsworth and M. Krstic. PID tuning using extremum seeking: online, model-free performance optimization. *IEEE Control Systems Magazine*, February:70–79, 2006.
- [395] H. J. Kim, M. I. Jordan, S. Sastry, and A. Y. Ng. Autonomous helicopter flight via reinforcement learning. In *Advances in neural information processing systems*, pages 799–806, 2004.

- [396] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [397] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [398] M. Kirby and L. Sirovich. Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):103–108, 1990.
- [399] V. C. Klema and A. J. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.
- [400] S. Klus, P. Gelß, S. Peitz, and C. Schütte. Tensor-based dynamic mode decomposition. *Nonlinearity*, 31(7):3359, 2018.
- [401] S. Klus, F. Nüske, and B. Hamzi. Kernel-based approximation of the koopman generator and schrödinger operator. *Entropy*, 22(7):722, 2020.
- [402] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, pages 1–26, 2018.
- [403] S. Klus, I. Schuster, and K. Muandet. Eigendecompositions of transfer operators in reproducing kernel hilbert spaces. *Journal of Nonlinear Science*, 30(1):283–315, 2020.
- [404] J. Kober and J. Peters. Reinforcement learning in robotics: A survey. In *Reinforcement Learning*, pages 579–610. Springer, 2012.
- [405] R. Koch. *The 80/20 Principle*. Nicholas Brealey Publishing, 1997.
- [406] R. Koch. *Living the 80/20 way*. Audio-Tech Business Book Summaries, Incorporated, 2006.
- [407] R. Koch. *The 80/20 principle: the secret to achieving more with less*. Crown Business, 2011.
- [408] R. Koch. *The 80/20 principle and 92 other powerful laws of nature: the science of success*. Nicholas Brealey Publishing, 2013.
- [409] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer. Machine learning accelerated computational fluid dynamics. *arXiv preprint arXiv:2102.01010*, 2021.
- [410] T. Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.
- [411] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.
- [412] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.
- [413] B. O. Koopman and J.-v. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):255, 1932.
- [414] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93(149–160), 2018.
- [415] M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.
- [416] P. Koumoutsakos, J. Freund, and D. Parekh. Evolution strategies for automatic optimization of jet mixing. *AIAA journal*, 39(5):967–969, 2001.
- [417] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.
- [418] K. Kowalski, W.-H. Steeb, and K. Kowalksi. *Nonlinear dynamical systems and Carleman linearization*. World Scientific, 1991.
- [419] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.
- [420] J. R. Koza, F. H. Bennett III, and O. Stiffelman. Genetic programming as a darwinian invention machine. In *Genetic Programming*, pages 93–108. Springer, 1999.

- [421] B. Kramer, P. Grover, P. Boufounos, M. Benosman, and S. Nabi. Sparse sensing and dmd based identification of flow regimes and bifurcations in complex flows. *SIAM J. Appl. Dyn. Syst.*, 16(2):1164–1196, 2017.
- [422] J. P. Krieger and M. Krstic. Extremum seeking based on atmospheric turbulence for aircraft endurance. *Journal of Guidance, Control, and Dynamics*, 34(6):1876–1885, 2011.
- [423] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [424] M. Krstic, A. Krupadanam, and C. Jacobson. Self-tuning control of a nonlinear model of combustion instabilities. *IEEE Tr. Contr. Syst. Technol.*, 7(4):424–436, 1999.
- [425] M. Krstić and H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595–601, 2000.
- [426] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.
- [427] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [428] K. Kunisch and S. Volkwein. Optimal snapshot location for computing pod basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):509–529, 2010.
- [429] J. N. Kutz. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. Oxford University Press, 2013.
- [430] J. N. Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.
- [431] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.
- [432] J. N. Kutz, X. Fu, and S. L. Brunton. Multi-resolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.
- [433] J. N. Kutz, S. Sargsyan, and S. L. Brunton. Leveraging sparsity and compressive sensing for reduced order modeling. In *Model Reduction of Parametrized Systems*, pages 301–315. Springer, 2017.
- [434] S. Lall, J. E. Marsden, and S. Glavaški. Empirical model reduction of controlled nonlinear systems. In *IFAC World Congress*, volume F, pages 473–478. International Federation of Automatic Control, 1999.
- [435] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 12(6):519–535, 2002.
- [436] Y. Lan and I. Mezić. Linearization in the large of nonlinear systems and Koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 242(1):42–53, 2013.
- [437] H. Lange, S. L. Brunton, and J. N. Kutz. From fourier to koopman: Spectral methods for long-term time series prediction. *J. Mach. Learn. Res.*, 22(41):1–38, 2021.
- [438] S. Lanka and T. Wu. Archer: Aggressive rewards to counter bias in hindsight experience replay. *arXiv preprint arXiv:1809.02070*, 2018.
- [439] A. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Transactions on automatic control*, 24(6):913–921, 1979.
- [440] H. Le, C. Voloshin, and Y. Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712. PMLR, 2019.
- [441] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [442] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [443] J. H. Lee. Model predictive control: Review of the three decades of development. *Inter-*

- national Journal of Control, Automation and Systems*, 9(3):415–424, 2011.
- [444] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(5):684–698, 2005.
- [445] A. M. Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.
- [446] V. Lenaerts, G. Kerschen, and J.-C. Golinval. Proper orthogonal decomposition for model updating of non-linear mechanical systems. *Mechanical Systems and Signal Processing*, 15(1):31–43, 2001.
- [447] I. Lenz, R. A. Knepper, and A. Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, 2015.
- [448] R. Leyva, C. Alonso, I. Queinnec, A. Cid-Pastor, D. Lagrange, and L. Martinez-Salamero. MPPT of photovoltaic systems using extremum-seeking control. *Ieee Transactions On Aerospace and Electronic Systems*, 42(1):249–258, Jan. 2006.
- [449] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.
- [450] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [451] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Multipole graph neural operator for parametric partial differential equations. *arXiv preprint arXiv:2006.09535*, 2020.
- [452] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.
- [453] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. Proper orthogonal decomposition and its applications- part i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.
- [454] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 581–588. ACM, 2013.
- [455] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104:20167–20172, 2007.
- [456] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [457] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.
- [458] J. Ling, A. Kurzwski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.
- [459] Y. Liu, J. N. Kutz, and S. L. Brunton. Hierarchical deep learning of multiscale differential equation time-steppers. *arXiv preprint arXiv:2008.09768*, 2020.
- [460] Y. Liu, C. Ponce, S. L. Brunton, and J. N. Kutz. Multiresolution convolutional autoencoders. *arXiv preprint arXiv:2004.04946*, 2020.
- [461] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1999.
- [462] S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*,

- 28(2):129–137, 1982.
- [463] M. Loeve. *Probability Theory*. Van Nostrand, Princeton, NJ, 1955.
- [464] J.-C. Loiseau. Data-driven modeling of the chaotic thermal convection in an annular thermosyphon. *Theoretical and Computational Fluid Dynamics*, 34(4):339–365, 2020.
- [465] J.-C. Loiseau and S. L. Brunton. Constrained sparse Galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.
- [466] J.-C. Loiseau, B. R. Noack, and S. L. Brunton. Sparse reduced-order modeling: sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844:459–490, 2018.
- [467] R. W. Longman. Iterative learning control and repetitive control for engineering practice. *International journal of control*, 73(10):930–954, 2000.
- [468] B. T. Lopez, J.-J. E. Slotine, and J. P. How. Dynamic tube mpc for nonlinear systems. In *2019 American Control Conference (ACC)*, pages 1655–1662. IEEE, 2019.
- [469] E. N. Lorenz. Empirical orthogonal functions and statistical weather prediction. Technical report, Massachusetts Institute of Technology, Dec. 1956.
- [470] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.
- [471] L. Lu, P. Jin, and G. E. Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.
- [472] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.
- [473] D. M. Luchtenburg and C. W. Rowley. Model reduction using snapshot-based realizations. *Bulletin of the American Physical Society*, 56, 2011.
- [474] J. Lumley. Toward a turbulent constitutive relation. *Journal of Fluid Mechanics*, 41(02):413–434, 1970.
- [475] B. Lusch, E. C. Chi, and J. N. Kutz. Shape constrained tensor decompositions using sparse representations in over-complete libraries. *arXiv preprint arXiv:1608.04674*, 2016.
- [476] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.
- [477] F. Lusseyran, F. Gueniat, J. Basley, C. L. Douay, L. R. Pastur, T. M. Faure, and P. J. Schmid. Flow coherent structures and frequency signature: application of the dynamic modes decomposition to open cavity flow. In *Journal of Physics: Conference Series*, volume 318, page 042036. IOP Publishing, 2011.
- [478] J. Lynch, P. Aughwane, and T. M. Hammond. Video games and surgical ability: a literature review. *Journal of surgical education*, 67(3):184–189, 2010.
- [479] Z. Ma, S. Ahuja, and C. W. Rowley. Reduced order models for control of fluids using the eigensystem realization algorithm. *Theor. Comput. Fluid Dyn.*, 25(1):233–247, 2011.
- [480] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.
- [481] A. Mackey, H. Schaeffer, and S. Osher. On the compressive spectral method. *Multiscale Modeling & Simulation*, 12(4):1800–1827, 2014.
- [482] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3:123–224, 2011.
- [483] A. J. Majda and J. Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1):201, 2012.
- [484] A. J. Majda and Y. Lee. Conceptual dynamical models for turbulence. *Proceedings of the National Academy of Sciences*, 111(18):6548–6553, 2014.

- [485] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.
- [486] S. Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065):20150203, 2016.
- [487] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.
- [488] J. Mandel. Use of the singular value decomposition in regression analysis. *The American Statistician*, 36(1):15–24, 1982.
- [489] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 2(1):52–63, 2016.
- [490] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A*, 473(2204):1–16, 2017.
- [491] J. Mann and J. N. Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, pages 1–13, 2016.
- [492] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton. Data-driven sparse sensor placement. *Invited for IEEE Control Systems Magazine*, 2017.
- [493] K. Manohar, S. L. Brunton, and J. N. Kutz. Environmental identification in flight using sparse approximation of wing strain. *Journal of Fluids and Structures*, 70:162–180, 2017.
- [494] K. Manohar, E. Kaiser, S. L. Brunton, and J. N. Kutz. Optimized sampling for multiscale dynamics. *SIAM Multiscale modeling and simulation*, 17(1):117–136, 2019.
- [495] K. Manohar, J. N. Kutz, and S. L. Brunton. Optimized sensor and actuator placement for balanced models. *arXiv preprint arXiv:1812.01574*, 2018.
- [496] A. Mardt, L. Pasquali, H. Wu, and F. Noé. VAMPnets: Deep learning of molecular kinetics. *Nature Communications*, 9(5), 2018.
- [497] J. E. Marsden and T. S. Ratiu. *Introduction to mechanics and symmetry*. Springer-Verlag, 2nd edition, 1999.
- [498] P.-G. Martinsson. Randomized methods for matrix computations and analysis of high dimensional data. *arXiv preprint arXiv:1607.01649*, 2016.
- [499] P.-G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30:47–68, 2011.
- [500] J. L. Maryak, J. C. Spall, and B. D. Heydon. Use of the Kalman filter for inference in state-space models with unknown noise distributions. *IEEE Transactions on Automatic Control*, 49(1):87–90, 2004.
- [501] L. Massa, R. Kumar, and P. Ravindran. Dynamic mode decomposition analysis of detonation waves. *Physics of Fluids (1994-present)*, 24(6):066101, 2012.
- [502] L. Mathelin, K. Kasper, and H. Abou-Kandil. Observable dictionary learning for high-dimensional statistical inference. *Archives of Computational Methods in Engineering*, 25(1):103–120, 2018.
- [503] R. Maulik, O. San, A. Rasheed, and P. Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858:122–144, 2019.
- [504] R. Maury, M. Keonig, L. Cattafesta, P. Jordan, and J. Delville. Extremum-seeking control of jet noise. *Aeroacoustics*, 11(3&4):459–474, 2012.
- [505] S. F. McCormick. *Multigrid methods*. SIAM, 1987.
- [506] B. J. McKeon and A. S. Sharma. A critical-layer framework for turbulent pipe flow. *J. Fluid Mech.*, 658:336–382, 2010.
- [507] X. Meng, Z. Li, D. Zhang, and G. E. Karniadakis. Ppinn: Parareal physics-informed neural network for time-dependent pdes. *Computer Methods in Applied Mechanics and Engineering*, 370:113250, 2020.

- [508] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
- [509] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Ann. Rev. Fluid Mech.*, 45:357–378, 2013.
- [510] I. Mezić. *Spectral operator methods in dynamical systems: Theory and applications*. Springer, 2017.
- [511] I. Mezić and A. Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1):101–133, 2004.
- [512] I. Mezić and S. Wiggins. A method for visualization of invariant sets of dynamical systems based on the ergodic partition. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 9(1):213–218, 1999.
- [513] M. Milano and P. Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.
- [514] M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.
- [515] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [516] Y. Mizuno, D. Duke, C. Atkinson, and J. Soria. Investigation of wall-bounded turbulent flow using dynamic mode decomposition. In *Journal of Physics: Conference Series*, volume 318, page 042040. IOP Publishing, 2011.
- [517] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.
- [518] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [519] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [520] J. P. Moeck, J.-F. Bourgouin, D. Durox, T. Schuller, and S. Candel. Tomographic reconstruction of heat release rate perturbations induced by helical modes in turbulent swirl flames. *Experiments in Fluids*, 54(4):1–17, 2013.
- [521] P. R. Montague, P. Dayan, and T. J. Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *Journal of neuroscience*, 16(5):1936–1947, 1996.
- [522] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26(1):17–32, 1981.
- [523] C. C. Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences*, 112(7):1907–1911, 2015.
- [524] K. L. Moore. *Iterative learning control for deterministic systems*. Springer Science & Business Media, 2012.
- [525] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.
- [526] J. Morton, F. D. Witherden, A. Jameson, and M. J. Kochenderfer. Deep dynamical modeling and control of unsteady fluid flows. *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*; *arXiv preprint arXiv:1805.07472*, 2018.
- [527] T. W. Muld, G. Efraimsson, and D. S. Henningson. Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. *Computers & Fluids*, 57:87–97, 2012.
- [528] T. W. Muld, G. Efraimsson, and D. S. Henningson. Mode decomposition on surface-mounted cube. *Flow, Turbulence and Combustion*, 88(3):279–310, 2012.

- [529] S. Müller, M. Milano, and P. Koumoutsakos. Application of machine learning algorithms to flow modeling and optimization. *Annual Research Briefs*, pages 169–178, 1999.
- [530] I. Munteanu, A. I. Bratcu, and E. Ceanga. Wind turbulence used as searching signal for MPPT in variable-speed wind energy conversion systems. *Renewable Energy*, 34(1):322–327, Jan. 2009.
- [531] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [532] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [533] D. Needell and J. A. Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, 2010.
- [534] J. v. Neumann. Proof of the quasi-ergodic hypothesis. *Proceedings of the National Academy of Sciences*, 18(1):70–82, 1932.
- [535] N. Nguyen, A. Patera, and J. Peraire. A best points interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4):521–543, 2008.
- [536] Y. Nievergelt and Y. Nievergelt. *Wavelets made easy*, volume 174. Springer, 1999.
- [537] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [538] B. R. Noack, T. Duriez, L. Cordier, M. Segond, M. Abel, S. L. Brunton, M. Morzyński, J.-C. Larentie, V. Parezanovic, and J.-P. Bonnet. Closed-loop turbulence control with machine learning methods. *Bulletin Am. Phys. Soc.*, 58(18):M25.0009, p. 418, 2013.
- [539] B. R. Noack, M. Morzynski, and G. Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer Science & Business Media, 2011.
- [540] B. R. Noack, W. Stankiewicz, M. Morzynski, and P. J. Schmid. Recursive dynamic mode decomposition of a transient cylinder wake. *Journal of Fluid Mechanics*, 809:843–872, 2016.
- [541] F. Noé and F. Nuske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.
- [542] E. Noether. Invariante variationsprobleme nachr. d. könig. gesellsch. d. wiss. zu göttingen, math-phys. klasse 1918: 235-257. *English Reprint: physics/0503066*, <http://dx.doi.org/10.1080/00411457108231446>, page 57, 1918.
- [543] G. Novati, H. L. de Laroussilhe, and P. Koumoutsakos. Automating turbulence modelling by multi-agent reinforcement learning. *Nature Machine Intelligence*, 3(1):87–96, 2021.
- [544] G. Novati, L. Mahadevan, and P. Koumoutsakos. Controlled gliding and perching through deep-reinforcement-learning. *Physical Review Fluids*, 4(9):093902, 2019.
- [545] G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. M. Van Rees, and P. Koumoutsakos. Synchronisation through learning for two self-propelled swimmers. *Bioinspiration Biomim.*, 12(3):aa6311, 2017.
- [546] F. Nüske, P. Gelß, S. Klus, and C. Clementi. Tensorbased edmd for the koopman analysis of high-dimensional systems. *arXiv preprint arXiv:1908.04741*, 2019.
- [547] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. Mey, and F. Noé. Variational approach to molecular kinetics. *Journal of chemical theory and computation*, 10(4):1739–1752, 2014.
- [548] F. Nüske, R. Schneider, F. Vitalini, and F. Noé. Variational tensor approach for approximating the rare-event kinetics of macromolecular systems. *J. Chem. Phys.*, 144(5):054105, 2016.
- [549] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the A. I. E. E.*, pages 617–644, FEB 1928.

- [550] G. Obinata and B. D. Anderson. *Model reduction for control system design*. Springer Science & Business Media, 2012.
- [551] M. Ornik, A. Israel, and U. Topcu. Control-oriented learning on the fly. *arXiv preprint arXiv:1709.04889*, 2017.
- [552] C. M. Ostoich, D. J. Bodony, and P. H. Geubelle. Interaction of a Mach 2.25 turbulent boundary layer with a fluttering panel using direct numerical simulation. *Physics of Fluids (1994-present)*, 25(11):110806, 2013.
- [553] S. E. Otto and C. W. Rowley. Linearly-recurrent autoencoder networks for learning dynamics. *arXiv preprint arXiv:1712.01378*, 2017.
- [554] Y. Ou, C. Xu, E. Schuster, T. C. Luce, J. R. Ferron, M. L. Walker, and D. A. Humphreys. Design and simulation of extremum-seeking open-loop optimal control of current profile in the DIII-D tokamak. *Plasma Physics and Controlled Fusion*, 50:115001–1–115001–24, 2008.
- [555] V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.
- [556] C. Pan, D. Yu, and J. Wang. Dynamical mode decomposition of Gurney flap wake flow. *Theoretical and Applied Mechanics Letters*, 1(1):012002, 2011.
- [557] S. Pan and K. Duraisamy. Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, 19(1):480–509, 2020.
- [558] X. Pan, Y. You, Z. Wang, and C. Lu. Virtual to real reinforcement learning for autonomous driving. *arXiv preprint arXiv:1704.03952*, 2017.
- [559] V. Parezanović, T. Duriez, L. Cordier, B. R. Noack, J. Delville, J.-P. Bonnet, M. Segond, M. Abel, and S. L. Brunton. Closed-loop control of an experimental mixing layer using machine learning control. *arXiv preprint arXiv:1408.3259*, 2014.
- [560] V. Parezanovic, J.-C. Laurentie, T. Duriez, C. Fourment, J. Delville, J.-P. Bonnet, L. Cordier, B. R. Noack, M. Segond, M. Abel, T. Shaqarin, and S. L. Brunton. Mixing layer manipulation experiment – from periodic forcing to machine learning closed-loop control. *Journal Flow Turbulence and Combustion*, 94(1):155–173, 2015.
- [561] E. J. Parish and K. T. Carlberg. Time-series machine-learning error models for approximate solutions to parameterized dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 365:112990, 2020.
- [562] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.
- [563] P. I. Pavlov. *Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex*. Oxford University Press, 1927.
- [564] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(7–12):559–572, 1901.
- [565] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.
- [566] B. Peherstorfer, Z. Drmac, and S. Gugercin. Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points. *SIAM Journal on Scientific Computing*, 42(5):A2837–A2864, 2020.
- [567] B. Peherstorfer and K. Willcox. Detecting and adapting to parameter changes for reduced models of dynamic data-driven application systems. *Procedia Computer Science*, 51:2553–2562, 2015.
- [568] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.

- [569] B. Peherstorfer and K. Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015.
- [570] S. Peitz and S. Klus. Koopman operator-based model reduction for switched-system control of PDEs. *arXiv preprint arXiv:1710.06759*, 2017.
- [571] S. D. Pendergrass, J. N. Kutz, and S. L. Brunton. Streaming GPU singular value and dynamic mode decompositions. *arXiv preprint arXiv:1612.07875*, 2016.
- [572] R. Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge Univ Press, 1955.
- [573] R. Penrose and J. A. Todd. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge Univ Press, 1956.
- [574] L. Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2013.
- [575] M. Phan, L. G. Horta, J. N. Juang, and R. W. Longman. Linear system identification via an asymptotically stable observer. *Journal of Optimization Theory and Applications*, 79:59–86, 1993.
- [576] M. A. Pinsky. *Introduction to Fourier analysis and wavelets*, volume 102. American Mathematical Soc., 2002.
- [577] C. Pivot, L. Mathelin, L. Cordier, F. Guéniat, and B. R. Noack. A continuous reinforcement learning strategy for closed-loop control in fluid dynamics. In *35th AIAA Applied Aerodynamics Conference*, page 3566, 2017.
- [578] T. Poggio. Deep learning: mathematics and neuroscience. *Views & Reviews, McGovern Center for Brains, Minds and Machines*, pages 1–7, 2016.
- [579] P. Poncet, G.-H. Cottet, and P. Koumoutsakos. Control of three-dimensional wakes using evolution strategies. *Comptes Rendus Mécanique*, 333(1):65–77, 2005.
- [580] C. Poultney, S. Chopra, Y. L. Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2007.
- [581] J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. N. Kutz. Exploiting sparsity and equation-free architectures in complex systems (invited review). *The European Physical Journal Special Topics*, 223(13):2665–2684, 2014.
- [582] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.
- [583] J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.
- [584] H. Qi and S. M. Hughes. Invariance of principal components under low-dimensional random projection of the data. *IEEE International Conference on Image Processing*, October 2012.
- [585] S. Qian and D. Chen. Discrete Gabor transform. *IEEE transactions on signal processing*, 41(7):2429–2438, 1993.
- [586] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In *AIChE Symposium Series*, volume 93, pages 232–256, 1997.
- [587] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.
- [588] T. Qin, K. Wu, and D. Xiu. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019.
- [589] Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems 27*, pages 3401–3409, 2014.

- [590] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.
- [591] A. Quarteroni and G. Rozza. *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *MS&A – Modeling, Simulation & Applications*. Springer, 2013.
- [592] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
- [593] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
- [594] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of fluid mechanics*, 865:281–302, 2019.
- [595] J. Rabault and A. Kuhnle. Deep reinforcement learning applied to active flow control. 2020.
- [596] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.
- [597] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [598] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.
- [599] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems*, 20(3):38–52, 2000.
- [600] S. Raychaudhuri, J. M. Stuart, and R. B. Altman. Principal components analysis to summarize microarray experiments: application to sporulation time series. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 455. NIH Public Access, 2000.
- [601] B. Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279, 2019.
- [602] G. Reddy, A. Celani, T. J. Sejnowski, and M. Vergassola. Learning to soar in turbulent environments. *Proc. Natl. Acad. Sci. USA*, 113(33):E4877–E4884, 2016.
- [603] G. Reddy, J. Wong-Ng, A. Celani, T. J. Sejnowski, and M. Vergassola. Glider soaring via reinforcement learning in the field. *Nature*, 562(7726):236–239, 2018.
- [604] S. Reddy, A. D. Dragan, and S. Levine. Shared autonomy via deep reinforcement learning. *arXiv preprint arXiv:1802.01744*, 2018.
- [605] A. D. Redish. Addiction as a computational process gone awry. *Science*, 306(5703):1944–1947, 2004.
- [606] W. T. Redman. On koopman mode decomposition and tensor component analysis. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(5):051101, 2021.
- [607] F. Regazzoni, L. Dede, and A. Quarteroni. Machine learning for fast and reliable solution of time-dependent differential equations. *Journal of Computational physics*, 397:108852, 2019.
- [608] R. H. Reichle, D. B. McLaughlin, and D. Entekhabi. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review*, 130(1):103–114, 2002.
- [609] P. A. Reinbold, D. R. Gurevich, and R. O. Grigoriev. Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Physical Review E*, 101(1):010203, 2020.
- [610] P. A. Reinbold, L. M. Kageorge, M. F. Schatz, and R. O. Grigoriev. Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nature communications*, 12(1):1–8, 2021.
- [611] B. Ren, P. Frihauf, R. J. Rafac, and M. Krstić. Laser pulse shaping via extremum seeking. *Control Engineering Practice*, 20:674–683, 2012.
- [612] A. Richards and J. How. Decentralized model predictive control of cooperating uavs.

- In *2004 43rd IEEE Conference on Decision and Control (CDC)*(IEEE Cat. No. 04CH37601), volume 4, pages 4286–4291. IEEE, 2004.
- [613] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of control*, 80(9):1517–1531, 2007.
- [614] B. Ristic, S. Arulampalam, and N. J. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2004.
- [615] A. J. Roberts. *Model emergent dynamics in complex systems*. SIAM, 2014.
- [616] C. A. Rohde. Generalized inverses of partitioned matrices. *Journal of the Society for Industrial & Applied Mathematics*, 13(4):1033–1035, 1965.
- [617] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31:1100–1124, 2009.
- [618] S. M. Ross. *Introduction to stochastic dynamic programming*. Academic press, 2014.
- [619] J. C. Rosser, P. J. Lynch, L. Cuddihy, D. A. Gentile, J. Klonsky, and R. Merrell. The impact of video games on training surgeons in the 21st century. *Archives of surgery*, 142(2):181–186, 2007.
- [620] C. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *Int. J. Bifurcation and Chaos*, 15(3):997–1013, 2005.
- [621] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D*, 189:115–129, 2004.
- [622] C. W. Rowley and J. E. Marsden. Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry. *Physica D: Nonlinear Phenomena*, 142(1):1–19, 2000.
- [623] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 645:115–127, 2009.
- [624] S. Roy, J.-C. Hua, W. Barnhill, G. H. Gunaratne, and J. R. Gord. Deconvolution of reacting-flow dynamics using proper orthogonal and dynamic mode decompositions. *Physical Review E*, 91(1):013001, 2015.
- [625] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(e1602614), 2017.
- [626] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [627] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017.
- [628] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.
- [629] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020.
- [630] T. P. Sapsis and A. J. Majda. Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. *Proceedings of the National Academy of Sciences*, 110(34):13705–13710, 2013.
- [631] S. Sargsyan, S. L. Brunton, and J. N. Kutz. Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries. *Physical Review E*, 92(033304), 2015.
- [632] S. Sarkar, S. Ganguly, A. Dalal, P. Saha, and S. Chakraborty. Mixed convective flow stability of nanofluids past a square cylinder by dynamic mode decomposition. *International Journal of Heat and Fluid Flow*, 44:624–634, 2013.
- [633] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science. 47th Annual IEEE Symposium on*, pages 143–152, 2006.
- [634] D. Sashidhar and J. N. Kutz. Bagging, optimized dynamic mode decomposition (bop-

- dmd) for robust, stable forecasting with spatial and temporal uncertainty-quantification. *arXiv preprint arXiv:2107.10878*, 2021.
- [635] T. Sayadi and P. J. Schmid. Parallel data-driven decomposition algorithm for large-scale datasets: with application to transitional boundary layers. *Theoretical and Computational Fluid Dynamics*, pages 1–14, 2016.
- [636] T. Sayadi, P. J. Schmid, J. W. Nichols, and P. Moin. Reduced-order representation of near-wall structures in the late transitional boundary layer. *Journal of Fluid Mechanics*, 748:278–301, 2014.
- [637] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.
- [638] H. Schaeffer. Learning partial differential equations via data discovery and sparse optimization. In *Proc. R. Soc. A*, volume 473, page 20160446. The Royal Society, 2017.
- [639] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences USA*, 110(17):6634–6639, 2013.
- [640] H. Schaeffer and S. G. McCalla. Sparse model selection via integral terms. *Physical Review E*, 96(2):023302, 2017.
- [641] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.
- [642] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- [643] I. Scherl, B. Strom, J. K. Shang, O. Williams, B. L. Polagye, and S. L. Brunton. Robust principal component analysis for particle image velocimetry. *Physical Review Fluids*, 5(054401), 2020.
- [644] M. Schlegel and B. R. Noack. On long-term boundedness of galerkin models. *Journal of Fluid Mechanics*, 765:325–352, 2015.
- [645] M. Schlegel, B. R. Noack, and G. Tadmor. Low-dimensional Galerkin models and control of transitional channel flow. Technical Report 01/2004, Hermann-Föttinger-Institut für Strömungsmechanik, Technische Universität Berlin, Germany, 2004.
- [646] M. Schmelzer, R. P. Dwight, and P. Cinnella. Discovery of algebraic reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion*, 104(2):579–603, 2020.
- [647] P. J. Schmid. Dynamic mode decomposition for numerical and experimental data. *J. Fluid. Mech*, 656:5–28, 2010.
- [648] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.
- [649] P. J. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, Nov. 2008.
- [650] P. J. Schmid, D. Violato, and F. Scarano. Decomposition of time-resolved tomographic PIV. *Experiments in Fluids*, 52:1567–1579, 2012.
- [651] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. i teil. entwicklung willkürlichen funktionen nach system vorgeschriebener. *Math. Ann.*, 3:433–476, 1907.
- [652] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.
- [653] M. D. Schmidt, R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson. Automated refinement and inference of analytical models for metabolic networks. *Physical biology*, 8(5):055011, 2011.
- [654] O. T. Schmidt and T. Colonius. Guide to spectral proper orthogonal decomposition. *Aiaa*

- journal*, 58(3):1023–1033, 2020.
- [655] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [656] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.
- [657] W. Schultz, P. Dayan, and P. R. Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.
- [658] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [659] A. Seena and H. J. Sung. Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. *International Journal of Heat and Fluid Flow*, 32(6):1098–1110, 2011.
- [660] E. Sejdić, I. Djurović, and J. Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, 19(1):153–183, 2009.
- [661] O. Semeraro, G. Bellani, and F. Lundell. Analysis of time-resolved PIV measurements of a confined turbulent jet using POD and Koopman modes. *Experiments in Fluids*, 53(5):1203–1220, 2012.
- [662] O. Semeraro, F. Lusseyran, L. Pastur, and P. Jordan. Qualitative dynamics of wavepackets in turbulent jets. *Physical Review Fluids*, 2(094605), 2017.
- [663] G. Shabat, Y. Shmueli, Y. Aizenbud, and A. Averbuch. Randomized LU decomposition. *Applied and Computational Harmonic Analysis*, 2016.
- [664] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.
- [665] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
- [666] C. E. Shannon. Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.
- [667] A. S. Sharma, I. Mezić, and B. J. McKeon. Correspondence between Koopman mode decomposition, resolvent mode decomposition, and invariant solutions of the Navier-Stokes equations. *Physical Review Fluids*, 1(3):032402, 2016.
- [668] D. E. Shea, S. L. Brunton, and J. N. Kutz. Sindy-bvp: Sparse identification of nonlinear dynamics for boundary value problems. *Physical Review Research*, 3(2):023255, 2021.
- [669] E. Shlizerman, E. Ding, M. O. Williams, and J. N. Kutz. The proper orthogonal decomposition for dimensionality reduction in mode-locked lasers and optical systems. *International Journal of Optics*, 2012, 2011.
- [670] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- [671] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [672] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.
- [673] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.
- [674] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations.

- SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.
- [675] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I-III. *Q. Appl. Math.*, XLV(3):561–590, 1987.
- [676] L. Sirovich and M. Kirby. A low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987.
- [677] S. Skogestad and I. Postlethwaite. *Multivariable feedback control*. Wiley, Chichester, 1996.
- [678] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.
- [679] G. Solari, L. Carassale, and F. Tubino. Proper orthogonal decomposition in wind engineering. part 1: A state-of-the-art and some prospects. *Wind and Structures*, 10(2):153–176, 2007.
- [680] G. Song, F. Alizard, J.-C. Robinet, and X. Gloerfelt. Global and Koopman modes analysis of sound generation in mixing layers. *Physics of Fluids (1994-present)*, 25(12):124101, 2013.
- [681] D. C. Sorensen and Y. Zhou. Direct methods for matrix Sylvester and Lyapunov equations. *Journal of Applied Mathematics*, 2003(6):277–303, 2003.
- [682] M. Sorokina, S. Sygletos, and S. Turitsyn. Sparse identification for nonlinear optical communication systems: SINO method. *Optics express*, 24(26):30433–30443, 2016.
- [683] J. C. Spall. The Kantorovich inequality for error analysis of the Kalman filter with unknown noise distributions. *Automatica*, 31(10):1513–1517, 1995.
- [684] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [685] I. Stakgold and M. J. Holst. *Green's functions and boundary value problems*, volume 99. John Wiley & Sons, 2011.
- [686] W.-H. Steeb and F. Wilhelm. Non-linear autonomous systems of differential equations and Carleman linearization procedure. *Journal of Mathematical Analysis and Applications*, 77(2):601–611, 1980.
- [687] R. F. Stengel. *Optimal control and estimation*. Courier Corporation, 2012.
- [688] G. W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.
- [689] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch. Detecting causality in complex ecosystems. *Science*, 338(6106):496–500, 2012.
- [690] C. Sun, E. Kaiser, S. L. Brunton, and J. N. Kutz. Deep reinforcement learning for optical systems: A case study of mode-locked lasers. *Machine Learning: Science and Technology*, 1(4):045013, 2020.
- [691] A. Surana. Koopman operator based observer synthesis for control-affine nonlinear systems. In *55th IEEE Conference on Decision and Control (CDC)*, pages 6492–6499, 2016.
- [692] A. Surana and A. Banaszuk. Linear observer synthesis for nonlinear systems using Koopman operator framework. *IFAC-PapersOnLine*, 49(18):716–723, 2016.
- [693] Y. Susuki and I. Mezić. A prony approximation of Koopman mode decomposition. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 7022–7027. IEEE, 2015.
- [694] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.
- [695] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [696] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information*

- processing systems*, pages 1057–1063, 2000.
- [697] A. Svenkeson, B. Glaz, S. Stanton, and B. J. West. Spectral decomposition of nonlinear systems with memory. *Phys. Rev. E*, 93:022211, Feb 2016.
- [698] S. Svoronos, D. Papageorgiou, and C. Tsiligiannis. Discretization of nonlinear control systems via the Carleman linearization. *Chemical engineering science*, 49(19):3263–3267, 1994.
- [699] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(8):831–836, 1996.
- [700] K. Taira, S. L. Brunton, S. T. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley. Modal analysis of fluid flows: An overview. *Aiaa Journal*, 55(12):4013–4041, 2017.
- [701] K. Taira and T. Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.
- [702] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. Dawson, and C.-A. Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA Journal*, 58(3):998–1022, 2020.
- [703] N. Takeishi, Y. Kawahara, Y. Tabei, and T. Yairi. Bayesian dynamic mode decomposition. *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.
- [704] N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.
- [705] N. Takeishi, Y. Kawahara, and T. Yairi. Subspace dynamic mode decomposition for stochastic Koopman analysis. *Physical Review E*, 96(033310), 2017.
- [706] F. Takens. Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*, 898:366–381, 1981.
- [707] Z. Q. Tang and N. Jiang. Dynamic mode decomposition of hairpin vortices generated by a hemisphere protuberance. *Science China Physics, Mechanics and Astronomy*, 55(1):118–124, 2012.
- [708] A. Taylor, A. Singletary, Y. Yue, and A. Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control*, pages 708–717. PMLR, 2020.
- [709] R. Taylor, J. N. Kutz, K. Morgan, and B. Nelson. Dynamic mode decomposition for plasma diagnostics and validation. *Review of Scientific Instruments*, 89(053501), 2018.
- [710] R. Tedrake, Z. Jackowski, R. Cory, J. W. Roberts, and W. Hoburg. Learning to fly like a bird. In *14th International Symposium on Robotics Research. Lucerne, Switzerland*, 2009.
- [711] G. Tesauro. Practical issues in temporal difference learning. *Machine learning*, 8(3):257–277, 1992.
- [712] G. Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.
- [713] S. Thaler, L. Paehler, and N. A. Adams. Sparse identification of truncation errors. *Journal of Computational Physics*, 397:108851, 2019.
- [714] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- [715] Z. Ting and J. Hui. Eeg signal processing based on proper orthogonal decomposition. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pages 636–640. IEEE, 2012.
- [716] S. Tirunagari, N. Poh, K. Wells, M. Bober, I. Gorden, and D. Windridge. Movement correction in DCE-MRI through windowed and reconstruction dynamic mode decomposition. *Machine Vision and Applications*, 28(3-4):393–407, 2017.

- [717] J. Tithof, B. Suri, R. K. Pallantla, R. O. Grigoriev, and M. F. Schatz. Bifurcations in a quasi-two-dimensional kolmogorov-like flow. *Journal of Fluid Mechanics*, 828:837–866, 2017.
- [718] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.
- [719] C. Torrence and G. P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.
- [720] A. Towne, O. T. Schmidt, and T. Colonius. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *Journal of Fluid Mechanics*, 847:821–867, 2018.
- [721] G. Tran and R. Ward. Exact recovery of chaotic systems from highly corrupted data. *SIAM Multiscale modeling and simulation*, 15(3):1108–1129, 2017.
- [722] L. N. Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.
- [723] L. N. Trefethen and D. Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.
- [724] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll. Hydrodynamic stability without eigenvalues. *Science*, 261(5121):578–584, 1993.
- [725] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.
- [726] J. A. Tropp. Recovery of short, complex linear combinations via l_1 minimization. *IEEE Transactions on Information Theory*, 51(4):1568–1570, 2005.
- [727] J. A. Tropp. Algorithms for simultaneous sparse approximation. part ii: Convex relaxation. *Signal Processing*, 86(3):589–602, 2006.
- [728] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.
- [729] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.
- [730] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.
- [731] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *IEEE Transactions on Information Theory*, 56(1):520–544, 2010.
- [732] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Randomized single-view algorithms for low-rank matrix approximation. *arXiv preprint arXiv:1609.00048*, 2016.
- [733] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Elsevier, 2000.
- [734] J. N. Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.
- [735] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.
- [736] J. H. Tu and C. W. Rowley. An improved algorithm for balanced POD through an analytic treatment of impulse response tails. *J. Comp. Phys.*, 231(16):5317–5333, 2012.
- [737] J. H. Tu, C. W. Rowley, E. Aram, and R. Mittal. Koopman spectral analysis of separated flow over a finite-thickness flat plate with elliptical leading edge. *AIAA Paper 2011*, 2864, 2011.
- [738] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang. Spectral analysis of fluid flows using sub-Nyquist-rate PIV data. *Experiments in Fluids*, 55(9):1–13, 2014.
- [739] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *J. Comp. Dyn.*, 1(2):391–421, 2014.
- [740] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

- [741] R. Van Der Merwe. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. 2004.
- [742] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.
- [743] C. Van Loan. *Computational frameworks for the fast Fourier transform*. SIAM, 1992.
- [744] D. Venturi and G. E. Karniadakis. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics*, 519:315–336, 2004.
- [745] S. Verma, G. Novati, and P. Koumoutsakos. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences*, 115(23):5849–5854, 2018.
- [746] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.
- [747] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- [748] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz*. see <http://www.uni-graz.at/ima/www/volkwein/POD.pdf>, 1025, 2011.
- [749] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 4:4, 2013.
- [750] S. Voronin and P.-G. Martinsson. RSVDPACK: Subroutines for computing partial singular value decompositions via randomized sampling on single core, multi core, and GPU architectures. *arXiv preprint arXiv:1502.05366*, 2015.
- [751] A. Wang et al. An industrial strength audio search algorithm. In *Ismir*, volume 2003, pages 7–13. Washington, DC, 2003.
- [752] H. H. Wang, M. Krstić, and G. Bastin. Optimizing bioreactors by extremum seeking. *Adaptive Control and Signal Processing*, 13(8):651–669, 1999.
- [753] H. H. Wang, S. Yeung, and M. Krstić. Experimental application of extremum seeking on an axial-flow compressor. *IEEE Transactions on Control Systems Technology*, 8(2):300–309, 2000.
- [754] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and C. Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*, 106:154101–1–154101–4, 2011.
- [755] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237:10–26, 2012.
- [756] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.
- [757] C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.
- [758] C. Wehmeyer and F. Noé. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of Chemical Physics*, 148(241703), 2018.
- [759] E. Weinan. *Principles of multiscale modeling*. Cambridge University Press, 2011.
- [760] E. Weinan, B. Engquist, and others. The heterogeneous multiscale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.
- [761] G. Welch and G. Bishop. An introduction to the Kalman filter, 1995.
- [762] P. Whittle. *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells, 1951.
- [763] O. Wiederhold, R. King, B. R. Noack, L. Neuhaus, L. Neise, W. an Enghard, and M. Swo-

- boda. Extensions of extremum-seeking control to improve the aerodynamic performance of axial turbomachines. In *39th AIAA Fluid Dynamics Conference*, pages 1–19, San Antonio, TX, USA, 2009. AIAA-Paper 092407.
- [764] S. Wiggins, S. Wiggins, and M. Golubitsky. *Introduction to applied nonlinear dynamical systems and chaos*, volume 2. Springer, 1990.
- [765] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.
- [766] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.
- [767] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1714–1721. IEEE, 2017.
- [768] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *Journal of Nonlinear Science*, 6:1307–1346, 2015.
- [769] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel approach to data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015.
- [770] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.
- [771] D. M. Witten and R. Tibshirani. Penalized classification using Fisher’s linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772, 2011.
- [772] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Journal of Applied and Computational Harmonic Analysis*, 25:335–366, 2008.
- [773] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2009.
- [774] C. J. Wu. On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103, 1983.
- [775] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.
- [776] H. Ye, R. J. Beamish, S. M. Glaser, S. C. Grant, C.-h. Hsieh, L. J. Richards, J. T. Schnute, and G. Sugihara. Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. *Proceedings of the National Academy of Sciences*, 112(13):E1569–E1576, 2015.
- [777] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. *arXiv preprint arXiv:1708.06850*, 2017.
- [778] B. Yildirim, C. Chrysostomidis, and G. Karniadakis. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Modelling*, 27(3):160–173, 2009.
- [779] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *preprint*, 12, 2009.
- [780] I. Zamora, N. G. Lopez, V. M. Vilches, and A. H. Cordero. Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo. *arXiv preprint arXiv:1608.05742*, 2016.
- [781] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In

- IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535, 2010.
- [782] C. Zhang and R. O. nez. Numerical optimization-based extremum seeking control with application to ABS design. *IEEE Transactions on Automatic Control*, 52(3):454–467, 2007.
- [783] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. Online dynamic mode decomposition for time-varying systems. *arXiv preprint arXiv:1707.02876*, 2017.
- [784] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In *IEEE Robotics and Automation (ICRA)*, pages 528–535, 2016.
- [785] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 528–535. IEEE, 2016.
- [786] W. Zhang, B. Wang, Z. Ye, and J. Quan. Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models. *AIAA journal*, 50(5):1019–1028, 2012.
- [787] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin. Sparse relaxed regularized regression: SR3. *IEEE Access*, 7(1):1404–1423, 2019.
- [788] S. Zlobec. An explicit form of the moore-penrose inverse of an arbitrary complex matrix. *SIAM Review*, 12(1):132–134, 1970.
- [789] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.