Reinforcement Learning-Based Techniques for the Optimal Control of Complex Systems

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING/INF-04 - AUTOMATIC CONTROLS

Final Dissertation

Reinforcement Learning-Based Techniques for the Optimal Control of Complex Systems

by
Paolo Roberto Massenio

Supervisor:

Prof. David Naso

Coordinator of Ph.D Program:
Prof. Luigi Alfredo Grieco

XXXIII Cycle - November 1st, 2017 - December 31st, 2020

Department of Electrical and Information Engineering

ELECTRICAL AND INFORMATION ENGINEERING

Ph.D. Program

SSD: ING/INF-04 - AUTOMATIC CONTROLS

Final Dissertation

Reinforcement Learning-Based Techniques for the Optimal Control of Complex Systems

by

Paolo Roberto Massenio

Supervisor:

Prof. David Naso

Coordinator of Ph.D Program:

Prof. Luigi Alfredo Grieco

XXXIII Cycle - November 1st, 2017 - December 31st, 2020

Abstract

This doctoral thesis presents the results of the three-years activities carried out during the XXXIII cycle of the Ph.D. program in Electrical and Information Engineering of the Polytechnic University of Bari, Bari, Italy. The topic of this thesis is the optimal control of complex systems using Reinforcement Learning (RL) based techniques. Optimal control theory is aimed at finding control policies that minimize a predefined closed-loop performance criterion, namely the utility function. While optimal control for linear systems is a well-established framework, several issues arise when nonlinearities come into the picture. Feedback optimal control policies for nonlinear systems are found by solving the Hamilton-Jacobi-Bellman (HJB) equation, which is in general analytically intractable. Starting from the 1980s, considerable efforts have been made by the research community to overcome such intractability. This resulted in the development of new approaches based on RL that find approximated solutions of the HJB equation using Neural Networks (NNs). RL is an important branch of the Machine Learning theory. It is inspired by the animal world where living beings improve their behaviors by interacting with an unknown environment, evaluating the effect of their actions and modifying them accordingly. The combination of RL paradigms, NNs, and optimal control results in the Adaptive Dynamic Programming (ADP) approach. ADP algorithms find optimal control laws by means of different learning strategies. Such approach demonstrates the increasing penetration of Artificial Intelligence (AI) in the field of complex control systems.

The main purpose of this thesis is to show the effectiveness of ADP-based control systems in real-world scenarios. In fact, although most of the ADP theory has been developed since the second half of the 2000s, experimental tests of real-world ADP-based controllers have only been published more recently. This thesis begins by over-viewing the main ADP algorithms that solve optimal control problems for nonlinear systems, covering the two main learning strategies: the Policy Iteration (PI) algorithm with on-policy learning and the PI algorithm with off-policy learning. The mathematical details of such approaches are presented, discussing the main properties along with pros and cons. Then, the powerful features of the ADP algorithm with off-policy learning are exploited to provide novel control strategies according to two different complex systems. It will be shown how the versatility and power of ADP-based techniques allow to solve control problems with different contexts and objectives in an innovative way.

As first case study, the optimal control of mechatronic devices based on dielectric elastomer membranes, namely the Dielectric Elastomer Actuators (DEAs), is considered. A DEA is typically constituted by a flexible polymeric membrane that undergoes a deformation when excited with an electrical voltage. DEAs have recently received a significant interest due to their high energy density, high deformation ranges, and low production costs. They have also showed to be quite attractive in the context of several applications, ranging from micro-positioning systems to soft-robotic structures. However, the interesting features of the DEAs are limited by their

strong nonlinear behavior and sensitivity to environmental conditions, which limit their penetration in the industrial sector. The strong nonlinearities due to the underlying physical behavior encouraged the development of advanced control strategies. Nevertheless, energy-efficient controllers have never been developed for such class of actuators. In this thesis, a novel minimum energy control strategy for DEAs is developed. The objective is to minimize the electrical energy required during a positioning task. In principle, the DEA dynamics can be detailed by an energy consistent model, which also describes the losses that occur in the actuator during any positioning task. An optimal feedback control strategy can be employed to minimize those losses, by formulating the energy-minimization problem as an optimal control problem. However, due to the involved nonlinearities, an analytic solution of the HJB equation does not exist. In this thesis, an ADP algorithm with off-policy learning is employed to deal with the optimal energy control problem. In particular, the ADP approach will be used as a tool to solve offline the HJB equation, deriving energy-efficient control laws for a given set of target displacement values. Finally, experimental tests will validate for the first time an energy consistent model of the DEA as well as the energy-efficient controllers. Substantially improvements in terms of energy saving will arise when comparing the proposed approach with other traditional control methods, such as Proportional Integral or feed-forward schemes.

The second complex system where ADP is applied is a DC microgrid featuring power buffers. Due to the increasing penetration of DC sources and loads, such as photo-voltaic generators or electrical vehicles, DC microgrids have recently gained significant attention. DC distribution systems are more efficient and reliable than AC microgrids, where redundant conversion stages are present. Moreover, DC microgrids do not suffer of many AC-related issues, such as frequency synchronization or reactive power flows. However, due to a lack of damping inertia, DC systems can face instability issues when volatile source and loads are considered. A possible solution is represented by power buffers, which can be used as damping elements in the DC microgrid. A power buffer is a power converter with a large storage element that can be exploited to decouple the distribution grid from the final load. In fact, when abrupt load changes occur, the energy stored in the buffer compensates the transient mismatch. The input impedance seen by the network can be actively controlled by the power buffer during transients, so that the stability properties of the DC system are improved. By introducing a communication network on top of the physical grid, distributed control policies for such buffers are enabled. Their effective range of action is thus extended to the neighboring power buffers. In this way, power buffers can assist each other during abrupt load changes. This thesis investigates the cooperative distributed control of power buffers. The cooperative assistive control objective is formulated as an optimal control objective, where the single utility function is shared among all the buffers. In contrast with the existing literature, the nonlinear dynamics is considered. Thus, ADP will represent the key tool in designing such optimal policies. Clearly, when dealing with distributed control schemes, the communication topology plays a crucial role. Based on the configuration of the communication network, in this thesis two different control approaches will be presented.

Firstly, the communication topology is fixed and inspired by the physical vicinity of the buffers. A set of optimal control policies able to provide assistance during abrupt load changes are learned offline, using the ADP with off-policy learning approach. Such policies are then interpolated in a real-time control scheme. The proposed approach overcome the issues of the existing distributed solutions for power buffers. For example, a feedback controller is directly provided, instead of open-loop policies that require additional control loop to be implemented.

By considering the fully nonlinear dynamics, the proposed approach does not rely on small-signal approximations. Thus, performances and stability will be guaranteed also for large-signal variations. Experimental validations conducted in a Controller/Hardware-in-the-Loop (CHIL) environment will asses the effectiveness of the proposed approach.

A second approach considers the communication topology a free parameter subject to optimization. In fact, there is no guarantee that a communication topology inspired by the physical vicinity is optimal with regard to the control objectives. Moreover, the energy availability of each power buffer is limited, thus the co-optimization of control performances and communication topologies is important when distributed solutions are considered. A sparsity-promoting optimal controller optimizes a closed-loop utility function, while minimizing at the same time the number of interactions between different control loops. Clearly, DC systems can benefit from sparse communication structures, minimizing computational and communication costs with a limited impact on the resulting closed-loop performances. However, the existing linear formulations for the sparsity-promoting optimal control are not practical for nonlinear systems as the DC microgrid with power buffers. This thesis presents the first attempt in solving sparsity-promoting optimal control problems for nonlinear systems. The versatility properties of the ADP algorithm with off-policy learning are exploited to provide such solution, without requiring the exact knowledge of the system dynamics. In fact, a single set of learning data is repetitively used to find optimal controllers for different communication topologies. The proposed data-driven algorithm employs Domain-of-Attraction estimation methods to check the stability of each distributed controller, while a Tabu Search procedure optimizes the combinatorial problem. The obtained sparsity-promoting controllers are then employed in the DC microgrid. The validity of the proposed approach will be assessed through exhaustive CHIL experiments. Quantitative and qualitative comparisons will show how the proposed methodology significantly outperforms existing approaches.

To my family.

”If you torture the data long enough, it will confess.”
Ronald H. Coase

Acknowledgments

I would like to thank my tutor Prof. David Naso from Politecnico di Bari for his inestimable supervision, crucial advice, and support throughout the course of my Ph.D studies.

A special thank goes also to Prof. Ali Davoudi and Prof. Frank Lewis from the University of Texas at Arlington for sharing their immense knowledge and expertise with me during my abroad studies.

I would like to express my infinite gratitude also to Prof. Gianluca Rizzello from the Universität des Saarlandes for his mentorship, friendship, and precious help and advice that made my research work successful.

I would like to thank also my friends and colleagues from the EFB lab of Politecnico di Bari and from the Complex Power Electronic Systems Laboratory of the University of Texas at Arlington.

I would like to thank my family for encouraging me.

Last but not least, thanks to all of my lifetime friends, and every special people I met during this incredible journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Artificial Intelligence and Control Theory

The current society has been distinguished for the significant development of IT-related technologies. This has led many industries and businesses to experience substantial changes in the adopted equipment and machinery, business systems, and labor models [1]. Advanced automated industrial systems played a very important role in determining and shaping such changes. A proof of this is the notable increase in the use of industrial robots in the last 10 years in almost all sectors of the world's industry, as shown in Figure 1.1, where the recent trends in the industrial robot installations by region (Fig. 1.1(a)) and by industries (Fig. 1.1(b)) are depicted [2].

However, the disruptive role of the automation comes to the price of a continuously growing complexity of industrial plants, production systems, and decision-making processes. In this context, control algorithms are crucial to guarantee performances and objectives of such complex systems. The design of efficient and reliable controllers is thus essential to achieve the strategic goals of industries and businesses. In order to deal with modern complex systems, guarantee performances, and satisfy strict specifications, advanced control design techniques play a key role in the near future of engineered systems.

The IEEE Computer Society draws up every year a prediction of the top 12 technology trends for the following year. The previsions for the 2020 see an increasing adoption of Artificial Intelligence (AI) technologies in several applications, such as cognitive robots, delivery drones, cybersecurity, and critical systems [3]. Regarding the latter, within five years AI will be significantly employed on various levels (e.g., control algorithms, communication infrastructures, etc.) of critical systems involved in the health and safety of the society. Any system where a failure could lead to a serious personal injury, damage the natural environment, or a loss of important assets or sensitive data is a critical system. Examples include medical devices and equipment, power generation and distribution, banking and stock trading systems. AI will enhance critical systems safety and reliability, while optimizing scarce resources [3].

The research community made several efforts in the last years to develop AI-based control techniques, providing well established frameworks that deal with complex systems. The subject of this thesis is the application of Reinforcement Learning (RL) based techniques, i.e., a class of AI methods, for the Optimal Control of nonlinear systems. The following sections 1.2 and 1.3 briefly overview optimal control and RL methods, respectively. Motivation and goals of this thesis are presented in Section 1.4. Finally, the structure of the thesis is reported in Section 1.5.

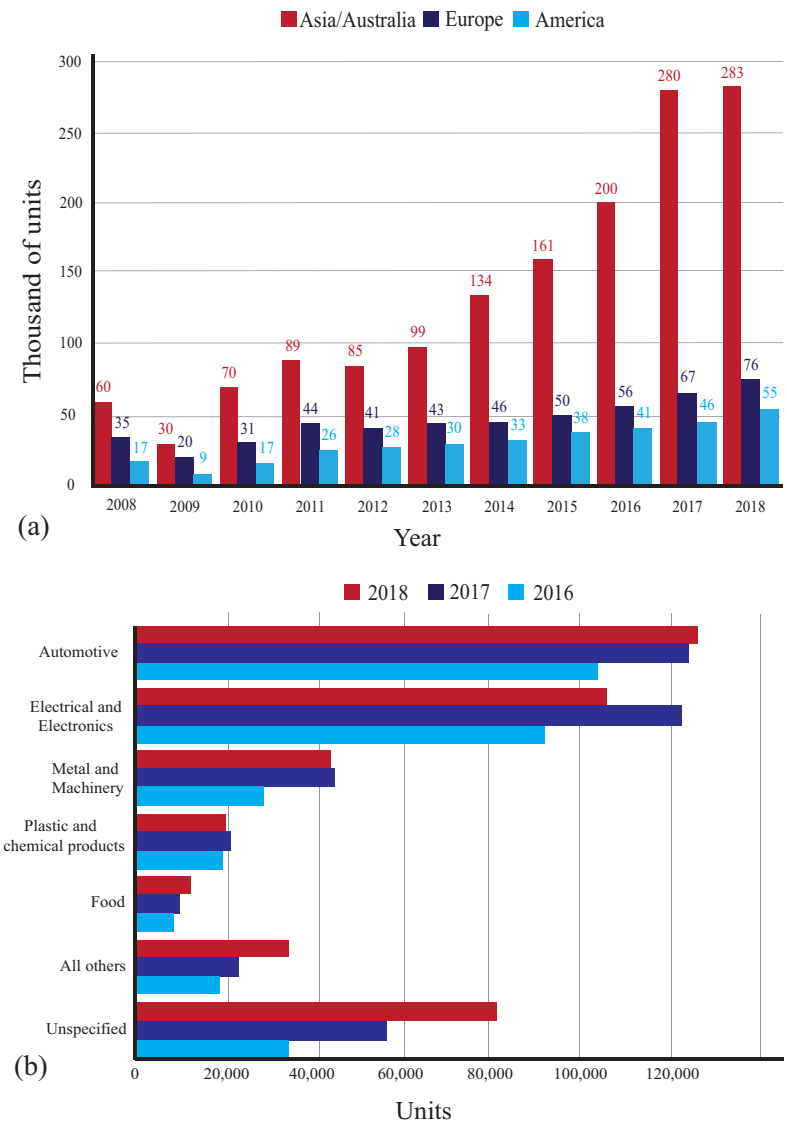

Figure 1.1: Recent trends in industrial robot installations: (a) Annual installations by region; (b) Annual installations by industries. Source: [2].

1.2 Optimal Control

Optimal control theory is one of the most used methods for designing feedback control systems. The history of optimal control stretches back 360 years, when the first theories based on the calculus of variations were developed. However, interest in optimal control consistently rose after the advent of the computer, with the first applications in the optimal trajectory prediction for the aerospace in the early 1960s [4]. Any optimal controller minimizes a given long-term performance index defined according to the resulting closed-loop system dynamics. The performance index includes both system states and control inputs, and reflects the desired transient response behaviors [5]. Several optimal control techniques have been developed for both discrete and continuous time systems, with infinite or finite optimization horizons. The work of this thesis is focused on the infinite-horizon optimal control of nonlinear continuous time systems.

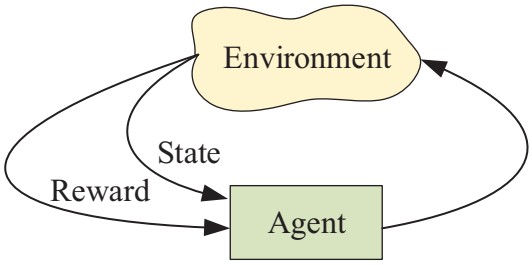

Figure 1.2: Standard RL model.

For general nonlinear systems, optimal control policies can be found using two approaches: the Pontryagin's Minimum Principle (PMP) and the Dynamic Programming (DP) [6, 7]. The PMP method is derived using the variational approach [8] and provides open loop controllers, while the DP approach leads to the Hamilton-Jacobi-Bellman (HJB) equation, whose solution provides a closed-loop controller with state-feedback. Moreover, the PMP method gives only a necessary condition for the solution optimality, i.e., it provides candidate solutions to be tested for optimality. By contrast, the HJB method provides both necessary and sufficient conditions, at the price of an higher complexity. In fact, while the PMP method is generally easier to tackle, the HJB partial differential equation is, in general, intractable. Due to better performances in terms of noise and disturbance rejections, feedback policies are usually preferred to open-loop controllers. In this thesis, Reinforcement Learning (RL) based techniques are employed to approximate the solution of the HJB equation, including the case of unknown system's dynamics.

1.3 Reinforcement Learning for Optimal Control

Machine Learning (ML) techniques are based on several learning paradigms, such as supervised learning, unsupervised learning, deep learning, etc. An important branch of the ML theory is constituted by the RL approach. RL methods are inspired by the animal world, where species survive and improve their behaviors by evaluating their actions on the external environment [9]. In the standard RL model, as depicted in Fig. 1.2, an agent, or actor, interacts with the environment by applying actions and receiving the current environment's state and a scalar value representing a reward or penalty, known as reinforcement signal. Based on the current state and reinforcement signal, the objective of the agent is to determine a sequence of actions that maximizes the rewards sum on the long term. Actions, or control policies, are modified based on the corresponding rewards, achieving the so called action-based learning. The main idea is that good actions, i.e., the ones resulting in high rewards, are remembered and reused. Note that since only the current states and rewards are required to the agent, such methods works also with unknown environments [9, 10]. The key concepts of the RL theory can be found in [11–13].

Mathematically an RL problem is usually formulated as an optimization problem where an optimal actions policy minimizes, or maximizes, a given objective function. Therefore, in principle RL can be used to solve optimal control problems. In fact, RL represents a well established approach in the control systems community to handle optimal control problems for unknown nonlinear systems. The first attempts of using RL to solve such problems were made initially for discrete time systems, where the DP approach is employed. A strict connection

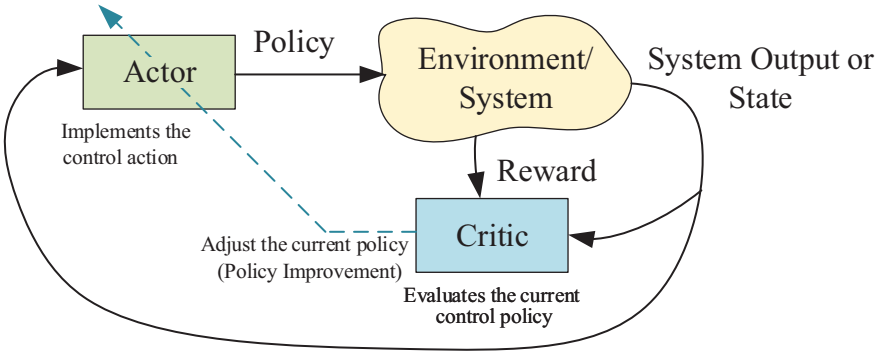

Figure 1.3: Actor-Critic structure.

between DP and RL was initially showed by Werbos in 1968 in [14]. After that, other RL-based approaches aimed at overcoming the issues of traditional DP techniques were presented in [15–18]. In particular, the traditional DP approach provides an exact solution of the discrete optimal control problem given an exhaustive search in the policy space of the system. Clearly, such approach is practical only for very small systems. Higher numbers of states could led to the well-known curse-of-dimensionality problem of the DP. Based on a RL method known as actor-critic structure [19], several methods that approximate the solution of the DP were presented [16, 20–22].

1.3.1 Actor-Critic Structure

The actor-critic structure features two main components, as depicted in Fig. 1.3. By applying a control policy, the actor component directly interacts with the environment, i.e., the system to be controlled in the optimal way. Then, a critic component evaluates the value of that policy according to a predefined performance index, i.e., the optimal control problem's cost function [9]. This approach consists of two steps performed iteratively: the policy evaluation by the critic and the policy improvement by the actor. The main idea is that the critic, based on the current policy evaluation, adjusts and improves the control policy of the actor so that the resulting performance index is improved with respect to the previous iteration. Clearly, the critic component evaluates the policy by observing the results on the environment of the current policy.

The Bellman's Principle provides the key equation in the optimal control theory, i.e., the Bellman optimality condition or discrete-time HJB equation [6]. The solution of such equation is the optimal control policy found by a backward-in-time process. Consequently, such method provides an offline solution without any online learning procedure. As an example, the solution of the discrete-time Linear Quadratic Regulator (LQR) problem is obtained by solving offline the Riccati equation, given the full knowledge of the system dynamics. The first step required to switch from an offline planning to an online learning problem is to provide a forward-in-time solution to the optimal control problem.

A basic implementation of the actor-critic structure, namely the Policy Iteration (PI) algorithm [22, 23], finds in an iterative way the optimal control policy using a forward-in-time approach, without solving the HJB equation. For discrete-time and continuous-time linear systems, the PI algorithm coincides with the well-known Hewer's algorithm [24] and Kleinman's algorithm [25], respectively.

However, when dealing with nonlinear systems the PI approach requires the solution of nonlinear Lyapunov equations, which are generally analytically intractable [9]. By making use of the Weierstrass approximation Theorem the optimal controllers can be approximated using Neural Networks (NN). The combination of actor-critic structures, e.g., the PI algorithm, with NNs results in the Adaptive Dynamic Programming (ADP) approach [16, 26, 27].

1.3.2 Adaptive Dynamic Programming

ADP methods formulate the optimal control problem as an on-line RL problem where the optimal controller is found using data measured along the system trajectories. The controller, i.e., the RL agent, approximates the optimal feedback policy without requiring a full knowledge of the environment, i.e., the system to be controlled. ADP algorithms learn optimal control policies by analyzing the online behavior of the system under non-optimal controllers. More specifically, a performance index quantifies the optimality of the current closed-loop system and drives the control policy updates. Such paradigm, where a control policy is modified according to the system responses, is strictly related to adaptive control techniques [9, 22].

Briefly, ADP employs NNs to provide approximated solutions to optimal control problems, i.e., approximated solutions of the HJB equation for continuous-time systems [28–32], and approximated forward-in-time solutions to the DP approach for discrete-time systems [33–36]. ADP techniques are mainly divided into PI and Value Iteration (VI) algorithms. Starting from an initial admissible stable control policy, PI algorithms iterate over two steps, i.e., a policy evaluation step and a policy improvement step [11, 37]. Conversely, a VI approach does not require an initial admissible stable control policy.

Based on the actor-critic structure, in the ADP framework a critic NN approximates the optimal value function, i.e., the optimal cost-to-go function, and an actor NN approximates the optimal control policy [38]. Furthermore, on-policy and off-policy learning methods can be defined. The former updates the current control policy using data obtained by applying the same control policy, while the latter permits the repeated use of the same set of data collected using a single initial stable control policy [39]. Nonlinear optimal control of continuous-time systems has been successfully tackled with several ADP-based algorithms. For instance, in [28] and [29], two on-policy PI algorithms have been proposed. The former is based on sequential updates of the actor and critic NNs, while in the latter the two NNs are synchronously updated. ADP for constrained-input systems is studied in [30]. In [31] and [32] two ADP algorithms with off-policy learning are proposed for systems with disturbances. For a comprehensive overview refer to [9, 27, 40].

1.3.3 The Exploration-Exploitation Dilemma

Each RL agent investigates the environment by applying actions and evaluating the corresponding rewards. In this context the agent can follow two different strategies. The first one makes use of the actions that resulted in high rewards in the previous steps, and thus consists in the exploitation of the knowledge gained by the RL agent so far. Alternatively, the agent applies a different set of actions that may result in an improved reward, or, in other words, the agent could explore the environment seeking a better set of actions. Therefore, ideally the agent should exploit its experience, but simultaneously explore the environment to learn new ac-

tions and improve future rewards. In the RL literature, this issue is commonly referred to as exploration-exploitation dilemma [11, 41].

As in the adaptive control methods, in the ADP framework the notion of persistent excitation (PE) is strictly related to the exploration-exploitation dilemma [42, 43]. If the PE conditions are not verified, the adaptive controller parameters will not converge to the optimal values. However, in order to satisfy the PE conditions, a probing noise is usually injected in the system, making the states and control inputs oscillatory and potentially causing instability. Therefore, from a control perspective, the dilemma consists in ensuring enough exploration (by satisfying the PE conditions) and guarantee stability and performances (exploitation) [44].

Thanks to the exploration capabilities of the ADP algorithms, they can be applied to systems with unknown dynamics and find optimal controllers with guaranteed convergence [44]. Note that given the difficulty in treating the HJB equation, such methods apply also when the system dynamics is known. In fact, by means of simulated models the ADP techniques can be used as a tool to solve offline the HJB equation and find the optimal control policy to be deployed on the real system.

1.4 Motivations and Goals of the Thesis

The focus of this thesis is the optimal control through ADP techniques of nonlinear continuous-time systems. Although the ADP theory for continuous-time problems has been developed since the second half of the 2000s [28, 45], still today most of the works focus on theoretical results only. To the author's best knowledge, the first experimental tests of ADP-based controllers to real-world plants have been published more recently. In [46] an ADP method with kernel-based function approximators is firstly proposed and then experimentally validated on single-link and double-link inverted pendulum systems. An ADP optimal controller is trained offline in [47] and used for the online torque control of a permanent magnet synchronous motor, achieving better performances when compared with traditional control techniques. The concept of concurrent learning is used in [48] to develop drop-free controllers for DC microgrids using ADP. In [49] the tracking control problem for unknown nonlinear systems is tackled with an ADP actor-critic structure featuring a NN identifier and then practically validated on an helicopter test-rig.

The aim of this thesis is to show the effectiveness of ADP-based control schemes in complex real-world scenarios, better assessing the potentials and drawbacks of such techniques. In particular, two main topics are treated. The first considers the minimization of the actuation energy of a Dielectric Elastomer Actuator, with the aim of developing and validating energy-efficient control policies. The second application consists in the distributed control of power buffers in DC microgrids. The goal is to develop assistive controllers where nearby power buffers share their stored energy to support each other during load changes. The following two subsections briefly cover these two applications.

1.4.1 Optimal Energy Control of Dielectric Elastomer Actuators

Dielectric elastomer (DE) transducers represent a possible solution to address the increasing demand of lightweight, fast, and precise electro-mechanical actuators. A DE is obtained by coating both sides of an elastomer film (e.g., silicone) with compliant electrodes (e.g., carbon grease), forming a flexible capacitor. When a voltage is applied between the electrodes, it

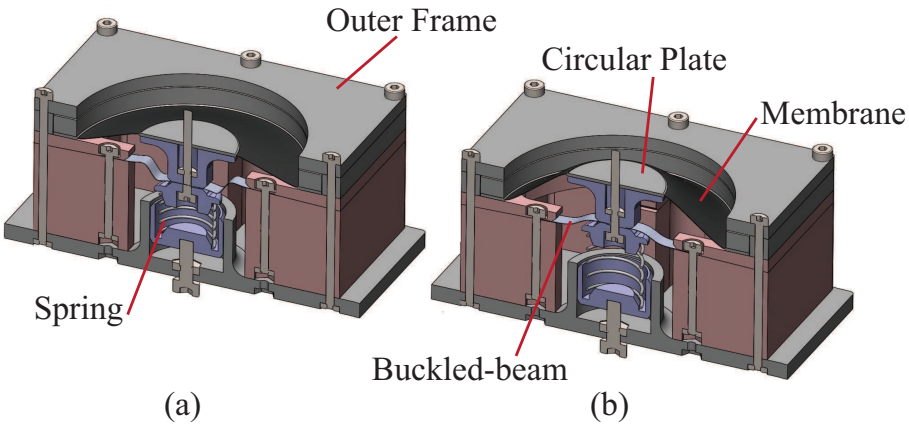

Figure 1.4: Cross section view of a DEA: (a) Unactuated (voltage off); (b) Actuated (voltage on).

generates a pressure that squeezes the membrane. This thickness reduction produces an area expansion which can be used for actuation purposes. An example of a DE Actuator (DEA) is depicted in Fig. 1.4, with both unactuated (Fig. 1.4(a)) and actuated (Fig. 1.4(b)) configurations. The DEA consists of a DE membrane placed in between an outer frame and a circular plate. Usually, a mechanical biasing system is employed to significantly amplify the actuation stroke [50]. For the DEA in Fig. 1.4 the biasing system consists of a combination of a bistable buckled-beam and a linear spring. In general, DEAs have gained a notable attention in recent years due to their unique combination of features, such as large deformations ($> 100\%$), fast response, and high energy density [51].

Despite these advantages, DE technology is currently limited by its highly nonlinear behavior, high voltage requirements (order of kV), and sensitivity to environmental conditions. To enhance the capabilities of DEAs, feedback control strategies have recently been developed, e.g., [50, 52]. While most authors focused on accurate position regulation, a feedback control scheme aimed at driving DEAs in an energy efficient way has not been tackled by the research community. In fact, despite DEAs are intrinsically energy efficient devices, electro-mechanical losses occur during actuation. The minimization of these losses can be addressed by means of optimal control strategies. Since the strong nonlinearity of the DE response makes the use of conventional optimal control theory not possible, ADP appears as a suitable tool to practically address the problem.

In this context, the main goals achieved in this thesis are summarized as follows:

- The obtained controller asymptotically tracks a position set-point by ensuring optimality with respect to a specific energy-related utility function, obtained via an accurate and thermodynamically consistent description of the system;
- Experimental studies validate the effectiveness of the energy-based model, which predicts with high accuracy the supplied electrical energy as well as the dissipation that occurs;
- The design of the optimal controller that minimizes the actuation energy is performed via an ADP algorithm with off-policy learning;
- The effectiveness of the optimal controller is verified through experimental tests and is

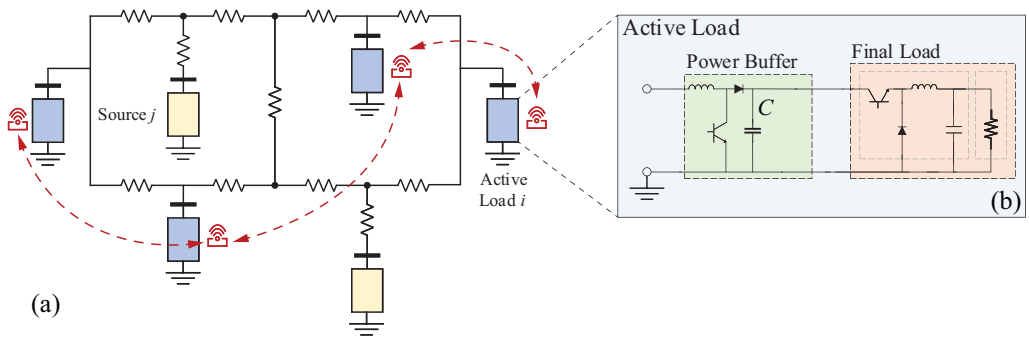

Figure 1.5: DC Microgrid with active loads: (a) DC microgrid as the interconnection of DC sources and active loads. A communication network (red lines) implements distributed control policies; (b) Active load as the series connection of a power buffer and a final load.

shown how the proposed approach outperforms, in terms of energy savings, other traditional control techniques.

1.4.2 Distributed Control of Power Buffers in DC Microgrids

A Direct Current (DC) microgrid consists of DC loads and sources spread on an electrical distribution network, as shown in Fig. 1.5(a). A common approach considers DC loads as active loads, i.e., the series connection of a controllable power buffer and a final load [53, 54], as in Fig. 1.5(b). The power buffer, i.e., a power converter with a large capacitor, features a fast voltage tracker that drives the final load, i.e., a power converter and a final resistive load. Placing a power buffer between the distribution network and the final load allows to better compensate, by means of the buffer's stored energy, the transient mismatch between the power supplied by the microgrid and the power delivered to the load (e.g. following an abrupt change in the load resistance). This results in improved stability performances of the DC network [55, 56].

Power buffers indeed provide an additional degree of freedom that can be exploited to design control laws that improve the overall network performances. In particular, the input impedance of the power buffers can be controlled in such a way they can provide assistance to other nearby active loads during transients, by sharing in some sense the stored energy. This results in a distributed controller deployed through a communication network spread among the active loads, as shown in Fig. 1.5(a). Thanks to this network, each power buffer can ideally extend its effective range by actively assisting a given set of neighboring loads during their transients. Within this perspective, a power buffer also reacts to load changes of its neighborhood, as well as to its own changes.

Clearly, the communication network plays an important role in the resulting distributed controller architecture. Two main cases are treated in the following thesis, as follows.

1. The communication network is fixed and inspired by the physical vicinity, and the fully nonlinear dynamics is considered. The main goal achieved are:

    - A distributed control law is designed, in the sense that each buffer's control law depends on its state and the one of its neighbors. By making use of limited information, the power buffers minimize a given performance function. Each buffer's

control law is designed according to the optimal control theory, using a common shared objective which changes according to the node that is requesting assistance. Given the intrinsically nonlinear coupling between DC sources and loads, the resulting optimal control problem is nonlinear, and, thus, is solved by means of ADP.

- With respect to the solutions proposed in previous research works, this approach provides an optimal control law without considering a small-signal analysis [53,54]. Thus, the resulting policy optimizes trajectories undergoing also large state deviations. Moreover, the proposed approach provides directly a state-feedback control law since it is derived by solving the HJB equation via ADP, instead of using the Pontryagin minimum principle as presented in the recent literature [57].

2. The communication network is free and subject to optimization. In particular, a sparsity-promoting objective is considered, i.e., the co-optimization of the closed loop optimal control performance and the number of the active communication links [58]. In order to provide better performances with respect to traditional approaches that consider first-order approximation, a second-order linearization of the system dynamics is considered. The main goals achieved are:

   - The first attempt to solve nonlinear sparsity-promoting and structured optimal control problems is proposed. In particular, an algorithm based on ADP and heuristic search methods (Tabu Search) handles arbitrary nonlinear utility functions and system dynamics.
   - When employed in DC networks, the optimal sparse controller has a limited impact on the performance function when compared with fully-connected communication topologies. In particular, the reciprocal assistance among power buffers is shown to increase with a less sparse communication structure.

Finally, Controller/Hardware-In-the-Loop (CHIL) experiments validates the effectiveness of the proposed solutions.

1.5 Structure of the Thesis

This doctoral thesis is organized as follows. Chapter 2 provides an overview of the ADP algorithms for the optimal control of nonlinear systems. The general continuous-time optimal control problem with infinite horizon is formulated. Then, the ADP with PI algorithms featuring on-policy and off-policy learning methods are discussed. Convergence proofs, exhaustive mathematical details, as well as pros and cons of the two learning approaches are reported. Numerical examples are provided to better familiarize the reader with the topic. Finally, a first application of ADP-based techniques for the structured optimal control of symmetrically-coupled linear systems is developed to show the potential and versatility of the ADP approach. Chapter 3 develops optimal energy controllers for DEAs using ADP. An energy-consistent model for such devices is first developed and experimentally validated. Then, the energy minimization problem is formulated according to the optimal control theory. An ADP algorithm with off-policy learning is used to find energy optimal policies regarding several desired displacement scenarios. Finally, the effectiveness of the proposed approach is verified through experimental tests.

Chapter 4 presents the application of ADP-based controllers for the distributed control of power buffers in DC microgrids. After an exhaustive literature overview, the operating principle of a power buffer is discussed. A nonlinear model of the DC microgrid with power buffer is developed. Then, two different approaches based on the configuration of the communication network implementing the distributed control routines are considered, as discussed in the previous subsection 1.4.2. Algorithmic details together with the resulting control schemes are provided. Extensive CHIL studies asses from a qualitative and quantitative perspective the performances and effectiveness of the proposed methods. Finally, conclusions and future perspectives are reported in Chapter 5.

1.6 List of Scientific Publications

1.6.1 Journals

1. P. R. Massenio, D. Naso, F. L. Lewis and A. Davoudi, "Assistive Power Buffer Control via Adaptive Dynamic Programming," in *IEEE Transactions on Energy Conversion*, vol. 35, no. 3, pp. 1534-1546, Sept. 2020. **Best paper award in the IEEE Transactions on Energy Conversion in the field of electric storage for the period 2019–2020**.
   doi: 10.1109/TEC.2020.2983154

2. P. R. Massenio, G. Rizzello, G. Comitangelo, D. Naso and S. Seelecke, "Reinforcement Learning-Based Minimum Energy Position Control of Dielectric Elastomer Actuators," in *IEEE Transactions on Control Systems Technology (Early Access)*.
   doi: 10.1109/TCST.2020.3022951

3. P. R. Massenio, D. Naso, F. L. Lewis and A. Davoudi, "Data-driven Sparsity-promoting Optimal Control of Power Buffers in DC Microgrids," in *IEEE Transactions on Energy Conversion (Early Access)*.
   doi: 10.1109/TEC.2020.3043709

1.6.2 Conference Proceedings

1. P. R. Massenio, G. Rizzello and D. Naso, "Fuzzy Adaptive Dynamic Programming Minimum Energy Control Of Dielectric Elastomer Actuators," *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, New Orleans, LA, USA, 2019, pp. 1-6.
   doi: 10.1109/FUZZ-IEEE.2019.8858901

2. P. R. Massenio, G. Rizzello and D. Naso, "Energy Optimal Control of Dielectric Elastomer Actuators via Adaptive Dynamic Programming," *2019 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 9: 15th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Anaheim, CA, USA, 2019.
   doi: 10.1115/DETC2019-98156

3. P. R. Massenio, G. Rizzello, D. Naso, F. L. Lewis and A. Davoudi, "Data-Driven Optimal Structured Control for Unknown Symmetric Systems," *2020 IEEE 16th International*

Conference on Automation Science and Engineering (CASE), Hong Kong, Hong Kong, 2020, pp. 179-184.
doi: 10.1109/CASE48305.2020.9216852

Chapter 2

Adaptive Dynamic Programming Algorithms

In this chapter, a brief introduction on Adaptive Dynamics Programming (ADP) methods is provided. First, the continuous-time nonlinear optimal control problem with infinite horizon is defined. Then, ADP methods that solve the optimal control problem for unknown nonlinear systems are summarized, covering the principles of Policy Iteration (PI) algorithms with on-policy and off-policy learning. Numerical examples are provided. Finally, a first application of Reinforcement Learning (RL) based techniques for the structured optimal control of unknown symmetric linear systems is reported.

2.1 Problem Statement

Optimal control designs find feedback control policies that minimize a given closed-loop performance index. Let's consider a general nonlinear control-affine system, as follows

$$\dot{x} = f(x) + g(x)u, \tag{2.1}$$

where $x \in \mathbb{X} \subseteq \mathbb{R}^n$ is the system's states vector, and $u \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control input vector, with $\mathbb{X}$ and $\mathbb{U}$ as compact sets. The following assumptions are made

- $f(x)$ is Lipschitz continuous on $\mathbb{X}$,
- $f(0) = 0$,
- the origin is the unique equilibrium of the system in $\mathbb{X}$,
- the system is feedback stabilizable, i.e., there exists at least one control feedback $u(x)$ such that the system is asymptotically stable on $\mathbb{X}$.

For any initial state $x(0) = x_0$, the infinite horizon integral cost, also known as *cost-to-go function* or *value function*, is defined as

$$V(x_0) = \int_0^\infty U(x, u)dt, \tag{2.2}$$

where $U(x, u)$ is a scalar function, known as *utility function*, embedding some specifications on system performances, e.g., minimum-fuel, minimum-energy, etc. For any stable control policy, u, (2.2) provides the performance measure of the system trajectory when going from the initial state to the equilibrium, i.e., the origin. Let's consider utility functions expressed in the following form

$$U(x, u) = Q(x) + u^{\intercal} R(x) u + \alpha^{\intercal}(x) u, \tag{2.3}$$

where $Q(x) : \mathbb{X} \to \mathbb{R}$ is a positive definite scalar function such that $Q(0) = 0$, $R(x) : \mathbb{X} \to \mathbb{R}^{m \times m}$ is an m-th dimensional symmetric positive definite matrix such that $R(0) = 0$, and $\alpha(x) : \mathbb{X} \to \mathbb{R}^m$ is defined such that $\alpha(0) = 0$ and $U(x, u) > 0$, $\forall (x, u) \neq (0, 0)$, with $U(0, 0) = 0$. This requirements are needed in order to well-define the utility function.

Definition 1 - *Admissible Policy* [59]. A feedback control policy (sometimes simply referred to as control policy or policy), $u(x)$, is admissible regarding (2.2) on set $\mathbb{X}$ if $u(0) = 0$, $u(x)$ is a continuous function on $\mathbb{X}$, $u(x)$ asymptotically stabilizes system (2.1) on $\mathbb{X}$, and the associated cost $V(x_0)$ is finite $\forall x_0 \in \mathbb{X}$.

The nonlinear optimal control problem with an infinite horizon cost function can be now defined as follows: Given the continuous-time system (2.1) and the infinite horizon cost function (2.2), find an admissible feedback control policy $u(x)$ that minimizes (2.2).

Regardless the optimality of a control policy $u(x)$, the cost (2.2) obtained by applying $u(x)$ to system (2.1) can be evaluated a priori if the closed form of the function $V(x)$ is known. To this end, if the cost function $V(x)$ in (2.2) associated to $u(x)$ is continuous, its infinitesimal version gives us the following nonlinear Lyapunov equation [29]

$$U(x, u(x)) + \nabla V^{\intercal}(x)(f(x) + g(x)u(x)) = 0, \quad V(0) = 0 \tag{2.4}$$

where $\nabla V(x) \in \mathbb{R}^n$ is the value function's derivative with respect to the system states x. Given any admissible controller $u(x)$, the nonlinear Lyapunov equation (2.4) can be solved for the value function $V(x)$, which represents a Lyapunov function for system (2.1) with policy $u(x)$.

The Hamiltonian of the optimal control problem is defined as

$$H(x, u, \nabla V) = U(x, u(x)) + \nabla V^{\intercal}(x)(f(x) + g(x)u(x)). \tag{2.5}$$

Clearly, the optimal value function, $V^*(x)$, satisfies the following equation

$$\min_u H(x, u, \nabla V^*(x)) = 0, \tag{2.6}$$

which is also known as the Hamilton-Jacobi-Bellman (HJB) equation. If it is assumed that the minimum Hamiltonian in (2.6) exists and is unique, the corresponding optimal control policy, $u^*(x)$, can be found as

$$u^*(x) = -\frac{1}{2} R^{-1}(x) \left(g^{\intercal}(x) \nabla V^*(x) + \alpha(x) \right). \tag{2.7}$$

By plugging (2.7) in (2.5) the formulation of the HJB equation in terms of the optimal cost function is obtained [29],

$$\begin{aligned} Q(x) - \frac{1}{4} \nabla V^{*\intercal}(x) g(x) R^{-1}(x) g^{\intercal}(x) \nabla V^*(x) \\ + \frac{1}{4} \alpha^{\intercal}(x) R^{-1}(x) \alpha(x) + \nabla V^{*\intercal}(x) f(x) = 0, \quad V^*(0) = 0 \end{aligned} \tag{2.8}$$

Note that $V^*(x)$ constitutes a well-defined Lyapunov function for the closed-loop system made of (2.1) and input (2.7), which globally asymptotically stabilizes the system at the origin.

The optimal control policy is found by solving the HJB equation (2.8) for the optimal value function, $V^*(x)$, and plugging it in (2.7). It can be easily proven that $V^*(x)$ is the only positive-definite solution of the HJB equation. When dealing with time invariant linear systems and quadratic cost functionals, (2.8) becomes the Algebric Riccati Equation. However, when dealing with general nonlinear systems, the HJB equation is generally intractable, thus, approximated solutions via the ADP framework can be found.

2.2 Policy Iteration

A RL-based iterative method that solves optimal control problems is the well-known PI algorithm [11]. It consists of two steps: 1) Policy Evaluation and 2) Policy Improvement. Starting from an initial admissible control policy, the PI algorithm evaluates the corresponding cost by solving a nonlinear Lyapunov equation as in (2.4). Then, this cost is used to obtain a new control policy by minimizing an Hamiltonian function, as in (2.6) and (2.7). The new control policy is improved with respect to the previous one in the sense that the corresponding cost is smaller. The two steps are iteratively executed until improvements are no longer obtained, i.e., the optimal control policy is found [29, 60].

The PI algorithm is reported in the following Algorithm 2.1,

Algorithm 2.1 Policy Iteration Algorithm

1. **Initialization:** Set $k = 0$, and $u^{(0)}(x)$ as the initial admissible controller for (2.1).
2. **Iteration:** Repeat until convergence
   a. **Policy Evaluation:** Solve for $V^{(k)}(x)$, with $V^{(k)}(0) = 0$, from the following:

$$U(x, u^{(k)}(x)) + \nabla V^{(k)\intercal}(x)(f(x) + g(x)u^{(k)}(x)) = 0. \tag{2.9}$$

   b. **Policy Improvement:** Update $u^{(k+1)}(x)$ as:

$$u^{(k+1)}(x) = -\frac{1}{2}R^{-1}(x)\left(g^{\intercal}(x)\nabla V^{(k)}(x) + \alpha(x)\right). \tag{2.10}$$

Proof of convergence of the PI algorithm to the optimal control policy is given in several references, such as [59, 61, 62].

Note that for continuous time systems with quadratic utility functions the PI method reduces to the Kleinman algorithm which iteratively solves the Algebraic Riccati Equation [25].

The Policy Evaluation step in the PI algorithm requires the solution of the nonlinear Lyapunov equation (2.9), which is still an intractable problem for general nonlinear systems. Thus, ADP algorithms seek approximated solutions for the Policy Evaluation step and find an approximated optimal control policy. In particular, by employing two Neural Networks (NNs) the PI algorithm can be implemented using an actor/critic structure (see Section 1.3.1). The critic NN is trained to provide an approximated solution of (2.9), while an actor NN is trained to provide the improved policy during the Policy Improvement step [29].

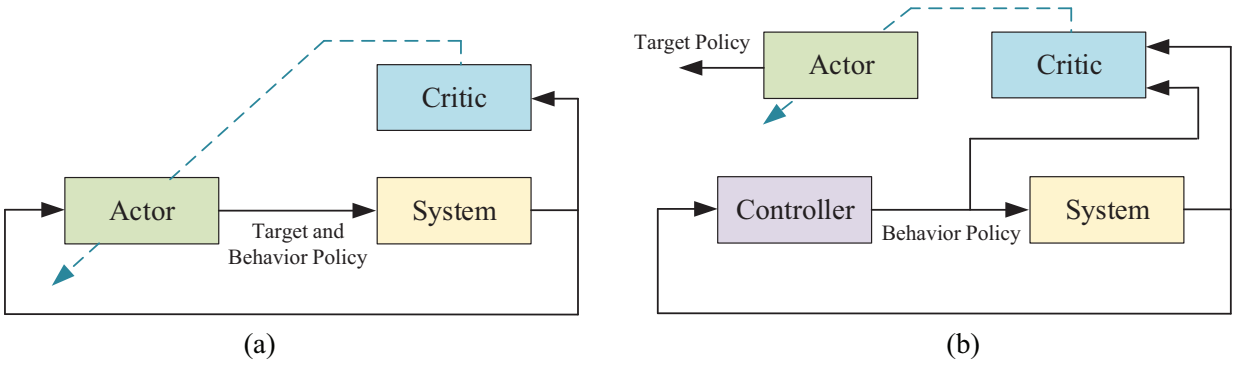

Figure 2.1: On-policy and off-policy learning methods: (a) On-policy; (b) Off-policy.

The PI algorithm can be implemented using two classes of learning methods, namely on-policy and off-policy. The on-policy method evaluates and improve the same control policy applied to the system, i.e., the policy is continuously updated in an online manner, as it happens in the adaptive control theory. In off-policy methods, instead, the policy being updated (namely the target policy) and the policy applied to the system (namely the behavior policy) are distinct. In other words, off-policy methods learn the optimal control policy by using the system's response to another policy. Figure 2.1 schematizes the two learning methods [11, 63].

For every iteration in the PI procedure with off-policy learning, (2.9) is solved by reusing the same data collected by executing a behavior policy. Thus, off-policy methods are in general more efficient and fast if compared with on-policy methods. However, if some of the system parameters change over time, the off-policy procedure must be re-executed, while an on-policy method provides real-time adaptation. Finally, off-policy algorithms can solve optimal control problems in case of completely unknown system dynamics [63].

On-policy and off-policy algorithms are presented in the following sections.

2.3 PI ADP with On-policy learning

In order to solve online (2.9), the critic component is implemented using a NN for the value function approximation. In particular, the final goal is to approximate $V^*(x)$ and its gradient. By assuming that (2.4) has a smooth solution for any admissible controller, i.e., $V(x) \in C^1$, the Weierstrass higher-order approximation theorem [59] ensures the existence of a independent basis function set such that $V(x)$ and its gradient are uniformly approximated on $\mathbb{X}$. That is, there exists a weights vector $\omega^* \in \mathbb{R}^{N_V}$ such that the value function $V(x)$ and its gradient are approximated as follows

$$V(x) = \sum_{l=1}^{N_V} \omega_l^* \gamma_l(x) + \epsilon_V(x) = \omega^{*\intercal} \Gamma(x) + \epsilon_V(x), \tag{2.11}$$

$$\nabla V(x) = \sum_{l=1}^{N_V} \omega_l^* \nabla \gamma_l(x) + \nabla \epsilon_V(x) = \nabla \Gamma^\intercal(x) \omega + \nabla \epsilon_V(x). \tag{2.12}$$

The linearly independent functions $\Gamma(x) = \{\gamma_1(x), \cdots, \gamma_{N_V}(x)\}$, with $\gamma_l(x) : \mathbb{R}^n \to \mathbb{R}$ and $\gamma_l(0) = 0$, can be seen as the activating functions of a NN with N_V neurons on the hidden

layer, while ω^* represents the weight's vector on the output layer, which has a linear activating function. Moreover, the weights on the hidden layer are fixed to 1 and polynomial activating functions are usually employed [29]. Clearly, as the number of the neurons in the hidden layer tends to infinity, i.e., $N_V \to \infty$, the approximating errors tends to zero, i.e., $\epsilon_V(x) \to 0$, and $\nabla\epsilon_V(x) \to 0$.

2.3.1 Critic's Tuning Law

Given a nonlinear Lyapunov equation as in (2.4) with a fixed admissible control policy $u(x)$, the goal is to determine a tuning law for the critic NN in (2.11) so that the solution of (2.4) is approximated. By plugging (2.12) into (2.4) the following expression is obtained

$$U(x, u(x)) + \omega^{*\intercal}\nabla\Gamma(x)(f(x) + g(x)u(x)) + \nabla\epsilon_V(x)(f(x) + g(x)u(x)) = 0. \tag{2.13}$$

By defining the residual error as $\epsilon_H(x) = -\nabla\epsilon_V(x)(f(x)+g(x)u(x))$, the following holds true

$$U(x, u(x)) + \omega^{*\intercal}\nabla\Gamma(x)(f(x) + g(x)u(x)) = \epsilon_H. \tag{2.14}$$

As in (2.11), ω^* represents the unknown weight's set providing the best approximation for the value function $V(x)$. Let ω be the current estimation of the ideal weights ω^*, thus, the current output of the critic NN is

$$\hat{V}(x) = \omega^\intercal\Gamma(x). \tag{2.15}$$

The current weights estimation error is defined as

$$\tilde{\omega} = \omega^* - \omega. \tag{2.16}$$

Thus, the current approximation error on the resulting Lyapunov equation, $\epsilon_L(x) \in \mathbb{R}$, is computed as

$$\begin{aligned}\epsilon_L(x) &= U(x, u(x)) + \omega^\intercal\nabla\Gamma(x)(f(x) + g(x)u(x)) \\ &= U(x, u(x)) + \omega^{*\intercal}\nabla\Gamma(x)(f(x) + g(x)u(x)) - \tilde{\omega}^\intercal\nabla\Gamma(x)(f(x) + g(x)u(x)) \\ &= -\tilde{\omega}^\intercal\nabla\Gamma(x)(f(x) + g(x)u(x)) + \epsilon_H,\end{aligned} \tag{2.17}$$

where (2.14) has been used in the last step. The goal is to find a tuning law for the weights ω so that the following error index is minimized

$$E_V = \frac{1}{2}\epsilon_L(x)^2, \tag{2.18}$$

then $\omega \to \omega^*$ and $\epsilon_L \to \epsilon_H$. The weights are tuned using a normalized gradient descent algorithm, i.e.,

$$\dot{\omega} = -\lambda_V\frac{\partial E_V}{\partial\omega} = -\lambda_V\epsilon_L\frac{\partial\epsilon_L}{\partial\omega}, \tag{2.19}$$

where the chain rule is used and $\lambda_V \in \mathbb{R}$ represents the learning rate. By defining $\sigma_V(x) \in \mathbb{R}^{N_V}$ as

$$\sigma_V(x) = \nabla\Gamma(x)(f(x) + g(x)u(x)), \tag{2.20}$$

and the normalization factor as $(\sigma_V^\intercal(x)\sigma_V(x) + 1)^2$, the following tuning law is obtained

$$\dot{\omega} = -\lambda_V\frac{\sigma_V(x)}{(\sigma_V^\intercal(x)\sigma_V(x) + 1)^2}(\sigma_V^\intercal(x)\omega + U(x, u(x))). \tag{2.21}$$

2.3.2 Convergence Analysis

To study the convergence of the critic's weights, i.e., if the ideal weights ω^* are obtained with (2.21), the dynamics of the critic weights estimation error, $\tilde{\omega}$, has to be analyzed. From (2.14) it follows that $U(x, u(x)) = -\omega^{*\intercal}\sigma_V(x) + \epsilon_H$. The dynamics of $\tilde{\omega}$ can be expressed as

$$\dot{\tilde{\omega}} = -\dot{\omega} = -\lambda_V \bar{\sigma_V}(x)\bar{\sigma_V}^{\intercal}(x)\tilde{\omega} + \lambda_V \bar{\sigma_V}(x)\frac{\epsilon_H}{m_V(x)}, \tag{2.22}$$

where $m_V(x) = \sigma_V^{\intercal}(x)\sigma_V(x) + 1$, and $\bar{\sigma_V}(x) = \frac{\sigma_V(x)}{m_V(x)}$. For compactness, the dependency of $\sigma_V(x)$ and $m_V(x)$ on x is omitted in the subsequent.

In order to guarantee the convergence of the critic weights is necessary to assume that the signal $\bar{\sigma_V}$ is Persistently Excited (PE). As in the adaptive control theory, where the PE condition is used to ensure the convergence of the identified system parameters, here the PE condition ensures proper identification of the critic parameters approximating the function $V(x)$. The PE condition states that if exist three constants $\beta_1^{PE} > 0$, $\beta_2^{PE} > 0$, and $T > 0$ so that the following holds

$$\beta_1^{PE} I \leq \int_t^{t+T} \bar{\sigma_V}(\tau)\bar{\sigma_V}^{\intercal}(\tau)d\tau \leq \beta_2^{PE} I, \quad \forall t, \tag{2.23}$$

where I is the identity matrix, then the signal $\bar{\sigma_V}$ is persistently excited over the time interval $[t, t+T]$. The inequality in (2.23) states that if $\bar{\sigma_V}(x)$ is PE, then the eigenvalues of the integral are positive and, thus, the integral is invertible. This condition is equivalent to the uniform complete observability [64] of the following linear time varying system

$$\begin{cases} \dot{\tilde{\omega}} = \lambda_V \bar{\sigma_V} u \\ y = \bar{\sigma_V}\tilde{\omega} \end{cases} \tag{2.24}$$

In fact, (2.23) represents the observablity gramiam of system (2.24), where u and y are the input and output, respectively.

The following theorem [29] proves that the tuning law defined in (2.21) is effective under the PE condition, so that the actual weights ω converge to the unknown ideal weights ω^*. Note that ω^* solve the general nonlinear Lypanuov equation (2.4) when the input $u(x)$ is fixed and admissible.

Theorem 1 - *Convergence of the critic weights.* Let $u(x)$ be an admissible control policy for system (2.1). Let us consider (2.21) as the tuning law for the critic's weights. Assume that $\bar{\sigma_V}$ is PE as in (2.23). Then the critic weights error converges exponentially to a residual set that shrinks as ϵ_H tend to 0, i.e., if the number of the hidden neurons is sufficiently large [59].
Proof. See [29].

Therefore, Theorem 1 ensures that through (2.21) the solution of any Lyapunov equation corresponding to the applied control policy can be found online. By setting the control policy as in (2.10), a solution of the HJB equation is obtained, i.e.,

$$\hat{u}(x) = -\frac{1}{2}R^{-1}(x)\left(g^{\intercal}(x)\nabla\Gamma^{\intercal}(x)\omega + \alpha(x)\right), \tag{2.25}$$

and, thus, the optimal control problem is solved online. This last equation represents the actor NN. The resulting actor/critic structure is depicted in Fig. 2.2. Note that a probing noise is added to the actor NN's output to guarantee the PE condition.

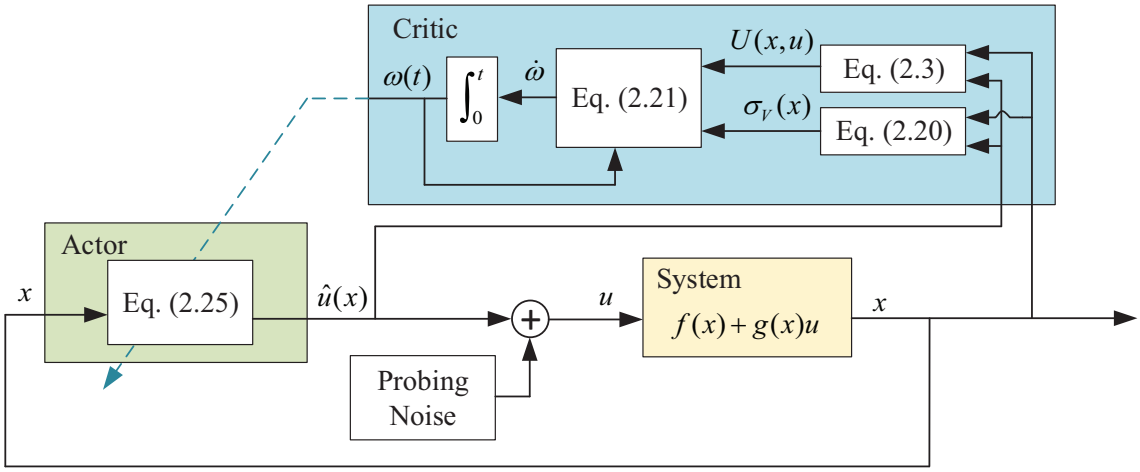

Figure 2.2: ADP with on-policy learning scheme.

A convergence proof with guaranteed stability of the closed loop system during the learning stage is proposed in [29], where the actor's weight are different from the critic's ones and tuned in a different way. However, the approach proposed in [29] does not consider the term $\alpha(x)u$ in the utility function, as in (2.3). Both schemes in Fig. 2.2 and in [29] represent a synchronous PI algorithm. In fact, by contrast with Algorithm 2.1, where the critic and actor NNs are updated sequentially, the synchronous PI simultaneously tunes both NNs in real-time.

In general, on-policy PI algorithms involve real-time tuning laws with non-standard extra terms employed to ensure closed-loop stability. However, such tuning laws are in general hard to set and design [29]. The real-time implementation of the PI algorithm with on-policy learning is computationally intensive and, as in (2.20), the full knowledge of the system dynamics is required. A hybrid PI algorithm constituted by a continuous-time actor component updated by a discrete-time critic structure is proposed in [60], where the knowledge of only $g(x)$ is required. The off-policy approach overcomes some limitations of the on-policy method.

2.4 PI ADP with Off-policy learning

The goal of the off-policy method is to implement a PI algorithm where (2.9) is easily solved by means of a least square approach, using only collected system data with a fixed control policy.

To this end [65], let us consider system (2.1) with an admissible control policy, $u^{(0)}(x)$, and a bounded exploration noise, $e_L(t) : \mathbb{R} \to \mathbb{R}^m$, injected for learning purposes, as follows

$$\dot{x} = f(x) + g(x)(u^{(0)}(x) + e_L(t)). \tag{2.26}$$

Clearly, the exploration noise must be chosen so that system (2.1) is input-to-state stable when $e_L(t)$ is the input. For any iteration $k \geq 0$ in Algorithm 2.1, (2.26) can be rewritten as follows

$$\dot{x} = f(x) + g(x)(u^{(k)}(x) + u^{(k)\prime}(x)), \tag{2.27}$$

where $u^{(k)\prime}(x) = u^{(0)}(x) - u^{(k)}(x) + e_L(t)$. By considering the policy evaluation and improvement steps, i.e., (2.9) and (2.10), respectively, the derivative with respect to the time of $V^{(k)}(x)$

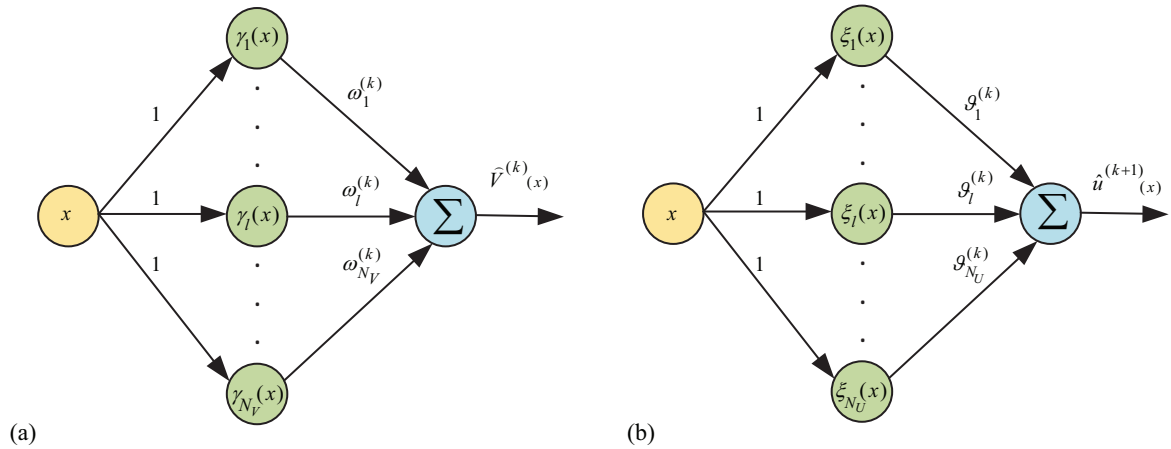

Figure 2.3: LIP approximators scheme: (a) Critic NN, as in (2.29); (b) Actor NN, as in (2.30).

along the state trajectories of (2.27) is

$$\begin{aligned}\dot{V}^{(k)}(x) &= \nabla V^{(k)\intercal}(x)(f(x) + g(x)u^{(k)}(x) + g(x)u^{(k)\prime}(x)) \\ &= -U(x, u^{(k)}(x)) + \nabla V^{(k)\intercal}(x)g(x)u^{(k)\prime}(x) = \\ &= -U(x, u^{(k)}(x)) - \left(2u^{(k+1)\intercal}(x)R(x) + \alpha^{\intercal}(x)\right) u^{(k)\prime}(x).\end{aligned} \tag{2.28}$$

As in the ADP algorithm with on-policy learning, also when using off-policy learning two NNs are employed. A critic NN approximates the value function at each step, i.e., $V^{(k)}(x)$, while an actor NN approximates the control policy, i.e., $u^{(k+1)}(x)$. In particular, linear-in-parameters (LIP) approximators are used, as follows

$$\hat{V}^{(k)}(x) = \sum_{l=1}^{N_V} \omega_l^{(k)} \gamma_l(x) = \omega^{(k)\intercal}\Gamma(x), \tag{2.29}$$

$$\hat{u}^{(k+1)}(x) = \sum_{l=1}^{N_U} \theta_l^{(k)} \xi_l(x) = \theta^{(k)\intercal}\Xi(x), \tag{2.30}$$

where $\gamma_l(x) : \mathbb{R}^n \to \mathbb{R}$, with $l = 1, ..., N_V$, and $\xi_l(x) : \mathbb{R}^n \to \mathbb{R}^m$, with $l = 1, ..., N_U$, are two sequences of linearly independent basis functions vanishing in the origin. N_V and N_U are two large integers representing the number of neurons, while $\omega^{(k)}$ and $\theta^{(k)}$ are two sets of unknown weights of suitable dimensions to be determined. Figure 2.3 depicts a scheme of the two considered NNs.

2.4.1 Integral Reinforcement Learning

The off-policy learning method is derived from the so-called integral reinforcement learning (IRL) equation [32,44]. That is, by integrating (2.28) on both sides on any time interval $[t_n, t_{n+1}]$ and substituting $V^{(k)}(x)$ and $u^{(k+1)}(x)$ with their approximations in (2.29) and (2.30), the fol-

lowing IRL equation is obtained

$$\begin{aligned}\omega^{(k)\intercal}\left[\Gamma(x(t_{n+1}))-\Gamma(x(t_n))\right] &= -\theta^{(k-1)\intercal}\left(\int_{t_n}^{t_{n+1}}\Xi(x)R(x)\Xi^{\intercal}(x)dt\right)\theta^{(k-1)}\\ &-2\theta^{(k)\intercal}\int_{t_n}^{t_{n+1}}\Xi(x)R(x)\left(u^{(0)}(x)+e_L(t)\right)dt+2\theta^{(k)\intercal}\left(\int_{t_n}^{t_{n+1}}\Xi(x)R(x)\Xi^{\intercal}(x)dt\right)\theta^{(k-1)}\\ &-\int_{t_n}^{t_{n+1}}Q(x)dt-\int_{t_n}^{t_{n+1}}\alpha^{\intercal}(x)\left(u^{(0)}(x)+e_L(t)\right)dt+\epsilon_n^{(k)},\end{aligned} \tag{2.31}$$

where $\epsilon_n^{(k)}$ is the error due to the approximation on the time interval $[t_n, t_{n+1}]$ and iteration k.

Given the sequence of time intervals $\{t_n\}_{n=0}^{N_L}$, (2.31) can be solved in a least-square sense using the data collected when the fixed control policy $u^{(0)}(x)+e_L(t)$ is applied to the system. Then, two sequences $\{\hat{u}^{(k+1)}(x)\}_{k=0}^{\infty}$ and $\{\hat{V}^{(k)}(x)\}_{k=0}^{\infty}$ can be generated. To guarantee the convergence of the two sequences to $u^{(k+1)}(x)$ and $V^{(k)}(x)$ in (2.9) and (2.10), a PE condition must be satisfied [65]. That is, the exploring noise $e_L(t)$ must be chosen so that there exist a N_L^0 and a $\beta^{PE}>0$ such that for all $N_L \geq N_L^0$ the following condition holds true

$$\sum_{n=0}^{N_L}\Theta_n^{(k)\intercal}\Theta_n^{(k)} \geq \beta^{PE} I_{N_V+N_U} \tag{2.32}$$

where $I_{N_V+N_U}$ is the identity matrix of dimension N_V+N_U. The row vector $\Theta_n^{(k)} \in \mathbb{R}^{N_V+N_V}$ is defined as

$$\Theta_n^{(k)\intercal} = \begin{bmatrix}\gamma_1(x(t_{n+1}))-\gamma_1(x(t_n))\\ \vdots\\ \gamma_{N_V}(x(t_{n+1}))-\gamma_{N_V}(x(t_n))\\ 2\int_{t_n}^{t_{n+1}}\xi_1^{\intercal}(x)R(x)u^{(k)\prime}(x)dt\\ \vdots\\ 2\int_{t_n}^{t_{n+1}}\xi_{N_U}^{\intercal}(x)R(x)u^{(k)\prime}(x)dt\end{bmatrix} \tag{2.33}$$

Note that the PE condition requires the positive definiteness of the matrix $\sum_{n=0}^{N_L}\Theta_n^{(k)\intercal}\Theta_n^{(k)}$.

2.4.2 Convergence Analysis

The following theorem ensures the convergence of the two sequences $\{\hat{u}^{(k+1)}(x)\}_{k=0}^{\infty}$ and $\{\hat{V}^{(k)}(x)\}_{k=0}^{\infty}$ to $u^{(k+1)}(x)$ and $V^{(k)}(x)$ in (2.9) and (2.10) when the number of neurons is sufficiency large and the PE condition is satisfied.

Theorem 2 - *Convergence of critic and actor weights [65].* When condition (2.32) is satisfied, for each $k \geq 0$ and for any given $\epsilon > 0$ there exist $N_V^* > 0$, $N_U^* > 0$ such that

$$\left|\sum_{l=1}^{N_V}\omega_l^{(k)}\gamma_l(x)-V^{(k)}(x)\right| < \epsilon \tag{2.34}$$

$$\left|\sum_{l=1}^{N_U}\theta_l^{(k)}\xi_l(x)-u^{(k+1)}(x)\right| < \epsilon \tag{2.35}$$

for all $x \in \mathbb{X}$ if $N_V > N_V^*$ and $N_U > N_U^*$.

Proof. The following proof, herein reported for completeness, is an extended version of the proof presented in [65], where a less general utility function has been considered, i.e., $\alpha(x) = 0$ in (2.3).

Given $\hat{u}^{(k)}(x)$, let $\tilde{V}^{(k)}(x)$ be the solution of the following Lyapunov equation

$$\nabla \tilde{V}^{(k)\intercal}(x) \left(f(x) + g(x)\hat{u}^{(k)}(x)\right) + U(x, \hat{u}^{(k)}(x)) = 0, \tag{2.36}$$

and let's define $\tilde{u}^{(k+1)}(x) = -\frac{1}{2}R^{-1}(x)\left[g^\intercal(x)\nabla\tilde{V}^{(k)}(x) + \alpha(x)\right]$. The first result to be proved is that for each $k \geq 0$ and for all $x \in \mathbb{X}$, it results that

$$\begin{aligned} \lim_{N_V, N_U \to \infty} \hat{V}^{(k)}(x) &= \tilde{V}^{(k)}(x) \\ \lim_{N_V, N_U \to \infty} \hat{u}^{(k+1)}(x) &= \tilde{u}^{(k+1)}(x), \end{aligned} \tag{2.37}$$

when (2.31) is solved using the least square method. Note that (2.36) is the Lyapunov equation in (2.9) when the actual estimate of the optimal control policy, i.e., $\hat{u}^{(k)}(x)$ is plugged in.

By considering (2.36), the time derivative of $\tilde{V}^{(k)}(x)$ along the trajectories of (2.27) when $u^{(k)}(x)$ is replaced by $\hat{u}^{(k)}(x)$ is

$$\begin{aligned} \dot{\tilde{V}}^{(k)}(x) &= \nabla\tilde{V}^{(k)\intercal}(x)\left(f(x) + g(x)\left(\hat{u}^{(k)}(x) + \hat{u}^{(k)\prime}(x)\right)\right) = \\ &= -U(x, \hat{u}^{(k)}(x)) + g^\intercal(x)\nabla\tilde{V}^{(k)}(x)\hat{u}^{(k)\prime}(x). \end{aligned} \tag{2.38}$$

Therefore, by integrating both members of this last equation it follows that

$$\begin{aligned} \tilde{V}^{(k)}(x(t_{n+1})) - \tilde{V}^{(k)}(x(t_n)) = &- \int_{t_n}^{t_{n+1}} U(x, \hat{u}^{(k)}(x))dt \\ &- \int_{t_n}^{t_{n+1}} \left(2\tilde{u}^{(k+1)\intercal}R(x) + \alpha^\intercal(x)\right)\hat{u}^{(k)\prime}(x)dt. \end{aligned} \tag{2.39}$$

By virtue of the Weierstrass higher order approximation theorem there exist some constant weights $\tilde{\omega}^{(k)}$ and $\tilde{\theta}^{(k)}$ so that

$$\begin{aligned} \tilde{V}^{(k)}(x) &= \sum_{l=1}^{\infty} \tilde{\omega}_l^{(k)}\gamma_l(x) = \sum_{l=1}^{N_V} \tilde{\omega}_l^{(k)}\gamma_l(x) + \sum_{l=N_V+1}^{\infty} \tilde{\omega}_l^{(k)}\gamma_l(x) \\ \tilde{u}^{(k+1)}(x) &= \sum_{l=1}^{\infty} \tilde{\theta}_l^{(k)}\xi_l(x) = \sum_{l=1}^{N_U} \tilde{\theta}_l^{(k)}\xi_l(x) + \sum_{l=N_V+1}^{\infty} \tilde{\theta}_l^{(k)}\xi_l(x), \end{aligned} \tag{2.40}$$

by substituting (2.20) into (2.39) the following expression is obtained

$$\begin{aligned} &\sum_{l=1}^{N_V} \tilde{\omega}_l^{(k)}\left[\gamma_l(x(t_{n+1})) - \gamma_l(x(t_n))\right] + \sum_{l=N_V+1}^{\infty} \tilde{\omega}_l^{(k)}\left[\gamma_l(x(t_{n+1})) - \gamma_l(x(t_n))\right] = \\ &= -2\int_{t_n}^{t_{n+1}} \sum_{l=1}^{N_U} \tilde{\theta}_l^{(k)}\xi_l^\intercal(x)R(x)\hat{u}^{(k)\prime}(x)dt - 2\int_{t_n}^{t_{n+1}} \sum_{l=N_U+1}^{\infty} \tilde{\theta}_l^{(k)}\xi_l^\intercal(x)R(x)\hat{u}^{(k)\prime}(x)dt \\ &- \int_{t_n}^{t_{n+1}} Q(x)dt - \int_{t_n}^{t_{n+1}} \hat{u}^{(k)\intercal}(x)R(x)\hat{u}^{(k)}(x)dt - 2\int_{t_n}^{t_{n+1}} \alpha^\intercal(x)\left(u^0(x) + e_L(t)\right)dt. \end{aligned} \tag{2.41}$$

Substituting the last three terms in (2.41) using (2.31) the following is derived

$$\begin{aligned}\sum_{l=1}^{N_V}\left(\tilde{\omega}_l^{(k)}-\omega^{(k)}\right)\left[\gamma_l(x(t_n))-\gamma_l(x(t_{n+1}))\right]\\ +2\int_{t_n}^{t_{n+1}}\sum_{l=1}^{N_U}\left(\tilde{\theta_l^{(k)}}-\theta_l^{(k)}\right)\xi_l^{\intercal}(x)R(x)\hat{u}^{(k)\prime}(x)dt+\tilde{\epsilon}_n^{(k)}=\epsilon_n^{(k)},\end{aligned} \tag{2.42}$$

where $\tilde{\epsilon}_n^{(k)}=\sum_{l=N_V+1}^{\infty}\tilde{\omega}_l^{(k)}\left[\gamma_l(x(t_{n+1}))-\gamma_l(x(t_n))\right]+2\int_{t_n}^{t_{n+1}}\sum_{l=N_U+1}^{\infty}\tilde{\theta}_l^{(k)}\xi_l^{\intercal}(x)R(x)\hat{u}^{(k)\prime}(x)dt$. Now (2.42) can be rewritten as

$$\epsilon_n^{(k)}=H^{(k)\intercal}\Theta_n^{(k)\intercal}+\tilde{\epsilon}_n^{(k)}, \tag{2.43}$$

where $H^{(k)\intercal}=\left(\begin{bmatrix}\tilde{\omega}_1^{(k)} & \cdots & \tilde{\omega}_{N_V}^{(k)} & \tilde{\theta}_1^{(k)} & \cdots & \tilde{\theta}_{N_U}^{(k)}\end{bmatrix}-\begin{bmatrix}\omega_1^{(k)} & \cdots & \omega_{N_V}^{(k)} & \theta_1^{(k)} & \cdots & \theta_{N_U}^{(k)}\end{bmatrix}\right)$ and $\Theta_n^{(k)}$ is defined as in (2.33). The error $\epsilon_n^{(k)}$ in (2.31) is minimized using a least square approach on a sequence of time intervals $\{t_n\}_{n=1}^{N_L}$, therefore it results that

$$\sum_{n=1}^{N_L}\epsilon_n^{(k)^2}\leq\sum_{n=1}^{N_L}\tilde{\epsilon}_n^{(k)^2}. \tag{2.44}$$

Note that by using (2.43) and the PE condition (2.32) the following holds true

$$\sum_{n=1}^{N_L}\left(\epsilon_n^{(k)}-\tilde{\epsilon}_n^{(k)}\right)^2=H^{(k)\intercal}\left(\sum_{n=1}^{N_L}\Theta_n^{(k)\intercal}\Theta_n^{(k)}\right)H^{(k)}\geq\beta^{PE}\left|H^{(k)}\right|^2. \tag{2.45}$$

Now using (2.44) the following expression is derived

$$\left|H^{(k)}\right|^2\leq\frac{1}{\beta^{PE}}\sum_{n=1}^{N_L}\left(\epsilon_n^{(k)}-\tilde{\epsilon}_n^{(k)}\right)^2\leq 2\sum_{n=1}^{N_L}\left(1-\epsilon_n^{(k)}\right)\tilde{\epsilon}_n^{(k)}. \tag{2.46}$$

If $N_V, N_U\to\infty$ then $\tilde{\epsilon}_n^{(k)}\to 0$ and $\left|H^{(k)}\right|\to 0$ from (2.46). Therefore, it can be concluded that $\hat{V}\to\tilde{V}$ and $\hat{u}\to\tilde{u}$ and, thus, (2.37) is proved.

As a consequence, it results also that given any $\epsilon>0$, two integers $N_V^0>0$ and $N_U^0>0$ can be found such that when $N_V>N_V^0$ and $N_U>N_U^0$ the following relations hold true

$$\begin{aligned}\left|\hat{V}^{(k)}(x)-\tilde{V}^k(x)\right|&\leq\sum_{l=1}^{N_V}\left|\omega^{(k)}-\tilde{\omega}^{(k)}\right|\left|\gamma_l(x)\right|+\sum_{l=N_V+1}^{\infty}\left|\tilde{\omega}^{(k)}\gamma_l(x)\right|\leq\epsilon,\\ \left|\hat{u}^{(k+1)}(x)-\tilde{u}^{k+1}(x)\right|&\leq\sum_{l=1}^{N_U}\left|\theta^{(k)}-\tilde{\theta}^{(k)}\right|\left|\xi_l(x)\right|+\sum_{l=N_U+1}^{\infty}\left|\tilde{\theta}^{(k)}\xi_l(x)\right|\leq\epsilon.\end{aligned} \tag{2.47}$$

Now Theorem 2 can be proved by induction, as follows.

(i) When $k=0$, it results that $\hat{u}^{(0)}(x)=u^{(0)}(x)$ and from (2.36) that $\tilde{V}^{(0)}(x)=V^{(0)}(x)$. Therefore by means of (2.37) the convergence is proved.

(ii) By assuming that the convergence is reached for $k-1$ (induction assumption), i.e., $lim_{N_V,N_U\to\infty}\hat{V}^{(k-1)}(x) = V^{(k-1)}(x)$ and $lim_{N_V,N_U\to\infty}\hat{u}^{(k)}(x) = u^{(k)}(x)$, the convergence has to be proved for k. Let us consider the system's trajectory in (2.27). Note that $u^{(k)}(x) + u^{(k)'}(x) = \hat{u}^{(k)}(x) + \hat{u}^{(k)'}(x)$ if $\hat{u}^{(k)'}(x) = u^{(0)}(x) + e_L(t) - \hat{u}^{(k)}(x)$. Therefore it follows that

$$u^{(k)'}(x) = \hat{u}^{(k)}(x) + \hat{u}^{(k)'}(x) - u^{(k)}(x). \tag{2.48}$$

By using this last equation, (2.28) can be rewritten as

$$\begin{aligned}\dot{V}^{(k)}(x) &= -U(x, u^{(k)}(x)) - \left[2u^{(k+1)\intercal}(x)R(x) + \alpha^{\intercal}(x)\right] u^{(k)'}(x)\\ &= -Q(x) - u^{(k)\intercal}(x)R(x)u^{(k)}(x) - 2u^{(k+1)\intercal}(x)R(x)\left(\hat{u}^{(k)}(x) - u^{(k)}(x)\right)\\ &\quad - 2u^{(k+1)\intercal}(x)R(x)\hat{u}^{(k)'}(x) - \alpha^{\intercal}(x)\left(\hat{u}^{(k)}(x) + \hat{u}^{(k)'}(x)\right),\end{aligned} \tag{2.49}$$

and (2.38) can be rewritten as

$$\begin{aligned}\dot{\tilde{V}}^{(k)}(x) &= -U(x, \hat{u}^{(k)}(x)) - \left[2\tilde{u}^{(k+1)\intercal}(x)R(x) + \alpha^{\intercal}(x)\right] \hat{u}^{(k)'}(x)\\ &= -Q(x) - \hat{u}^{(k)\intercal}(x)R(x)\hat{u}^{(k)}(x) - 2\tilde{u}^{(k+1)\intercal}(x)R(x)\hat{u}^{(k)'}(x)\\ &\quad - \alpha^{\intercal}(x)\left(\hat{u}^{(k)}(x) + \hat{u}^{(k)'}(x)\right).\end{aligned} \tag{2.50}$$

Then, by considering that the state's trajectory is derived from the same system, i.e., (2.26), it results that

$$\begin{aligned}\left|V^{(k)}(x) - \tilde{V}^{(k)}(x)\right| &\leq \left|\int_t^\infty \left[\hat{u}^{(k)\intercal}(x)R(x)\hat{u}^{(k)}(x) - u^{(k)\intercal}(x)R(x)u^{(k)}(x)\right] dt\right|\\ &\quad + 2\left|\int_t^\infty u^{(k+1)\intercal}(x)R(x)\left(\hat{u}^{(k)}(x) - u^{(k)}(x)\right) dt\right|\\ &\quad + 2\left|\int_t^\infty \left[\tilde{u}^{(k+1)\intercal}(x) - u^{(k+1)\intercal}(x)\right] R(x)\hat{u}^{(k)'}(x)dt\right|.\end{aligned} \tag{2.51}$$

Note that for the induction assumption the first two terms on the right side of (2.51) tend to zero when $N_V, N_U \to \infty$. Also, the PE assumption ensures that $\lim_{N_V,N_U\to\infty}\left|\tilde{u}^{(k+1)}(x) - u^{(k+1)}(x)\right| = 0$. Therefore it can be concluded that

$$\lim_{N_V,N_U\to\infty}\left|V^{(k)}(x) - \tilde{V}^{(k)}(x)\right| = 0. \tag{2.52}$$

However, to prove the convergence of $\hat{V}^{(k)}(x)$ to the solution of the (2.9), i.e., $V^{(k)}(x)$, note that

$$\left|\hat{V}^{(k)}(x) - V^{(k)}(x)\right| \leq \left|V^{(k)}(x) - \tilde{V}^{(k)}(x)\right| + \left|\tilde{V}^{(k)}(x) - \hat{V}^{(k)}(x)\right|. \tag{2.53}$$

Therefore, by using (2.37) and (2.52) it follows that

$$\lim_{N_V,N_U\to\infty}\left|\hat{V}^{(k)}(x) - V^{(k)}(x)\right| = 0. \tag{2.54}$$

It can be proved in a similar way that also $\lim_{N_V,N_U\to\infty}\left|\hat{u}^{(k+1)}(x) - u^{(k+1)}(x)\right| = 0$, and, thus, the proof of Theorem 2 is concluded.

2.4.3 Implementation

The PI procedure in Algorithm 2.1 solves the HJB equation in an iterative way. Therefore, once the convergence of $\hat{V}^{(k)}$ and $\hat{u}^{(k+1)}(x)$ to $V^{(k)}$ and $u^{(k+1)}(x)$ has been ensured, an approximated solution of the HJB equation can be found. Let's consider the following quantities

$$\begin{aligned}
\Delta\Gamma(t_{n+1}) &= \Gamma(x(t_{n+1})) - \Gamma(x(t_n)) \in \mathbb{R}^{N_V}, \\
\Phi(t_{n+1}) &= \int_{t_n}^{t_{n+1}} \Xi(x)R(x)\Xi^{\intercal}(x)dt \in \mathbb{R}^{N_U \times N_U}, \\
\Psi(t_{n+1}) &= \int_{t_n}^{t_{n+1}} \Xi(x)R(x)\left(u^{(0)}(x) + e_L(t)\right)dt \in \mathbb{R}^{N_U}, \\
Q_I(t_{n+1}) &= \int_{t_n}^{t_{n+1}} Q(x)dt \in \mathbb{R}, \\
A_I(t_{n+1}) &= \int_{t_n}^{t_{n+1}} \alpha^{\intercal}(x)\left(u^{(0)}(x) + e_L(t)\right)dt \in \mathbb{R}.
\end{aligned} \tag{2.55}$$

The approximated IRL equation in (2.31) can be rewritten as

$$\begin{aligned}
\omega^{(k)\intercal}\Delta\Gamma(t_{n+1}) = &-\theta^{(k-1)\intercal}\Phi(t_{n+1})\theta^{(k-1)} - 2\theta^{(k)\intercal}\left(\Psi(t_{n+1}) - \Phi(t_{n+1})\theta^{(k-1)}\right) \\
&- Q_I(t_{n+1}) - A_I(t_{n+1}).
\end{aligned} \tag{2.56}$$

The following algorithm implements the ADP PI procedure with off-policy learning, where the unknown weights are found using the least-square approach.

Algorithm 2.2 shows the ADP with off-policy learning scheme. Once the convergence is obtained, the probing noise is removed and the control policy switches to the approximated optimal controller. Note that NNs approximate nonlinear functions only on compact sets and not on the entire system's state space. Therefore, the domain of attraction (DoA) of the resulting closed-loop system with the newly learned control policy should be determined. The approximated optimal policy is applied once the state's trajectory goes inside the DoA. Several methods can be used to estimate the DoA [65–68]. Nevertheless, by using an appropriate probing noise, the DoA can be made sufficiently large to include the state's region practically used by the system.

In general, the off-policy method benefits of an easier implementation with less computational requirements when compared with the on-policy procedure. Moreover, with an appropriate system representation and proper choice of the actor's NN activating functions, the off-policy method can be used as a tool to solve offline the HJB equation and then implement the optimal policy using standard techniques, such as Proportional-Integral-Derivative (PID) controllers. Finally, off-policy methods provide approximated optimal solutions when the system dynamics is fully unknown.

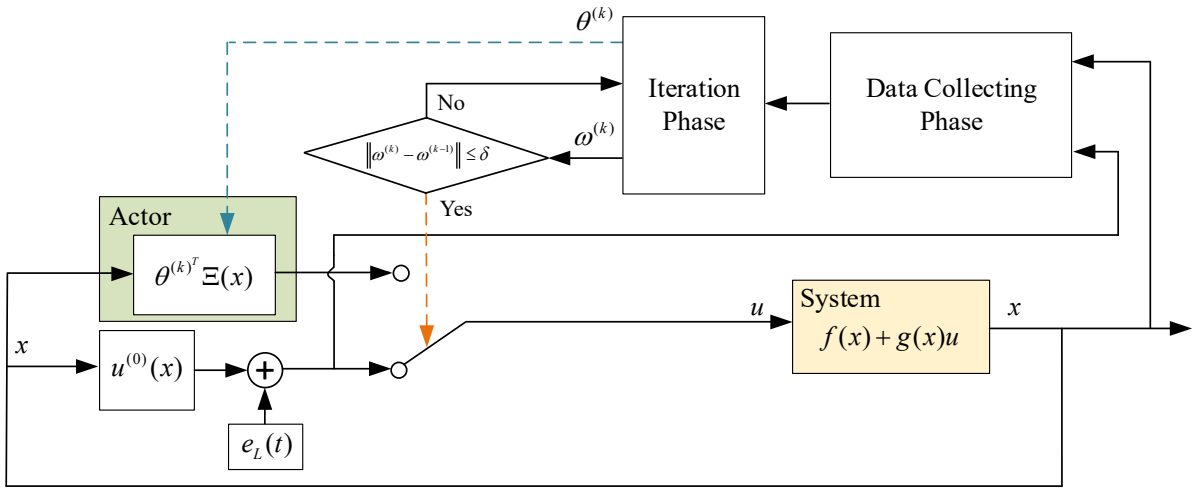

Figure 2.4: ADP with off-policy learning scheme.

Algorithm 2.2 ADP PI Algorithm with off-policy learning

1. Initialization: Define the initial stable control policy, $u^{(0)}(x) = \theta^{(0)\intercal}\Xi(x)$, the exploration noise, $e_L(t)$, the initial weights, $\omega^{(0)}$, the number of learning time intervals N_L, and a small positive constant, δ. Set $k = 1$.

2. Data Collecting Phase: Apply the input $\left(u^{(0)}(x) + e_L(t)\right)$ to the system and record $\Delta\Gamma(t_n)$, $\Phi(t_n)$, $\Psi(t_n)$, $Q_I(t_n)$, and $A_I(t_n)$, for $n = 1, ..., N_L$. Define the following matrices

$$\begin{aligned} X_\Gamma &= \left[\Delta\Gamma^\intercal(t_1) \cdots \Delta\Gamma^\intercal(t_{N_L})\right]^\intercal \in \mathbb{R}^{N_L \times N_V} \\ B_Q &= -\left(\left[Q_I(t_1) \cdots Q_I(t_{N_L})\right] + \left[A_I(t_1) \cdots A_I(t_{N_L})\right]\right)^\intercal \in \mathbb{R}^{N_L} \end{aligned} \tag{2.57}$$

3. Iteration Phase:

a. **Data Evaluation:** Evaluate the following matrices

$$\begin{aligned} X_1 &= 2\begin{bmatrix} \Psi^\intercal(t_1) - \theta^{(k-1)\intercal}\Phi^\intercal(t_1) \\ \vdots \\ \Psi^\intercal(t_1) - \theta^{(k-1)\intercal}\Phi^\intercal(t_1) \end{bmatrix} \in \mathbb{R}^{N_L \times N_U} \\ B_\Phi &= -\left[\theta^{(k-1)\intercal}\Phi(t_1)\theta^{(k-1)} \quad \cdots \quad \theta^{(k-1)\intercal}\Phi(t_{N_L})\theta^{(k-1)}\right]^\intercal \in \mathbb{R}^{N_L} \end{aligned} \tag{2.58}$$

b. **Policy Improvement:** Find $\omega^{(k)}$ and $\theta^{(k)}$ from the following least square problem

$$\begin{bmatrix} X_\Gamma & X_1 \end{bmatrix} \begin{bmatrix} \omega^{(k)} \\ \theta^{(k)} \end{bmatrix} = B_Q + B_\Phi \tag{2.59}$$

4. Off-policy Iteration: If $\left|\left|\omega^{(k)} - \omega^{(k-1)}\right|\right| \geq \delta$ set $k = k+1$ and repeat Step 3. Otherwise, stop and return the approximated optimal value function and control policy, i.e., $\omega^{(k)}$ and $\theta^{(k)}$.

2.5 Examples

The two following examples show how the ADP algorithm with both on-policy and off-policy learning effectively solves the optimal control problem for linear and nonlinear systems.

2.5.1 Linear System

Let's consider the following two-dimensional linear system

$$\dot{x} = \begin{bmatrix} -1 & -2 \\ 2 & -4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{2.60}$$

with the utility function defined as

$$U(x, u) = x^\intercal Q x + u^\intercal R u + x^\intercal N u, \tag{2.61}$$

where $Q = \begin{bmatrix} 4 & 0 \\ 0 & 3 \end{bmatrix}$, $R = 2$, and $N = \begin{bmatrix} 1 & 2 \end{bmatrix}^\intercal$.

The optimal controller is in the linear state-feedback form $u^*(x) = -K^* x$, where K^* is the unknown optimal feedback matrix, while the optimal value function is in the form $V^*(x) = x^\intercal P^* x$. By using the Riccati approach it results that

$$\begin{aligned} K^* &= \begin{bmatrix} 0.2812 & 1.1357 \end{bmatrix}, \\ P^* &= \begin{bmatrix} 1.0458 & -0.4375 \\ -0.4375 & 0.2713 \end{bmatrix}. \end{aligned} \tag{2.62}$$

The ADP with on-policy learning scheme depicted in Fig. 2.2 is applied to system (2.60). The learning rate λ_V is set to 5, and the initial value of ω is $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^\intercal$. A white probing noise guarantees the PE condition. The critic NN activating functions vector and its gradient are

$$\begin{aligned} \Gamma(x) &= \begin{bmatrix} x_1^2 & x_1 x_2 & x_2^2 \end{bmatrix}^\intercal, \\ \nabla\Gamma(x) &= \begin{bmatrix} 2x_1 & 0 \\ x_2 & x_1 \\ 0 & 2x_2 \end{bmatrix}, \end{aligned} \tag{2.63}$$

Figure 2.5 elaborates the results. The critic NN weights converge in about $70s$, as in Fig. 2.5(b), to $\omega = \begin{bmatrix} 1.042 & -0.871 & 0.273 \end{bmatrix}^\intercal$, with a small error with respect to the optimal values in (2.62). After $70s$ the probing noise is removed and the learned near-optimal policy is applied, as in Fig. 2.5(a). The optimality of the learned solution can be quantified through the error on the Hamiltonian, i.e., ϵ_H in (2.14).

As shown in Fig. 2.5(c), the Hamiltonian exhibits higher values when the critic weights have not reached the convergence yet. Once the critic weights are close to the optimal ones, the Hamiltonian tends to zero, i.e., the optimality is achieved. Figure 2.5(d) shows how the PE condition in (2.23) is satisfied throughout the experiment, i.e., the eigenvalues of the matrix $\int_0^t \bar{\sigma_V}(\tau)\bar{\sigma_V}^\intercal(\tau)d\tau \in \mathbb{R}^{3\times 3}$ are always positive for any $t \in [0s, 70s]$. Finally, Fig. 2.5(e) and Fig. 2.5(f) depict the approximated optimal value function and its approximation error, respectively.

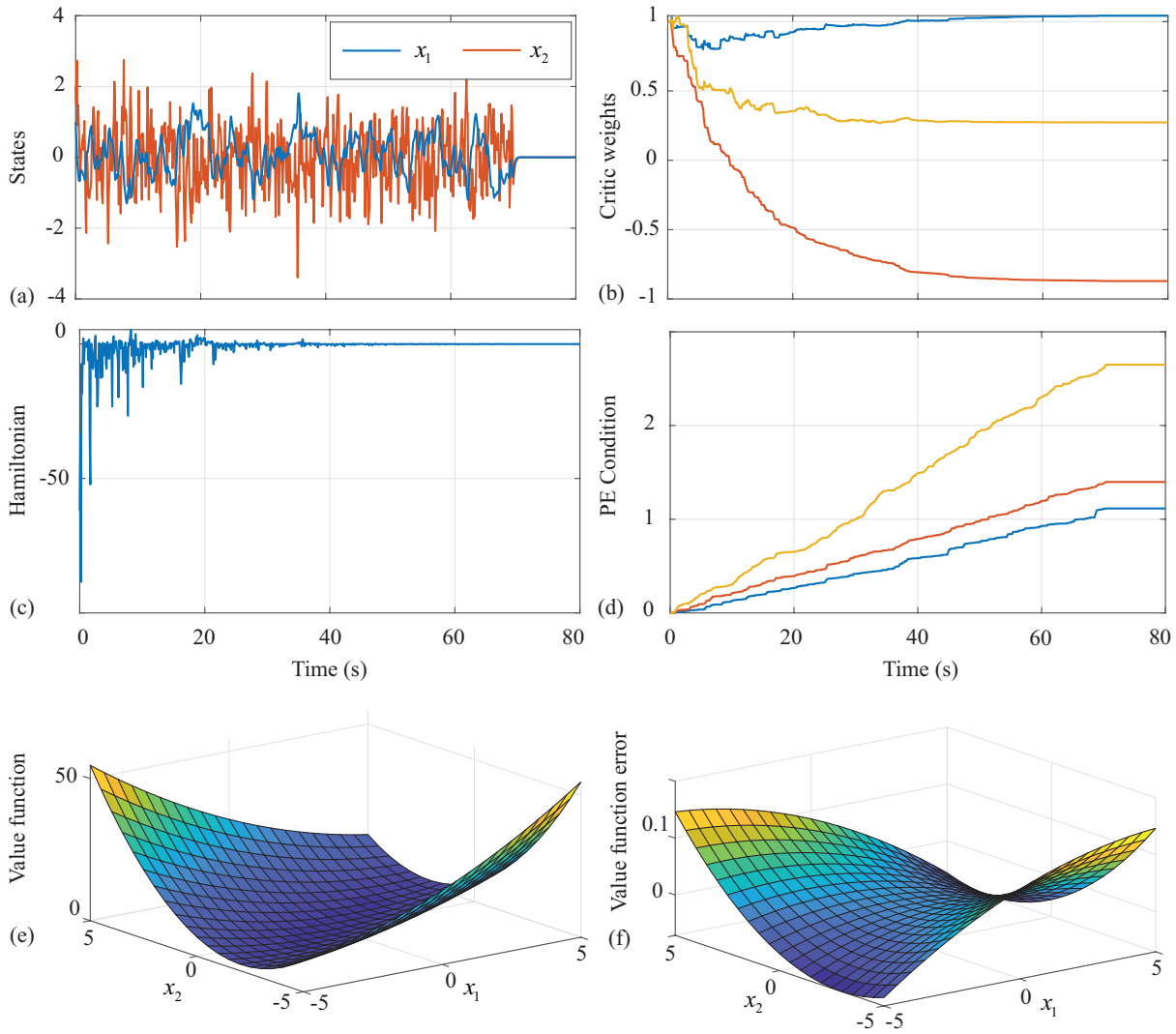

Figure 2.5: Results of the ADP with on-policy learning algorithm when applied to a linear system: (a) States trajectory during the learning experiment; (b) Convergence of the critic NN weights, ω; (c) Error on the Hamiltonian, ϵ_H, throughout the experiment; (d) Eigenvalues of the matrix in (2.23), i.e., the PE condition, during the experiment; (e) Approximated optimal value function; (f) Approximation error for the value function.

The same optimal control problem is solved using the ADP with off-policy learning procedure in Algorithm 2.2. The optimal value function is approximated using the same $\Gamma(x)$ in (2.63), while the optimal control policy is approximated using the basis functions $\Xi(x) = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\intercal$. Due to the open-loop stability of system (2.60), the initial control policy is set to $u^0(x) = 0$. The same white noise employed with the on-policy procedure is used as probing input. The number of learning time intervals is $N_L = 1000$, each of $0.1s$.

Figure 2.6 shows the results of the off-policy algorithm. After performing the data collecting phase, the actor and critic NN weights converge after 2 iterations, as in Fig. 2.6(a) and Fig. 2.6(b), where 10 iterations are executed. The approximated optimal weights are $\theta^{(10)} = \begin{bmatrix} -0.281 & -1.136 \end{bmatrix}^\intercal$ for the actor, and $\omega^{(10)} = \begin{bmatrix} 1.046 & -0.875 & 0.271 \end{bmatrix}^\intercal$ for the critic,

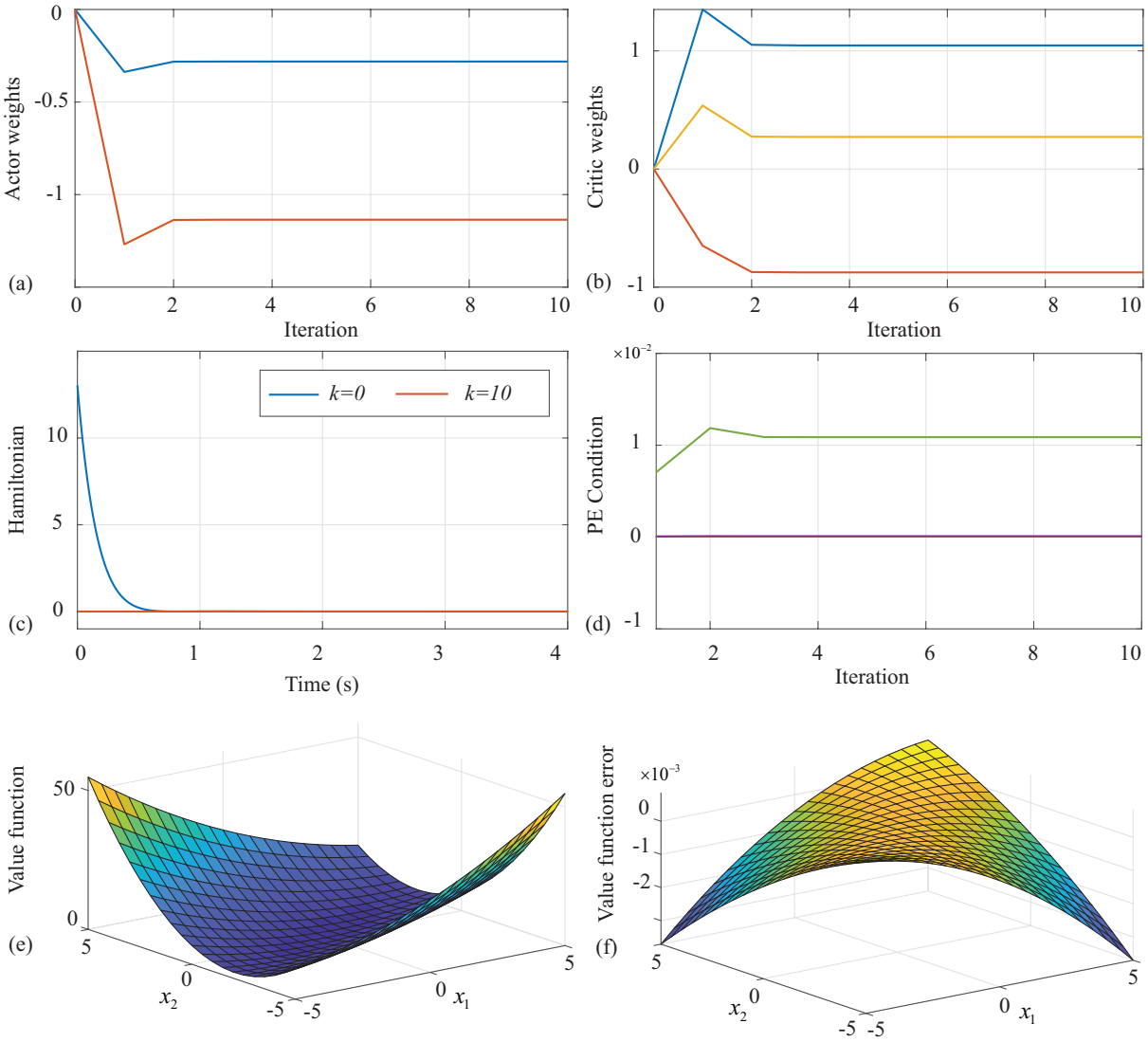

Figure 2.6: Results of the ADP with off-policy learning algorithm when applied to a linear system: (a) Convergence of the actor NN weights, $\theta^{(k)}$; (b) Convergence of the critic NN weights, $\omega^{(k)}$; (c) Resulting Hamiltonian for the initial and obtained policy; (d) Eigenvalues of the matrix in (2.32), i.e., the PE condition, during each iteration; (e) Approximated optimal value function; (f) Approximation error for the value function.

with practically no error if compared with the optimal values in (2.62). The optimality of the learned solution can be quantified through the Hamiltonian. As shown in Fig. 2.6(c), the Hamiltonian exhibits higher values when the initial controller, i.e., $u^{(0)}(x)$, is employed. The learned approximated optimal policy, i.e., $u^{(10)}(x)$, makes the Hamiltonian equal to zero, i.e., the optimality is achieved. Figure 2.6(d) shows how the PE condition in (2.32) is satisfied throughout the experiment, i.e., the eigenvalues of the matrix $\frac{1}{N_L}\sum_{n=0}^{N_L}\Theta_n^{(k)\intercal}\Theta_n^{(k)} \in \mathbb{R}^{5\times 5}$ are always positive for any iteration. Finally, Fig. 2.6(e) and Fig. 2.6(f) depict the approximated optimal value function and the error with respect to the optimal one in (2.62), respectively.

2.5.2 Nonlinear System

The two ADP algorithms deal also with nonlinear systems. Let's consider the following nonlinear dynamics

$$\dot{x} = \begin{bmatrix} -x_1 + x_2 \\ -0.5x_1 - 0.5x_2\left[1 - (cos(2x_1) + 2)\right]^2 \end{bmatrix} + \begin{bmatrix} 0 \\ cos(2x_1) + 2 \end{bmatrix} u. \tag{2.64}$$

The utility function is selected as

$$U(x, u) = x^\intercal Q x + u^\intercal R u, \tag{2.65}$$

with $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $R = 1$. For such system, the optimal value function is given in [29] as

$$V^*(x) = \frac{1}{2}x_1^2 + x_2^2. \tag{2.66}$$

First, the ADP with on-policy learning algorithm is employed. The learning rate, the initial value of ω, and the probing noise are set as in the previous example. The critic NN activating functions vector and its gradient are the same as in (2.63).

Figure 2.7 shows the results. As in the previous example, the critic NN weights converge in about $70s$, as depicted in Fig. 2.7(b), to $\omega = \begin{bmatrix} 0.49 & 0.02 & 0.98 \end{bmatrix}^\intercal$, close to the optimal values in (2.66). After $70s$ the probing noise is removed and the learned near-optimal policy is applied, as in Fig. 2.7(a). The error on the Hamiltonian quantifies the optimality of the learned solution. In fact, in Fig. 2.7(c), the Hamiltonian exhibits higher values when the critic weights are far from converging. Once the critic weights are close to the optimal ones, the Hamiltonian tends to zero and the optimality is reached. Figure 2.7(d) shows how the PE condition in (2.23) is satisfied throughout the experiment, i.e., the eigenvalues of the matrix $\int_0^t \bar{\sigma_V}(\tau)\bar{\sigma_V}^\intercal(\tau)d\tau \in \mathbb{R}^{3\times 3}$ are always positive for any $t \in [0s, 70s]$. Finally, Fig. 2.5(e) and Fig. 2.5(f) depict the approximated optimal value function and the error with respect to the optimal one in (2.66), respectively.

The same optimal control problem is solved using the ADP with off-policy learning. Since the utility function is quadratic, the same $\Gamma(x)$ in (2.63) is chosen to approximate the optimal value function. Due to the system's nonlinearity, the optimal control input is expected to be nonlinear, thus nonlinear polynomial terms are chosen to approximate the optimal control policy, i.e., $\Xi(x) = \begin{bmatrix} x_1 & x_2 & x_1^2 & x_2^2 & x_1x_2 & x_1^4 & x_2^4 \end{bmatrix}^\intercal$. $u^0(x) = -x_1 - 2x_2$ is used as the initial stable feedback policy. The probing noise, the number and length of the learning time intervals are set as in the previous example.

Figure 2.8 elaborates the results of the off-policy algorithm. After performing the data collecting phase, the actor and critic NN weights converge in 6 iterations, as in Fig. 2.8(a) and Fig. 2.8(b), where 10 iterations are executed. The optimal policy is approximated through the weights $\theta^{(10)} = \begin{bmatrix} -1.063 & -3.235 & 1.677 & 0.204 & 1.250 & -0.405 & -0.013 \end{bmatrix}^\intercal$, while $\omega^{(10)} = \begin{bmatrix} 0.561 & -0.025 & 1.015 \end{bmatrix}^\intercal$ approximates the optimal value function, with a small error if compared with the optimal values in (2.66). The optimality of the learned solution can be quantified through the Hamiltonian. As shown in Fig. 2.8(c), the Hamiltonian exhibits higher values when the initial controller, i.e., $u^{(0)}(x)$, is employed. The learned approximated optimal policy, i.e., $u^{(10)}(x)$, makes the Hamiltonian equal to zero. Figure 2.8(d) shows how the PE condition in (2.32) is satisfied throughout the experiment, i.e., the eigenvalues of the matrix

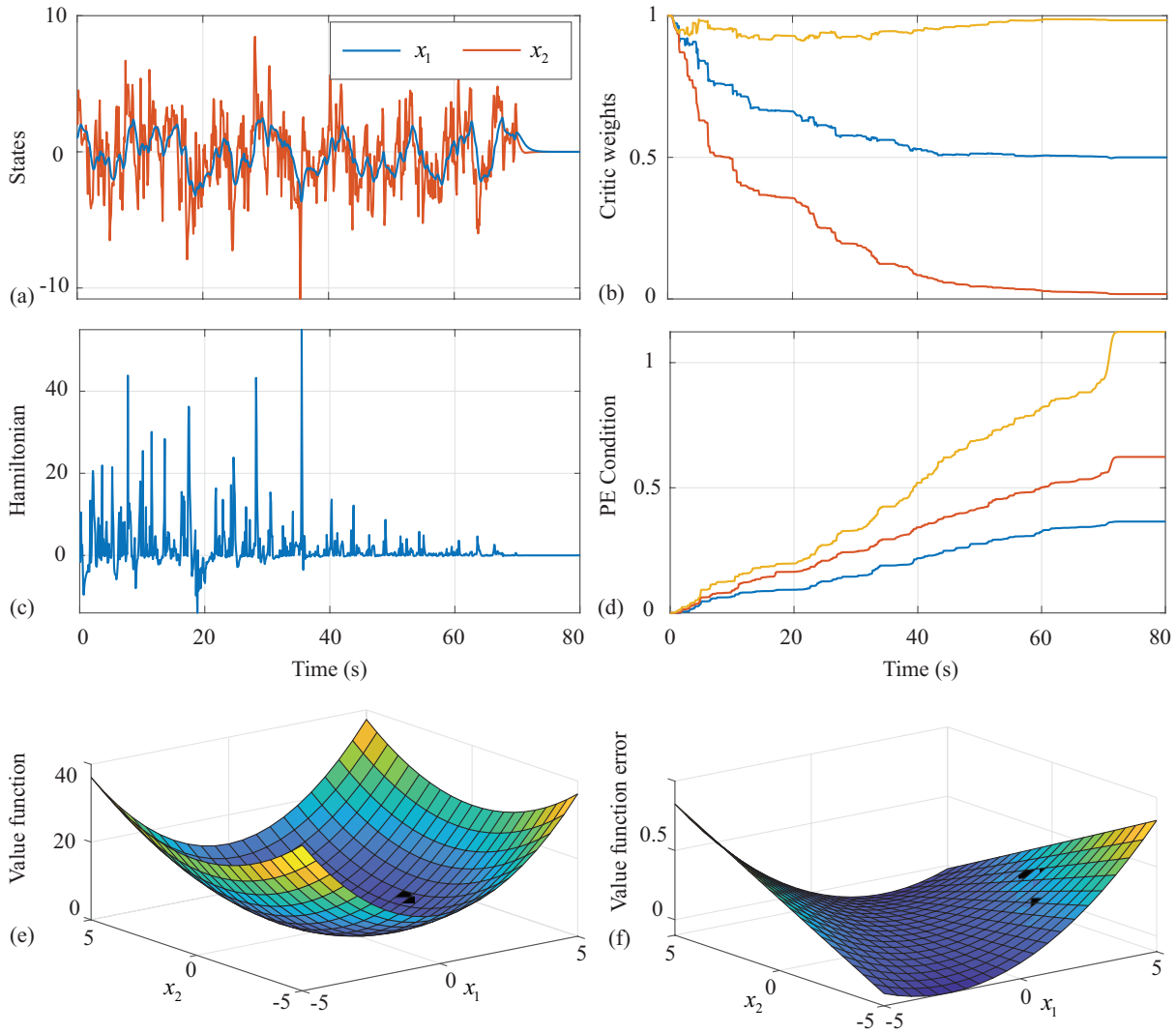

Figure 2.7: Results of the ADP with on-policy learning algorithm when applied to a nonlinear system: (a) States trajectory during the learning experiment; (b) Convergence of the critic NN weights, ω; (c) Error on the Hamiltonian, ϵ_H, throughout the experiment; (d) Eigenvalues of the matrix in (2.23), i.e., the PE condition, during the experiment; (e) Approximated optimal value function; (f) Approximation error for the value function.

$\frac{1}{N_L}\sum_{n=0}^{N_L}\Theta_n^{(k)\intercal}\Theta_n^{(k)} \in \mathbb{R}^{10\times 10}$ are always positive for any iteration. Finally, Fig. 2.8(e) and Fig. 2.8(f) depict the approximated optimal value function and the error with respect to the optimal one in (2.66), respectively. Note that the off-policy procedure does not assume the knowledge of any system function. However, if $g(x)$ is known (usually it can be easily identified), some of the approximating functions in $\Xi(x)$ can be inspired by $g(x)$, improving the overall approximation with a less computational expense.

The next section presents a more complex problem: the structured optimal control of symmetrically coupled linear systems. It is shown how the combination of traditional optimization and ADP with off-policy learning provides an effective tool to solve such problems when the dynamics is partially-unknown.

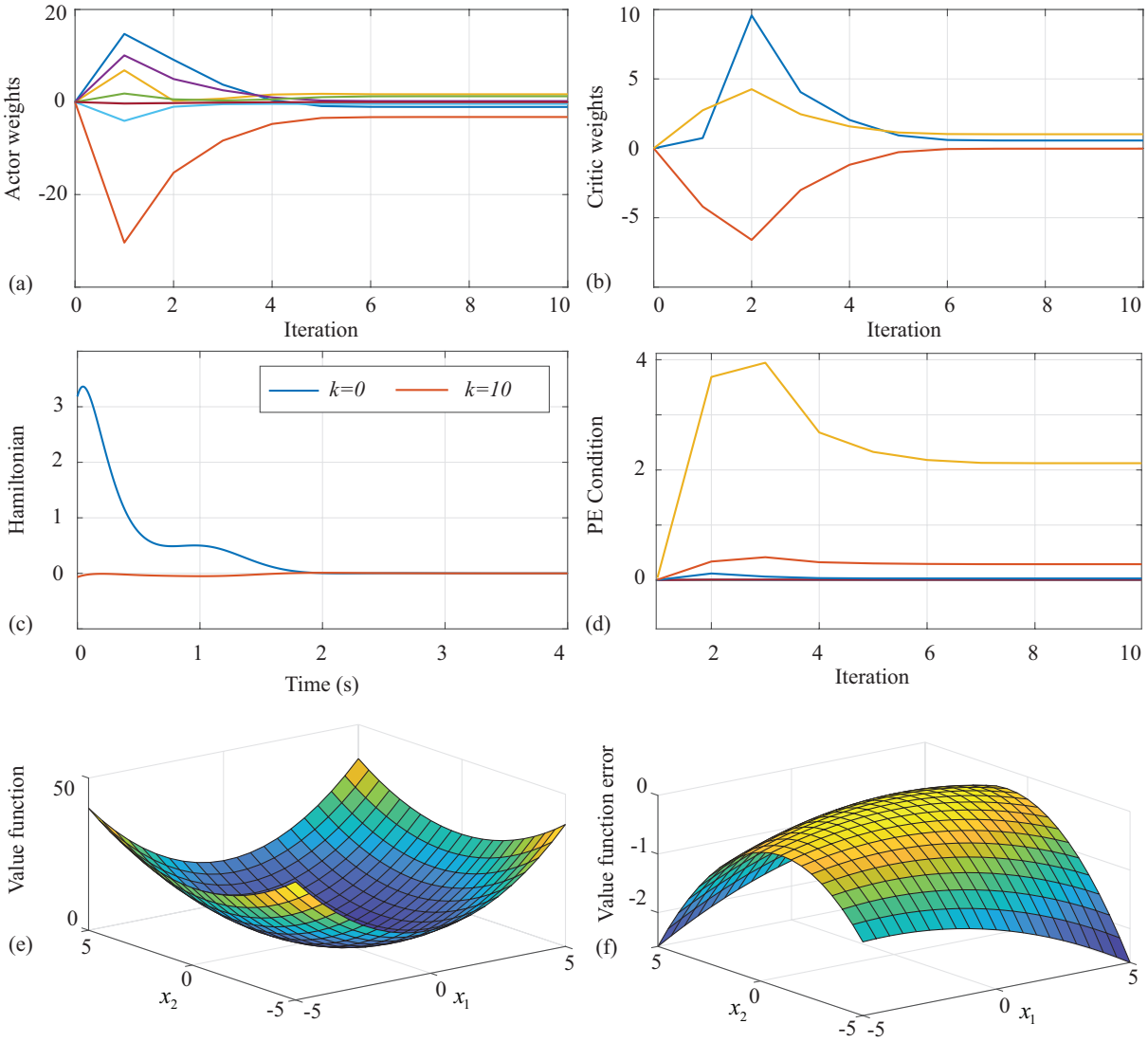

Figure 2.8: Results of the ADP with off-policy learning algorithm when applied to a nonlinear system: (a) Convergence of the actor NN weights, $\theta^{(k)}$; (b) Convergence of the critic NN weights, $\omega^{(k)}$; (c) Resulting Hamiltonian for the initial and obtained policy; (d) Eigenvalues of the matrix in (2.32), i.e., the PE condition, during each iteration; (e) Approximated optimal value function; (f) Approximation error for the value function.

2.6 Structured optimal control via ADP

2.6.1 Problem Statement

Let's consider a set of N interconnected first-order systems, each described by the following dynamics

$$\dot{x}_i = a_i x_i + \sum_{j=1}^{N} a_{ij} x_j + b_i u_i \quad i = 1, ..., N, \tag{2.67}$$

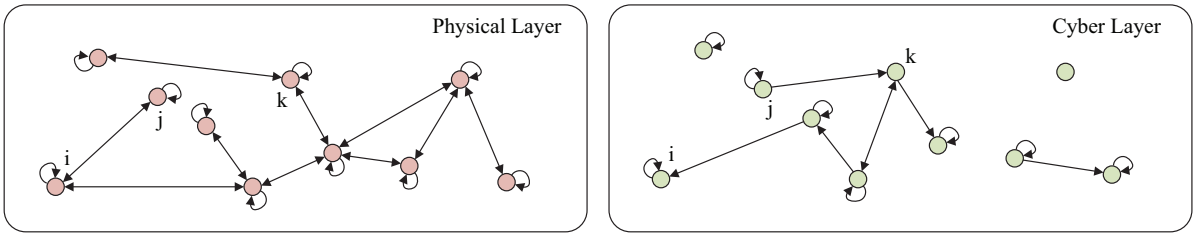

Figure 2.9: An interconnected system with cyber and physical layers.

where $x_i \in \mathbb{R}$ and $u_i \in \mathbb{R}$ are the state and the input of system i, respectively, while a_{ij} represents the coupling gain between systems i and j. Let us assume symmetric couplings, i.e., $a_{ij} = a_{ji}, \forall i, j$. The overall dynamics is given by

$$\dot{x} = Ax + Bu, \tag{2.68}$$

where $A = A^\intercal \in \mathbb{R}^{N\times N}$, $x = [x_1 \cdots x_N]^\intercal \in \mathbb{R}^N$, $u = [u_1 \cdots u_N]^\intercal \in \mathbb{R}^N$, and $B = \mathrm{diag}([b_1 \cdots b_N]) \in \mathbb{R}^{N\times N}$. System (2.68) is assumed to be partially unknown, i.e., symmetric matrix A is unknown while matrix B is available. Let the state x be fully accessible for feedback purposes. No further assumptions are made on the open-loop stability.

Let L be the subspace embedding some structural constraints, i.e., some entries in specified locations of every matrix $K \in L$ are zero. The goal is to find an LQR feedback controller of the form

$$u = -Kx, \quad K \in L \subset \mathbb{R}^{N\times N}, \tag{2.69}$$

that minimizes a given performance function expressed as

$$J = \int_0^\infty (x^\intercal Q x + u^\intercal R u)\, dt, \tag{2.70}$$

where $Q = Q^\intercal$ is positive semi-definite and $R = R^\intercal$ is positive definite. It is also assumed that $(\sqrt{Q}, A)$ is detectable and system (2.68) is stabilizable, i.e., there exists $K \in L$ such that $A - BK$ is a Hurwitz matrix.

As shown in Fig. 2.9, the physical layer of the N interconnected systems is defined according to a weighted undirected graph with self loops, where A is the adjacency matrix. The optimal $K \in L$ that minimizes (2.70) is structured according to a cyber layer over which communication and control are carried out. This cyber configuration is represented by a directed graph with the adjacency matrix $\mathcal{A}_C \in \mathbb{R}^{N\times N}$, where $(\mathcal{A}_C)_{ij} = 1$ if u_i depends on x_j, i.e., $(K)_{ij}$ is allowed to be nonzero, and the system j can send its state information to system i; Otherwise, $(\mathcal{A}_C)_{ij} = 0$. Once $\mathcal{A}_C$ is defined, the subspace L can be easily expressed as

$$L = \left\{ M \in \mathbb{R}^{N\times N} | M \circ (1_{N\times N} - \mathcal{A}_C) = 0_{N\times N} \right\}, \tag{2.71}$$

where $\circ$ denotes the Hadamard product, and $1_{N\times N}$,$0_{N\times N}$ are $N \times N$ matrices of 1 and 0, respectively.

2.6.2 Necessary Conditions for Optimal Structured Feedback

Given any fixed feedback stabilizable controller K, and any initial state of system (2.68), x_0, the resulting cost (2.70) is given by $J = x_0^\intercal P x_0$, where $P = P^\intercal \geq 0$ is the solution of the

following Lyapunov equation [6]

$$l = (A - BK)^\intercal P + P(A - BK) + Q + K^\intercal RK = 0. \tag{2.72}$$

If the solution to (2.72) exists, and $A - BK$ is Hurwitz, the cost J obtained by applying the feedback controller $u = -Kx$ can be computed without solving the closed-loop dynamics. However, the dependency of the final cost on the initial state (i.e., $J = x_0^\intercal P x_0$) makes the objective function explicitly depend on x_0, which may be unknown. To overcome this issue, a common way [69] is to minimize the expected value of the performance index, denoted as $E\{J\}$, as follows

$$E\{J\} = E\{x_0^\intercal P x_0\} = \mathrm{Tr}(PX_0), \tag{2.73}$$

where $\mathrm{Tr}(\cdot)$ is the trace operator, and $X_0 = E\{x_0^\intercal x_0\}$ is an $N \times N$ symmetric matrix representing the initial auto-correlation of the initial state. X_0 provides a description of the surface where x_0 is uniformly distributed, e.g., X_0 is the identity matrix when the initial states are uniformly distributed on a sphere with a unitary radius.

The structured feedback optimal control problem can now be defined as

$$\begin{aligned} \text{minimize} \quad & J = \mathrm{Tr}(PX_0) \\ \text{s.t.} \quad & A_C^\intercal P + PA_C + Q + K^\intercal RK = 0 \\ & K \in L \end{aligned} \tag{2.74}$$

where $A_C = A - BK$. The necessary conditions for the solution of (2.74) are derived by using the Lagrange multiplier approach. Let us define the Hamiltonian $\mathcal{H}$ by adjoining the first constraint to the objective function

$$\mathcal{H} = \mathrm{Tr}(PX_0) + \mathrm{Tr}(lS), \tag{2.75}$$

where l is the Lyapunov equation (2.72), and $S \in \mathbb{R}^{N\times N}$ is a symmetric matrix of Lagrange multipliers to be determined. P, S, and $K \in L$ represent three unknown matrices. The first two necessary conditions for optimality are easily derived as

$$\frac{\partial \mathcal{H}}{\partial S} = l = (A - BK)^\intercal P + P(A - BK) + Q + K^\intercal RK = 0, \tag{2.76}$$

$$\frac{\partial \mathcal{H}}{\partial P} = (A - BK)S + S(A - BK)^\intercal + X_0 = 0. \tag{2.77}$$

As in [70, 71], the third necessary condition can be derived by defining the control matrix as $K = (K_F \circ \mathcal{A}_C) \in L$, where $K_F \in \mathbb{R}^{N\times N}$ is an arbitrary free matrix. By defining the set $\Xi = \{(i,j)|(\mathcal{A}_C)_{ij} = 1\}$, the following equality holds

$$K = (K_F \circ \mathcal{A}_C) = \sum_{(i,j)\in\Xi} \Omega_i K_F \Omega_j, \tag{2.78}$$

where $\Omega_k \in \mathbb{R}^{N\times N}$ is zero everywhere except for $(\Omega_k)_{kk} = 1$. Optimizing with respect to the free elements of K is now equivalent to optimizing with respect to K_F. Thus, the third necessary condition is derived as follow

$$\frac{\partial \mathcal{H}}{\partial K_F} = 2 \sum_{(i,j)\in\Xi} \Omega_i \left(RKS - B^\intercal PS\right) \Omega_j = 2\left(RKS - B^\intercal PS\right) \circ \mathcal{A}_C = 0 \tag{2.79}$$

In summary, the three necessary conditions for optimality are given by the two Lyapunov equations (2.76) and (2.77), and by condition (2.79). To find the optimal K, these three coupled equations need to be solved. Without the structured constraint, (2.74) results in the well-known regular state-feedback LQR problem with fully centralized structure. In such case, the matrix S is no longer required and the optimization problem is independent from the initial state.

2.6.3 Data-driven Solution of Lyapunov Equations

Inspired by the IRL approach in (2.31), the two Lyapunov equations (2.76), (2.77) can be solved without the knowledge of the system matrix A. As in the ADP with off-policy learning procedure, let us consider a control input $u^{(0)} = \hat{u}^{(0)}(x) + e_L(t)$ composed by a feedback controller $\hat{u}^{(0)}(x)$ and an exogenous exploration noise $e_L(t)$. We assume that the resulting closed-loop system given by (2.68) with $u = u^{(0)}$ is stable. For any fixed feedback matrix K, the following relation holds

$$\dot{x} = Ax + Bu^{(0)} = A_C x + B(u^{(0)} + Kx), \tag{2.80}$$

The time-derivative of the term $x^\intercal Px$ along the trajectories of (2.80) is

$$\begin{aligned} \frac{d}{dt}(x^\intercal Px) &= x^\intercal (A_C^\intercal P + PA_C)x + 2(u^{(0)} + Kx)^\intercal B^\intercal Px \\ &= -x^\intercal \underbrace{(Q + K^\intercal RK)}_{\tilde{Q}(K)} x + 2(u^{(0)} + Kx)^\intercal B^\intercal Px, \end{aligned} \tag{2.81}$$

where the Lyapunov equation (2.76) is used to replace the term $x^\intercal(A_c^\intercal P + PA_C)x$ with $-x^\intercal(Q + K^\intercal RK)x$. In this way, (2.81) no longer depends on the unknown matrix A, and can be solved by using collected measurements along the trajectories of (2.80) once matrices K and B are given.

Integrating both sides of (2.81) over the time interval $[t_n, t_{n+1}]$ leads to

$$x^\intercal(t_{n+1})Px(t_{n+1}) - x^\intercal(t_n)Px(t_n) = -\int_{t_n}^{t_{n+1}} x^\intercal \tilde{Q}(K)x dt + 2\int_{t_n}^{t_{n+1}} (u^{(0)} + Kx)^\intercal B^\intercal Px dt. \tag{2.82}$$

By employing the properties of the vector operator and Kronecker product [72] it results that

$$\begin{aligned} x^\intercal Px &= (x^\intercal \otimes x^\intercal)\mathrm{vec}(P), \\ x^\intercal \tilde{Q}(K)x &= (x^\intercal \otimes x^\intercal)\mathrm{vec}(\tilde{Q}(K)), \\ (u^{(0)} + Kx)^\intercal B^\intercal Px &= (x^\intercal \otimes (u^{(0)\intercal} B^\intercal))\mathrm{vec}(P) + (x^\intercal \otimes x^\intercal)(I_N \otimes (K^\intercal B^\intercal))\mathrm{vec}(P), \end{aligned} \tag{2.83}$$

where $\mathrm{vec}(\cdot)$ is the vector operator, $\otimes$ denotes the Kronecker product, and I_N is the $N \times N$ identity matrix. Furthermore, we define the following matrices

$$\begin{aligned} \delta_\gamma(n) &= (x^\intercal \otimes x^\intercal)|_{t_{n-1}}^{t_n} \in \mathbb{R}^{N^2} \\ \gamma(n) &= \int_{t_{n-1}}^{t_n} (x^\intercal \otimes x^\intercal) dt \in \mathbb{R}^{N^2} \\ \lambda(n) &= \int_{t_{n-1}}^{t_n} (x^\intercal \otimes (u^{(0)\intercal} B^\intercal)) dt \in \mathbb{R}^{N^2}. \end{aligned} \tag{2.84}$$

Considering (2.83), (2.82) can be rewritten as follows

$$[\delta_\gamma(n) - 2\lambda(n) - 2\gamma(n)(I_N \otimes (K^\intercal B^\intercal))]\mathrm{vec}(P) = -\gamma(n)\mathrm{vec}(\tilde{Q}(K)). \tag{2.85}$$

By evaluating (2.85) on an increasing series of time intervals $\{t_n\}_{n=1}^{N_L}$, where $N_L > 0$ is a sufficiently large integer, the following relation is obtained

$$\Phi_P(K)\mathrm{vec}(P) = -\Gamma\mathrm{vec}(\tilde{Q}(K)), \tag{2.86}$$

in which $\Phi_P(K)$ is defined as

$$\Phi_P(K) = \Delta_\Gamma - 2\Lambda - 2\Gamma(I_N \otimes (K^\intercal B^\intercal)), \tag{2.87}$$

where

$$\begin{aligned} \Delta_\Gamma &= [\delta_\gamma(1)^\intercal \cdots \delta_\gamma(N_L)^\intercal]^\intercal \in \mathbb{R}^{N_L \times N^2} \\ \Lambda &= [\lambda(1)^\intercal \cdots \lambda(N_L)^\intercal]^\intercal \in \mathbb{R}^{N_L \times N^2} \\ \Gamma &= [\gamma(1)^\intercal \cdots \gamma(N_L)^\intercal]^\intercal \in \mathbb{R}^{N_L \times N^2}. \end{aligned} \tag{2.88}$$

In order to solve for the unique P in (2.86), $\Phi_P(K)$ has to be a full-rank matrix, i.e., $\mathrm{rank}(\Phi_P(K)) = N^2$. To this end, the exploration noise $e_L(t)$ plays a crucial role since its right choice will affect the rank condition.

Equation (2.86) finds P once the collected data and matrix K are given. One could solve (2.77) with a similar approach by using the same collected data. Given the symmetry of A, it can be easily verified that the time-derivative of $x^\intercal Sx$ along the trajectories of (2.80) is

$$\frac{d}{dt}(x^\intercal Sx) = -x^\intercal X_0 x + 2(u^{(0)}B^\intercal + x^\intercal BK)Sx, \tag{2.89}$$

where (2.77) makes the derivative independent of the knowledge of A. By integrating both members in (2.89) and using the Kronecker product properties, it holds that

$$[\delta_\gamma(n) - 2\lambda(n) - 2\gamma(n)(I_N \otimes (BK))]\mathrm{vec}(S) = -\gamma(n)\mathrm{vec}(X_0), \tag{2.90}$$

where the terms in the left member are defined as in (2.84). Note that (2.90) shares the same structure of (2.85), except for the term $(I_N \otimes (BK))$ used in the place of $(I_N \otimes (K^\intercal B^\intercal))$. Therefore, the following relationship holds true

$$\Phi_S(K)\mathrm{vec}(S) = -\Gamma\mathrm{vec}(X_0), \tag{2.91}$$

where $\Phi_S(K)$ is defined as

$$\Phi_S(K) = \Delta_\Gamma - 2\Lambda - 2\Gamma(I_N \otimes (BK)), \tag{2.92}$$

with the same definition of Δ_Γ, Λ, and Γ as in (2.88). As before, the rank condition $\mathrm{rank}(\Phi_S(K)) = N^2$ needs to be verified. Equations (2.86) and (2.91) provide a data-driven solution for the two Lyapunov equations (2.76) and (2.77), respectively.

Note that the proposed method does not require a data collecting phase for each Lyapunov equation to be solved. In fact, as in Algorithm 2.2, the same collected data is used to find the solution of (2.76) or (2.77), i.e., matrices P and S, respectively, according to a fixed matrix K defined a priori. The only requirement is the knowledge of system matrix B.

2.6.4 Proposed Algorithm

Several iterative approaches have been presented to optimize the problem (2.74) [70, 71]. The algorithm proposed herein integrates the data-based solution of the two Lyapunov equations with existing optimization approaches. In particular, the standard gradient descent algorithm is used. Starting from an initial stable controller $K_0 \in L$, a sequence $\{K^{(k)} \in L\}$ of controllers is obtained. For each pair of consecutive solutions $K^{(k)}$ and $K^{(k+1)}$, it is verified that $J^{(k+1)} \leq J^{(k)}$ where $J^{(j)}$ is the cost associated with $K^{(j)}$, $\forall j \in \mathbb{N}$. The sequence is generated by the following

$$K^{(k+1)} = K^{(k)} - \alpha^{(k)} \nabla J^{(k)}(K^{(k)}), \tag{2.93}$$

where $\alpha^{(k)}$ and $\nabla J^{(k)}(K^{(k)})$ represent the step size and the gain update direction during the kth iteration, respectively. The step size is determined according to the standard backtracking line search method, in which $\alpha^{(k)}$ is decreased until the closed-loop system is stable and the objective function decreases. To determine the descent direction $\nabla J^{(k)}(K^{(k)})$, note that if $P^{(k)}$ is the solution of (2.76) when $K = K^{(k)}$, then $l = 0$. This implies that $J^{(k)} = \mathcal{H}^{(k)}$, thus $\partial J^{(k)}/\partial K^{(k)} = \partial \mathcal{H}^{(k)}/\partial K^{(k)}$, where $\mathcal{H}^{(k)}$ is the Hamiltonian at the kth iteration. Therefore, (2.79) gives the update direction, that is

$$\nabla J^{(k)}(K^{(k)}) = 2(RK^{(k)}S^{(k)} - B^{\intercal}P^{(k)}S^{(k)}) \circ \mathcal{A}_C, \tag{2.94}$$

where $S^{(k)}$ is the solutions of (2.77) when $K = K^{(k)}$. In summary, given $K^{(k)}$, the update direction is obtained by (2.94) once the two Lyapunov equations are solved. Algorithm 2.3 implements the data-based Lyapunov equation solution with the search algorithm. Note that parameter $\beta < 1$ is employed in the backtracking line search to decrease the step size $\alpha^{(k)}$ during the kth iteration.

If matrix B is also unknown, a preliminary Value Iteration algorithm [73] can be employed to deal with fully unknown systems. After a preliminary data collection phase, such approach provides a fully-connected feedback matrix K_f, and the solution of the associated Lyapunov equation P_f. No assumptions are made on the stability of the system during the training phase. Once matrices K_f, P_f, and R are known, one can find B as $B = (RK_fP_f^{-1})^{\intercal}$. Note that the obtained matrix K_f can be used to define the initial stabilizing controller $\hat{u}^{(0)}(x)$ in Algorithm 1. Finally, K_f can find an initial $K^{(0)} \in L$, e.g., $K^{(0)} = K_f \circ \mathcal{A}_C$, once the closed loop stability is checked through the positive definiteness of the corresponding matrix $P^{(0)}$.

2.6.5 Application Example

The proposed algorithm can be used to find optimal structured feedback controllers for any system in form of (2.68) with $A = A^{\intercal}$. In this section, the algorithm performance are evaluated on a numerical example. The case study consists of a multi tank system composed by the interconnection of N identical tanks, as shown in Fig. 2.10. For each tank, let V_i be the volume of the incompressible liquid, q_i^{in} the amount of inflow liquid, and q_i^{out} the amount of outflow liquid. Thus, the following holds

Algorithm 2.3 Data-based algorithm to solve the structured optimal control problem

Inputs: Initial stabilizing and exploring control policy $u^{(0)} = \hat{u}^{(0)}(x) + e_L(t)$; System matrix B; Initial auto-correlation X_0; Desired feedback structure $\mathcal{A}_C$; Performance index parameters Q, R; Any initial stable controller $K^{(0)} \in L$; Sequence of learning intervals $\{t_n\}_{n=1}^{N_L}$; Stopping threshold ε; Parameter $\beta < 1$.

Outputs: Optimal feedback matrix $K^* \in L$;

1. **Data Collecting Phase**
   Apply $u^{(0)}$ at system (2.68) and collect matrices Δ_Γ, Λ, Γ.
2. **Initialization**
   Determine matrices $\Phi_P(K^{(0)})$ and $\Phi_S(K^{(0)})$; Obtain $P^{(0)}$ and $S^{(0)}$ from (2.86) and (2.91); Set $k = 0$.
3. **Cost and Direction Computation**
   Evaluate cost $J^{(k)} = \mathrm{Tr}(P^{(k)}X_0)$ and direction $\nabla J^{(k)}(K^{(k)})$ as in (2.94); Set $\alpha^{(k)} = 1$.
4. **Backtracking Line Search**
   a. Evaluate $K^{(k+1)}$ as in (2.93) and determine matrices $\Phi_P(K^{(k+1)})$ and $\Phi_S(K^{(k+1)})$.
   b. if $\Phi_P(K^{(k+1)})$ and $\Phi_S(K^{(k+1)})$ are not full-rank, set $\alpha^{(k)} = \beta\alpha^{(k)}$ and go to Step 4a; Otherwise, go to Step 4c.
   c. Obtain $P^{(k+1)}$ and $S^{(k+1)}$ from (2.86) and (2.91); Evaluate new cost $J^{(k+1)} = tr(P^{(k+1)}X_0)$.
   d. if $P^{(k+1)} > 0$ and $J^{(k+1)} \leq J^{(k)}$, go to Step 5; Otherwise, set $\alpha^{(k)} = \beta\alpha^{(k)}$ and go to Step 4a.
5. **Stopping Criterion**
   if $|J^{(k+1)} - J^{(k)}| \leq \varepsilon$, stop and return $K^* = K^{(k+1)}$; Otherwise, set $k = k + 1$ and go to Step 3.

$$q_i^{in} - q_i^{out} = \dot{V}_i = a_t \dot{h}_i, \quad i = 1, ..., N \tag{2.95}$$

where a_t is the base area of each tank, and h_i is the level of liquid. The inflows q_i^{in} represent external inputs, while the outflows q_i^{out} can be expressed via the Torricelli law

$$q_i^{out} = k_{ii}\sqrt{2g|h_i|} + \sum_{j \in N_i} k_{ij}\mathrm{sign}(h_i - h_j)\sqrt{2g|h_i - h_j|}, \tag{2.96}$$

where N_i is the set of indices j such that tank i and tank j are connected, while $\mathrm{sign}(\cdot)$ denotes the sign function. Coefficients k_{ii} and k_{ij} are constant values that depend on the parameters of corresponding pipes (e.g., orifice area, discharge coefficient). Clearly, $k_{ij} = k_{ji}, \forall i$ and $\forall j \in N_i$.

The control objective is to maintain each tank level to a specified value $h_i^* > 0$. By solving (2.95) for each i at the steady state, the corresponding target inflows $q_i^{in^*}, \forall i = 1, ..., N$, are found. It can be shown [74] that linearizing system (2.95) around the target equilibrium point gives

$$\dot{x}_i = -\underbrace{(a_i + \sum_{j \in N_i} a_{ij})}_{a_{ii}} x_i + \sum_{j \in N_i} a_{ij} x_j + u_i, \tag{2.97}$$

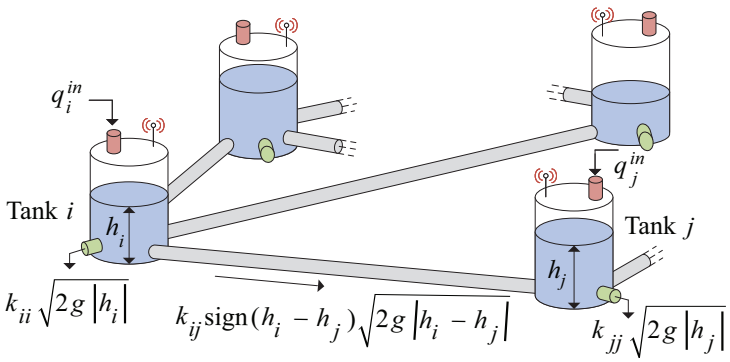

Figure 2.10: Interconnected tanks system.

where $x_i = h_i - h_i^*$ represents the fluid level errors, with $a_i = \frac{k_{ii}\sqrt{2g}}{2a_t\sqrt{|h_i^*|}}$ and $a_{ij} = \frac{k_{ij}\text{sign}^2(h_i^* - h_j^*)\sqrt{2g}}{2a_t\sqrt{|h_i^* - h_j^*|}}$, and $u_i = \frac{q_i^{in} - q_i^{in^*}}{a_t}$ are the control inputs. Clearly, the overall system is in form (2.68) with $A = A^\intercal$ and $B = I_N$. As in Fig. 2.10, each tank is equipped with a communication module that implements distributed control policies. Due to spatial constraints, a structured optimal feedback is preferred (e.g., only tanks in physical proximity can communicate).

Consider a system with $N = 6$ interconnected tanks, described by the following state matrix

$$A = \begin{bmatrix} -6.4 & 1.1 & 0 & 2.1 & 1.5 & 1.3 \\ 1.1 & -7.2 & 1.7 & 3 & 0 & 1.2 \\ 0 & 1.7 & -4.4 & 0 & 2.5 & 0 \\ 2.1 & 3 & 0 & -5.7 & 0 & 0 \\ 1.5 & 0 & 2.5 & 0 & -6.5 & 2.2 \\ 1.3 & 1.2 & 0 & 0 & 2.2 & -4.9 \end{bmatrix}. \tag{2.98}$$

Note that (2.98) is Hurwitz. The corresponding physical graph is reported in Fig. 2.11(a). Algorithm 2.3 finds the solution of the structured feedback with respect to four different structures, i.e., $\mathcal{A}_{C_1}$, $\mathcal{A}_{C_2}$, $\mathcal{A}_{C_3}$, and $\mathcal{A}_{C_4}$ depicted in Fig. 2.11(b), Fig. 2.11(c), Fig. 2.11(d), and Fig. 2.11(e), respectively. Structure $\mathcal{A}_{C_1}$ implements a fully decentralized controller, i.e., no communication is needed. $\mathcal{A}_{C_2}$ considers the tanks 1,2, and 6 not accessible for control purposes. In $\mathcal{A}_{C_3}$ only nearby tanks are allowed to communicate, i.e., tanks 2,3, and 5, are not in the physically vicinity of tanks 1,4, and 6. Finally, $\mathcal{A}_{C_4}$ consider a fault in the level sensor of tank 1, i.e., its control does not depend on its own state but depends on the level errors of nearby tanks 2,4, and 6. The performance matrices are selected as $Q = 10I_N$ and $R = I_N$. For simplicity, $X_0 = I_N$. Within the data collecting phase, a sequence of $N_L = 100$ time intervals of 0.01 s is used. The open-loop stability of A allows us to use $\hat{u}^{(0)} = 0$ as a stabilizing controller; Therefore, $u^{(0)} = e_L(t)$. The learning noise $e_L(t)$ is given by 6 filtered white noises. Parameters β and ε are set to 0.5 and 10^{-6}, respectively.

Once the data collecting phase is completed, the open-loop cost is found by solving (2.86) with $K = 0_{6\times 6}$, providing the value of $J^{(0)} = 20.34$. A preliminary value-iteration phase finds the optimal fully connected feedback matrix K_f, and the corresponding matrix P_f. The resulting fully-connected optimal cost is $J_f = \text{Tr}(P_f X_0) = 6.77$. Four different design based on Algorithm 2.3 are obtained, one for each communication graph in Fig. 2.11. The initial controller is obtained by truncating K_f according to the considered structure. Stability of initial

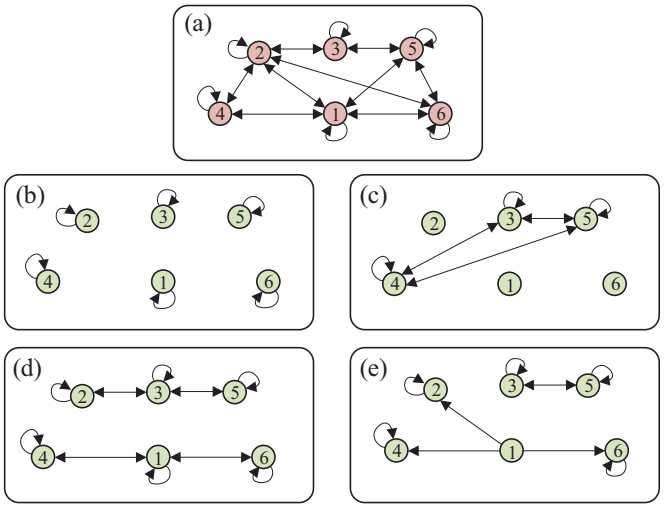

Figure 2.11: Considered graph structures: (a) Physical interconnection graph; (b) $\mathcal{A}_{C_1}$; (c) $\mathcal{A}_{C_2}$; (d) $\mathcal{A}_{C_3}$; and (e) $\mathcal{A}_{C_4}$.

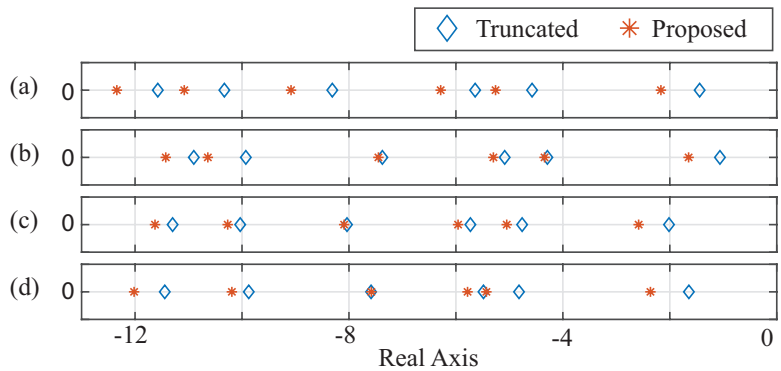

Figure 2.12: Closed-loop eigenvalues for each communication structure: (a) $\mathcal{A}_{C_1}$; (b) $\mathcal{A}_{C_2}$; (c) $\mathcal{A}_{C_3}$; and (d) $\mathcal{A}_{C_4}$.

Table 2.1: Performance comparison

Structure	J'	J	variation
$\mathcal{A}_{C_1}$	8.56	7.33	-6.78%
$\mathcal{A}_{C_2}$	9.32	8.49	-8.94%
$\mathcal{A}_{C_3}$	7.19	7.03	-2.22%
$\mathcal{A}_{C_4}$	7.62	7.25	-4.75%

controllers is verified by the positive definiteness of the corresponding P matrices found by solving (2.86). The optimal costs computed for each configuration are reported in Table 2.1, where column J refers to the feedback gain obtained via Algorithm 2.3, and column J' refers to the truncated optimal matrix K_f. In all configurations, the proposed approach outperforms the cost corresponding to truncated matrices. Finally, the eigenvalues of the closed-loop system obtained with both methods are shown in Fig. 2.12, for each communication graph. Note how the proposed method always leads to a faster dominant eigenvalue.

2.7 Publications

The results presented in Section 2.6 have been published by the author in [75].

Chapter 3

Energy Optimal Control of Dielectric Elastomer Actuators

A first application of the Adaptive Dynamic Programming (ADP) for the control of complex nonlinear systems is developed in this chapter, where the closed loop optimal control of mechatronic devices based on dielectric elastomer membranes is considered. The goal is to minimize the input electrical energy required to achieve a given position regulation task, i.e., an energy optimal position control scheme for such actuators is developed. The actuator's model is based on a free-energy framework, which provides a thermodynamically consistent characterization of the losses occurring during actuation. Due to the strongly nonlinear behavior of both system model and dissipation function, traditional techniques based on analytical solution of the Hamilton-Jacobi-Bellman (HJB) equation cannot be applied. Therefore, the ADP algorithm with off-policy learning in Chapter 2 is employed as a tool to solve offline the HJB equation related to the energy minimization problem. After discussing the theory, experimental results are presented to validate the effectiveness of the proposed approach.

3.1 Overview and Objectives

3.1.1 Dielectric Elastomer Actuators

Dielectric elastomers (DEs) are an attractive class of mechatronic transducers which has received a significant interest over the last two decades [76]. A DE membrane consists of an elastic polymeric film coated on both sides with compliant electrodes. The resulting structure is a flexible capacitor which can be used as an actuator, by converting an applied voltage into motion, as well as a sensor, since capacitance changes can be related to the membrane geometry. Other interesting DE features include high energy density (0.4 J/g), large deformation ranges ($> 100\%$), high compliance (Young's modulus between 0.1 and 10 MPa), and self-sensing capabilities which allow to implement closed-loop controllers without the need of displacement sensors [51, 77]. Grippers [78, 79], wave energy harvesters [80], pumps [81, 82], valves [83], prostheses [84], micro-positioning systems [85], and bio-inspired robots [86, 87] represent only some of the many DE-based devices presented in the recent literature.

Strong nonlinear behavior, sensitivity to environmental conditions, and high voltage requirements (order of kV) currently represent the major limitations for DE actuators (DEAs) in indus-

trial applications. The strong nonlinearities due to the physical behavior of a DE membrane motivates the need for developing advanced modeling and control techniques [88]. In fact, a number of research works dealing with dynamical modeling and analysis of the underlying physical phenomena have been published over recent years [89–91]. In addition, most of the recent control-related literature focuses on the position control of DEAs with several techniques, such as sliding-mode control [52], feed-forward [92], adaptive gain-scheduling [93], Proportional-Integral-Derivative (PID) based controllers [94], cerebellar-inspired controllers [95, 96], and robust linear control based on linear matrix inequalities [50]. Other recent approaches include compensation methods used to remove the nonlinearities of the DEA, such as inverse viscoelastic hysteresis compensations [97, 98], and PID controllers combined with an identified inverse model of the DEA [99].

3.1.2 Objectives and Procedure

The goal of this chapter is to develop a novel minimum energy control strategy for DEAs. Note that despite several types of controllers have been presented for DEA systems, the development of energy efficient control approaches has not received attention from the research community so far. In [100], the authors proposed an energetically-consistent DEA model based on the port-Hamiltonian framework. Since such model is structurally passive, it allows to consistently quantify the amount of energy dissipated in each part of the system, i.e., due to electrical (Joule effect) and mechanical (viscoelasticity) losses. Therefore, in order to enhance the energy-efficiency capabilities of DEA devices during a positioning task, an optimal feedback control strategy can be employed to minimize these losses, by formulating the energy-minimization objective as an optimal control problem. However, due to the strongly nonlinear behavior of the DEAs, the solution of the HJB equation is intractable. Thus, an ADP approach is employed.

The proposed approach can be summarized as follows. First, an energy consistent model of the system is discussed. Such model effectively describes the losses that occur in the actuator. Then, an identification procedure characterizes and validates such model. The energy losses function is then used as utility function for an optimal control problem, which is solved by means of an ADP algorithm with off-policy learning. The ADP approach is used as a tool to solve offline the HJB equation, and derive energy-efficient control laws for a given set of target displacement values. Finally, experimental tests show the effectiveness of the proposed method.

Main features and contributions of this work are summarized as follows.

- A general and accurate description of both DEA model and energy losses, by considering a finite value of the elastomer leakage resistance, is provided. In this way, the presented approach is valid for the more realistic class of DEAs in which the energy losses depend also on the leakage resistance;

- The first experimental validation of a DEA energy consistent model is presented. It is shown how the developed passive model properly predicts the coupled electro-mechanical response of the DEA and, thus, is well-suited for the design of energy efficient controllers;

- The optimal energy controller is experimentally validated, highlighting substantial improvements in terms of energy saving when compared with other traditional position control techniques, such as PID or feed-forward controllers;

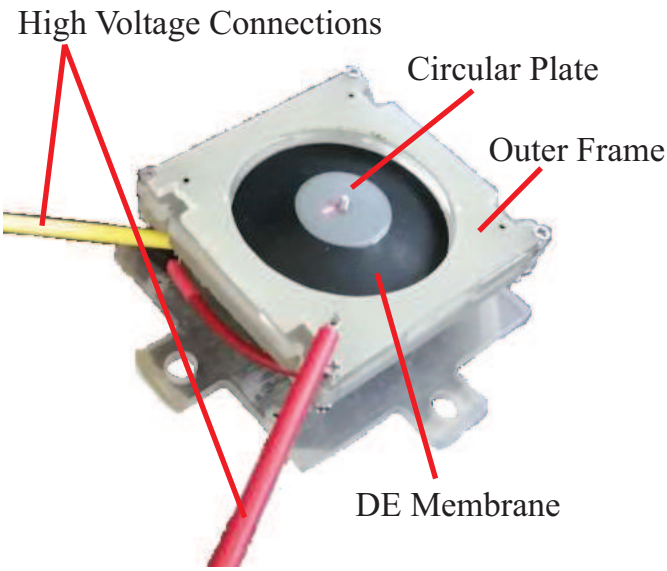

Figure 3.1: Picture of the DE actuator considered in this work.

- A robustness analysis aimed at evaluating the sensitivity of the closed loop performances with respect to changes in constitutive DE parameters is developed.

3.1.3 Chapter's Outline

This chapter is organized as follows. Section 3.2 provides the DEA physical model, by discussing the model development and analyzing the passivity of such model. The results of the experimental identification procedure are reported in Section 3.3. The energy optimal control problem is formulated and solved via ADP in Section 3.4. Section 3.5 presents the experimental results along with the robustness analysis for varying parameters. Finally, concluding remarks are reported in Section 3.6.

3.2 Dynamic Model of the DEA

The DEA considered in this work is shown in Fig. 3.1. A schematic representation of the overall system is given in Fig. 3.2(a) and (b) for both unactuated and actuated conditions, respectively. It consists of ring-shaped, pre-stretched silicone membranes placed in between an outer frame and an inner circular plate, both made of rigid plastic. The membrane actuator is made of several silicone layers, mechanically connected in parallel. Each polymeric layer is sandwiched between two carbon-based compliant electrodes, connected to two high-voltage connectors.

When a voltage is applied to the electrodes, the resulting deformable capacitor is subject to a pressure known as Maxwell Stress that squeezes the material along the thickness direction [101]. The consequent area expansion results in an actuation. The amount of stroke depends on the type of mechanical biasing system connected to the membrane. As shown in Fig. 3.2, the biasing system considered in this work is made of a linear spring and a nonlinear bi-stable buckled beam coupled with the membrane through a rigid spacer. This solution permits to

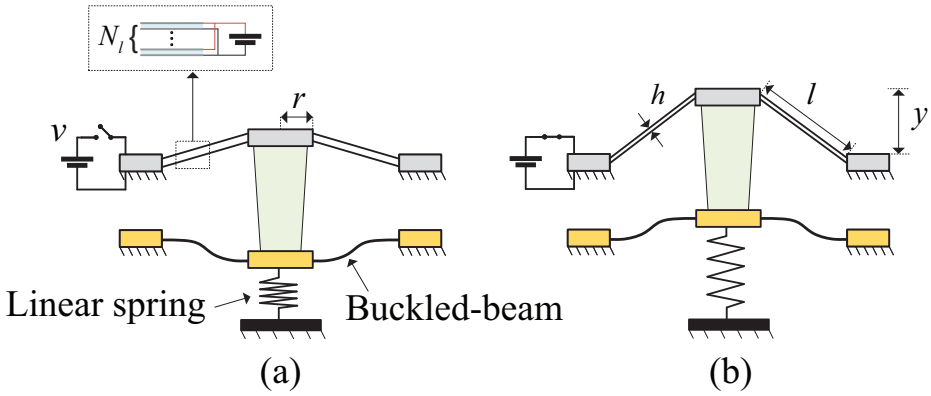

Figure 3.2: Actuating configurations: (a) Unactuated; (b) Actuated.

significantly magnify the stroke compared to simple linear springs, at the expense of increasing the actuator nonlinearity [94].

In this section, a dynamic model of the considered actuator is first provided. The constitutive differential equations are based on the work previously presented in [50, 94, 100]. The energetic consistency of such model is analyzed, and the energy minimization problem is subsequently expressed in terms of an optimal control problem.

3.2.1 Model Development

A state-space representation of the actuator model is developed in this section. The control input of the actuator is the applied voltage v, while the states are chosen as the out-of-plane displacement y, the circular plate momentum p in the out-of-plane direction, the M internal states of the material viscoelastic dynamics ε_{kj}, $j = 1, ..., M$, and the electric charge stored on the electrodes q. The complete state vector is defined as

$$z = [y \quad p \quad \varepsilon_{k1} \quad \cdots \quad \varepsilon_{kM} \quad q]^{\intercal}. \tag{3.1}$$

Let m be the mass of the circular plate. By definition it results that

$$\dot{y} = \frac{p}{m}. \tag{3.2}$$

The time derivative of the momentum p is given by the summation of the applied forces, that is

$$\dot{p} = -mg - F_{LS}(y) - F_{BB}(y) - N_l F_{DE}(z), \tag{3.3}$$

where g is the gravitational acceleration, N_l is the number of DE layers, while $F_{LS}(z)$, $F_{BB}(y)$ and $F_{DE}(y)$ represent the forces produced by the linear spring, the buckled-beam, and the single DE layer, respectively.

The forces provided by the linear spring and buckled-beam are given as follows

$$\begin{aligned} F_{LS}(y) &= -k_{bl}(y - y_{0l}), \\ F_{BB}(y) &= k_{bn1}(y - y_{0n}) - k_{bn7}(y - y_{0n})^7. \end{aligned} \tag{3.4}$$

Coefficients k_{bl} and y_{0l} represent the stiffness and the initial displacement of the linear spring, respectively. The buckled-beam, modeled as a bi-stable nonlinear spring, is defined by stiffness coefficients k_{bn1} and k_{bn7} and initial displacement y_{0n}.

The DE layer force $F_{DE}(z)$ can be expressed as a function of the DE Helmholtz free-energy $\Psi(z)$, i.e.,

$$F_{DE}(z) = \frac{\partial \Psi(z)}{\partial \varepsilon_1(y)} \frac{d\varepsilon_1(y)}{dy} + \sum_{j=1}^{M} \frac{\partial \Psi(z)}{\partial \varepsilon_{kj}} \frac{d\varepsilon_1(y)}{dy} + \frac{\eta_{v0} V_{DE}}{\varepsilon_1(y)+1} \left(\frac{d\varepsilon_1(y)}{dy} \right)^2 \frac{p}{m}, \tag{3.5}$$

where

$$\varepsilon_1(y) = \sqrt{1 + \left(\frac{y}{l_0} \right)^2} - 1 \tag{3.6}$$

represents the radial strain of the membrane with respect to the undeformed radial length l_0. The volume of each layer is defined as $V_{DE} = \pi (2r + l_0) l_0 h_0$, with r and h_0 representing the radius of the inner circular plate and the thickness of the undeformed and unactuated membrane (flat configuration). Due to incompressibility of the elastomer, V_{DE} remains constant during actuation. The Helmholtz free-energy for a single DE membrane is defined as follows

$$\begin{aligned} \Psi(z) = V_{DE} \sum_{i=1}^{N_O} \left\{ \frac{\beta_i}{\alpha_i} \left[(1+\varepsilon_1(y))^{\alpha_i} - 1 \right] + \frac{\gamma_i}{\alpha_i} \left[(1+\varepsilon_1(y))^{-\alpha_i} - 1 \right] \right\} + \frac{1}{2C(y)} q^2 \\ + V_{DE} \sum_{j=1}^{M} k_{vj} \left[\varepsilon_{kj} - a_j(y, \varepsilon_{kj}) \log \left(\frac{\varepsilon_1(y)+1}{\varepsilon_1(y) - \varepsilon_{kj} + 1} \right) \right] \end{aligned} \tag{3.7}$$

with $a_j(y, \varepsilon_{kj}) = \varepsilon_1(y) - \varepsilon_{kj} + 1$, $j = 1, \ldots, M$. Function $\Psi(z)$ is always non-negative for every admissible operating state of the actuator, and vanishes to zero for $z = 0$. The first term in (3.7) represents the hyperelastic energy contribution due to material deformation, described via a modified Ogden model of order N_O with coefficients α_i, β_i, and γ_i, $i = 1, \ldots, N_O$. The second term in (3.7) describes the electrostatic energy stored in the flexible capacitor. The capacitance $C(y)$ can be expressed as a function of displacement via the well-known parallel-plate capacitor formula, which results into

$$C(y) = \epsilon_0 \epsilon_r \frac{V_{DE}}{h_0^2} \left(1 + \varepsilon_1(y) \right)^2, \tag{3.8}$$

where ϵ_0 and ϵ_r represent vacuum and material relative permittivity, respectively. The third and final term represents an additional energy storage contribution due to viscoelastic relaxation. In particular, the internal viscoelasticity of the material is modeled as a parallel connection of a damper with damping coefficient η_{v0} and M serial spring-damper systems, each one of them characterized by spring stiffness k_{vj} and damping η_{vj}. In this way, each state ε_{kj} and the corresponding $j-th$ term in the last summation appearing in (3.7) can be interpreted as the strain and the energy stored in the viscoelastic spring k_{vj}, respectively (see [100] for details). The viscoelastic internal states are related to the parameters k_{vj} and η_{vj} as in the following

$$\dot{\varepsilon}_{kj} = -\frac{k_{vj}}{\eta_{vj}} \varepsilon_{kj} + \frac{d\varepsilon_1(y)}{dy} \frac{p}{m}, \qquad j = 1, \ldots, M. \tag{3.9}$$

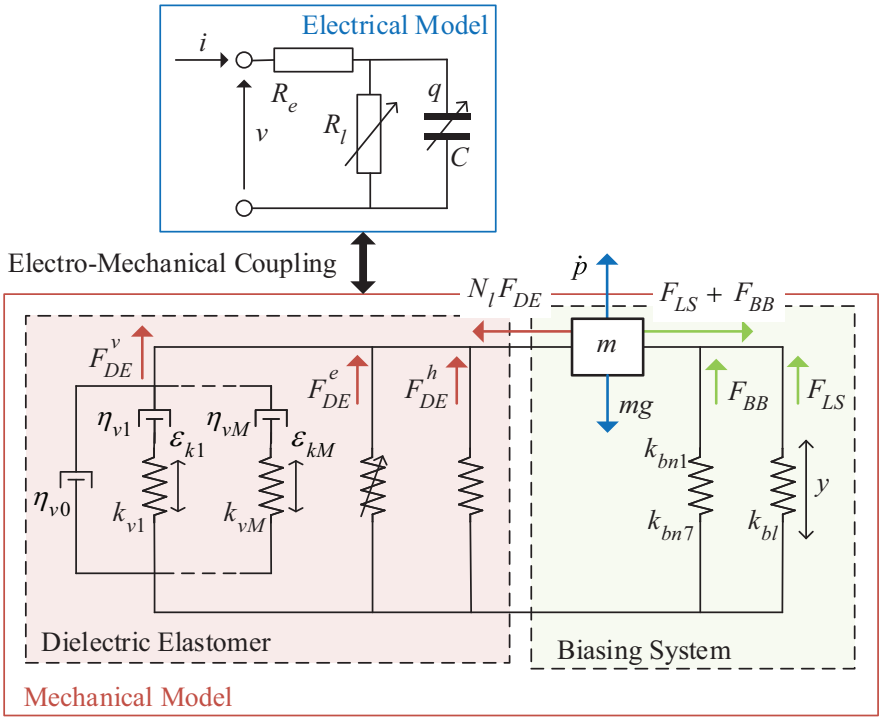

Figure 3.3: Equivalent electro-mechanical scheme representing the overall actuator model.

By plugging (3.6) and (3.7) into (3.5), the complete expression of the DE force is derived

$$F_{DE}(z) = \underbrace{-V_{DE}\frac{y}{l_0^2+y^2}\left(\sum_{j=1}^{M} k_{vj}\varepsilon_{kj} + \eta_{v0}\frac{y}{l_0\sqrt{l_0^2+y^2}}\frac{p}{m}\right)}_{F_{DE}^v(z)} \underbrace{+\frac{y}{l_0^2+y^2}\frac{1}{C(y)}q^2}_{F_{DE}^e(z)}$$
$$\underbrace{-V_{DE}\frac{y}{l_0^2+y^2}\sum_{i=1}^{N_O}\left[\beta_i\left(1+\frac{y^2}{l_0^2}\right)^{\frac{\alpha_i}{2}} - \gamma_i\left(1+\frac{y^2}{l_0^2}\right)^{-\frac{\alpha_i}{2}}\right]}_{F_{DE}^h(z)} \tag{3.10}$$

with $F_{DE}^v(z)$, $F_{DE}^e(z)$, and $F_{DE}^h(z)$ representing the contributions due to the viscoelasticity of the material, the electro-mechanical coupling, and the hyperelasticity of the material, respectively. For the ease of presentation, a conceptual sketch containing an equivalent electro-mechanical model representing the overall actuator system is depicted in Fig. 3.3. It is remarked how the depicted diagram represents only a conceptual equivalence, since most of the actuator nonlinearities and coordinate transformation are not included.

Finally, the electrical dynamics is modeled as in [100], where an equivalent nonlinear RC circuit is constructed by connecting a capacitive element (representing the DE) to both a serial and a parallel resistor, as in Fig. 3.3. A model for the equivalent RC circuit is given in the

following expression

$$\begin{aligned} \dot{q} &= -\left(\frac{1}{R_e C(y)} + \frac{1}{R_l(y)C(y)}\right) q + \frac{1}{R_e} v, \\ i &= -\frac{N_l}{R_e C(y)} q + \frac{N_l}{R_e} v. \end{aligned} \tag{3.11}$$

In (3.11), R_e is the equivalent electrode resistance of each membrane, and $R_l(y)$ represents the leakage resistance of the variable capacitor, i.e.,

$$R_l(y) = \rho \frac{h_0^2}{V_{DE}} \left(1 + \varepsilon_1(y)\right)^{-2}, \tag{3.12}$$

with ρ describing the resistivity of the dielectric. The total current flowing in the stacked membranes is represented by i.

By collecting equations (3.2), (3.3), (3.9), and (3.11), the nonlinear state-space model of the DEA can be defined as follows

$$\dot{z} = f(z) + Bv, \tag{3.13}$$

where $B = [0 \quad 0 \quad \dots \quad 1/R_e]^\intercal$, and $f(z)$ is expressed as

$$f(z) = \begin{bmatrix} \frac{p}{m} \\ -mg + k_{bl}(y - y_{0l}) - k_{bn1}(y - y_{0n}) + k_{bn7}(y - y_{0n})^7 - N_l F_{DE}(z) \\ -\frac{k_{v1}}{\eta_{v1}} \varepsilon_{k1} + \frac{d\varepsilon_1(y)}{dy} \frac{p}{m} \\ \vdots \\ -\frac{k_{vM}}{\eta_{vM}} \varepsilon_{kM} + \frac{d\varepsilon_1(y)}{dy} \frac{p}{m} \\ -\left(\frac{1}{R_e C(y)} + \frac{1}{R_l(y)C(y)}\right) q \end{bmatrix}, \tag{3.14}$$

where $F_{DE}(z)$ is as in (3.10), $C(y)$ as in (3.8), and $R_l(y)$ as in (3.12).

3.2.2 Passivity Analysis

Let be $\Psi_b(y, p)$ the total energy associated with the biasing system, given by the sum of the potential and kinetic energies of the mass m, linear spring, and buckled-beam, i.e.,

$$\Psi_b(y, p) = mgy + \frac{1}{2m} p^2 + \frac{1}{2} k_{bl}(y - y_{0l})^2 - \frac{1}{2} k_{bn1}(y - y_{0n})^2 + \frac{1}{8} k_{bn7}(y - y_{0n})^8. \tag{3.15}$$

The total energy of the actuator system $\Psi_a(z)$ can then be computed as the sum between the total Helmholtz free-energy related to all the stacked membranes and the mechanical energy of the biasing system, as follows

$$\Psi_a(z) = N_l \Psi(z) + \Psi_b(y, p). \tag{3.16}$$

The derivative with respect to the time of $\Psi_a(z)$ is

$$\dot{\Psi}_a(z) = \frac{\partial \Psi_a(z)}{\partial y} \dot{y} + \frac{\partial \Psi_a(z)}{\partial p} \dot{p} + \frac{\partial \Psi_a(z)}{\partial q} \dot{q} + \sum_{j=1}^{M} \frac{\partial \Psi_a(z)}{\partial \varepsilon_{kj}} \dot{\varepsilon}_{kj}. \tag{3.17}$$

By developing the partial derivatives and replacing the time derivatives of the state as in (3.13), it follows that

$$\dot{\Psi}_a(z) = vi - s(z, u), \tag{3.18}$$

where

$$\begin{aligned} s(z, v) = & \frac{N_l}{R_l(y)} \left(\frac{q}{C(y)} \right)^2 + \frac{1}{N_l} R_e i^2 + N_l \frac{\eta_{v0} V_{DE}}{\varepsilon_1(y) + 1} \left(\frac{d\varepsilon_1(y)}{dy} \frac{p}{m} \right)^2 \\ & + N_l \sum_{j=1}^{M} \frac{k_{vj}^2 V_{DE}}{\eta_{vj}} \log \left(\frac{\varepsilon_1(y) + 1}{\varepsilon_1(y) - \varepsilon_{kj} + 1} \right) \varepsilon_{kj}. \end{aligned} \tag{3.19}$$

Note the explicit dependency of $s(z, v)$ on input voltage v, due to the dependency of the current i on v. As proven in [100], each term appearing in the summation in (3.19) is always non-negative in the useful operating range of the actuator, and vanishes to zero if $\varepsilon_{kj} = 0$. Therefore, it can be concluded that function $s(z, v)$ is always non-negative and vanishes in the equilibrium for $v = 0$. This result implies that

$$\dot{\Psi}_a(z) \leq vi \tag{3.20}$$

holds true for every admissible trajectory of (3.13). Hence, system (3.13) is passive with respect to storage function $\Psi_a(z)$, supply rate vi, and dissipation function $s(z, v)$ [102]. Clearly, storage function $\Psi_a(z)$ and supply rate vi can be interpreted as the total electro-mechanical energy stored in the system and the input electric power supplied to the actuator, respectively. Consequently, the dissipation function in (3.19) can be naturally interpreted as the total energy loss due to the dissipative nature of the DE. In particular, the first two terms on the right-hand side of (3.19) describe the Ohmic losses in the electrode and leakage resistances, respectively (resistive elements in Fig. 3.3). The third and fourth terms, instead, describe the mechanical losses due to the viscoelasticity of the material (damping elements in Fig. 3.3). By integrating both sides of (3.18) over an arbitrary time interval $[t_0, t_1]$ and rearranging the terms it results that

$$\Psi_a(z(t_1)) - \Psi_a(z(t_0)) + \int_{t_0}^{t_1} s(z, v) dt = \int_{t_0}^{t_1} vi dt. \tag{3.21}$$

This last relationship permits to express the energy supplied over any time interval as the sum between the change in energy between initial and final state and the (non-negative) energy loss due to internal dissipation. Since $\Psi_a(z)$ is a state function, the change in energy is uniquely determined once a positioning task is given in terms of initial $(z(t_0))$ and final $(z(t_1))$ states. Conversely, energy supply and energy dissipation explicitly depend on the trajectory taken by the system between t_0 and t_1. Therefore, the minimization of the input energy required to drive the actuator between two given equilibrium states can be equivalently stated as the minimization of the energy dissipated during the process.

3.3 Parameter Identification

The sketch of the experimental bench used to test and validate the developed model is shown in Fig. 3.4. Data acquisition, signal processing, and control routines are implemented on a STM32 Nucleo-144 board, operating at a sampling rate of 1 ms. Displacement values y are acquired

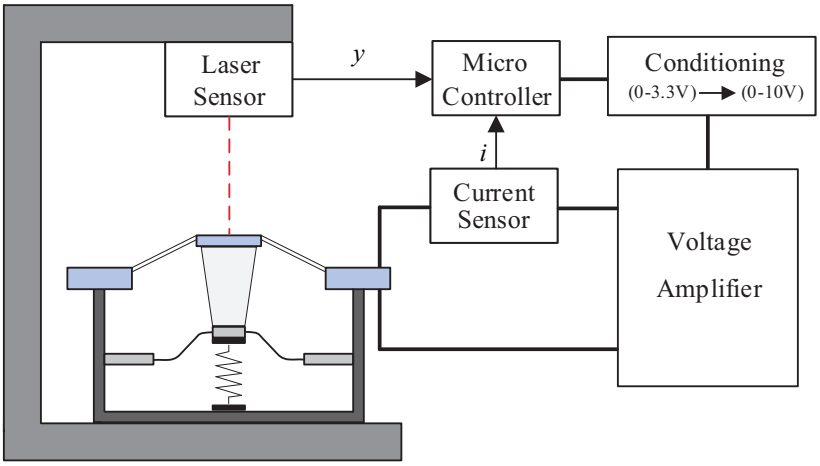

Figure 3.4: Sketch of the experimental test bench.

Table 3.1: Known DEA Parameters

Symbol	Unit	Value	Symbol	Unit	Value
N_l		4	m	g	5
r	mm	10	k_{bl}	N/mm	1.2
l_0	mm	12.5	k_{bn1}	N/mm	1.36
h_0	μm	51	k_{bn7}	N/mm	$4 \cdot 10^{15}$
g	m/s^2	9.81	y_{0l}	mm	9.69
ϵ_0	F/m	$8.85 \cdot 10^{-12}$	y_{0n}	mm	7.58

through a Keyence LK-G157 laser sensor (0.15 μm of resolution), while current values i are acquired through a current sensor (range of $\pm$ 2 mA). The microcontroller provides the input voltage v, translated from (0-3.3) V to (0-10) V by an appropriate conditioning circuit that drives a TREK 610E voltage amplifier connected to the DEA. Maximum current and voltage limits are set to 3 kV and 2 mA, respectively, compatibly with DE breakdown voltage and hardware limitations.

Some DEA parameters are known in advance, as reported in Table 3.1. Moreover, the actuator displacement y can range from $y_{min} = 6.37$ mm to $y_{max} = 7.89$ mm. An experimental identification is required for the remaining unknown parameters, i.e., order of viscoelastic dynamics M and its parameters η_{v0}, η_{vj}, and k_{vj}, with $j = 1, \ldots, M$; order of Ogden model N_O and its parameters α_i, β_i, and γ_i, with $i = 1, \ldots, N_O$; material relative permittivity ϵ_r; series resistance R_e; dielectric resistivity ρ. Based on earlier works, as well as to limit the computational requirements of the identification algorithm, some parameters are set *a priori*, namely order of the internal viscoelastic model $M = 1$, order of the Ogden model $N_O = 3$, and Ogden coefficients $\alpha_1 = 2$, $\alpha_2 = 4$, and $\alpha_3 = 6$. The value $M = 1$ is motivated by the fact that higher values of this parameter do not lead to substantial improvements in the FIT values, and have the only effect of increasing the model complexity. This fact is observed by means of numerical studies which are not shown here for conciseness. However, the generalized approach here presented is still valid when higher values of M are considered. Higher values of M could be

Table 3.2: Identified DEA Parameters

Symbol	Unit	Value	Symbol	Unit	Value
k_{v1}	kPa	315.5	γ_1	MPa	53.69
η_{v1}	kPa·s	82.31	γ_2	MPa	-57.32
η_{v0}	kPa·s	74.53	γ_3	MPa	19.12
β_1	MPa	29.67	ϵ_r	-	2.32
β_2	MPa	-17.28	R_e	MΩ	1.14
β_3	MPa	3.19	ρ	Ω· m	$6.5 \cdot 10^{10}$

preferred, for instance, in case of DEAs operating in a broader frequency range, see, e.g., [103].

The identification procedure is implemented through a MATLAB routine based on the Nelder-Mead simplex method. Given an input voltage waveform, once the corresponding displacement and current values are recorded, the algorithm finds the best parameters set that maximizes a weighted sum of the displacement FIT, the current FIT, and the input energy FIT. The experimental input energies are evaluated by integrating the product between measured voltages and currents. Herein the FIT is defined as follows

$$FIT = 100\left(1 - \frac{\|x_{meas} - x_{model}\|}{\|x_{meas} - \text{mean}(x_{meas})\|}\right), \tag{3.22}$$

where x_{meas} and x_{model} represent a generic measured and model-predicted quantity, e.g., displacement y or current i, and $||x||$ represents the Euclidean norm of vector x.

Three different experimental tests, shown in Fig. 3.5(a), Fig. 3.5(e), and Fig. 3.5(i), are used to calibrate and validate the model. The identified parameters are reported in Table 3.2. Figures 3.5(b), 3.5(c) and 3.5(d) show the comparison between the experimental and predicted displacement, current, and input energy, respectively, when the signal is chosen as a sum of sine waves having different amplitudes, frequencies, and phases. Figures 3.5(f), 3.5(g) and 3.5(h) show the same comparison according to an amplitude-modulated pseudo random binary signal (APRBS) chosen as input. Finally, figures 3.5(l), 3.5(m), and 3.5(n) report the comparison according to an increasing sequence of steps. Model validation's has been carried out both with the whole third signal and with portions of the APRBS and sum of sines not used during the training phase. The identified model parameters, with a viscoelastic model of order $M = 1$, permit to reproduce the experimental results with satisfactory accuracy. The FIT values on both training and validation data are reported in Table 3.3. Note that the current FIT shows lower values if compared with displacement and energy FITs. This is a consequence of the adopted FIT measure, since it penalizes signals which are often time close to zero (this is, indeed, more common for the current rather than for the other two quantities). Moreover, the current measurement is affected by a higher signal-to-noise ration with respect to the other two measurements. This fact unavoidably affects the corresponding value of the current FIT. Nevertheless, the overall accuracy is still satisfactory for the application under investigation.

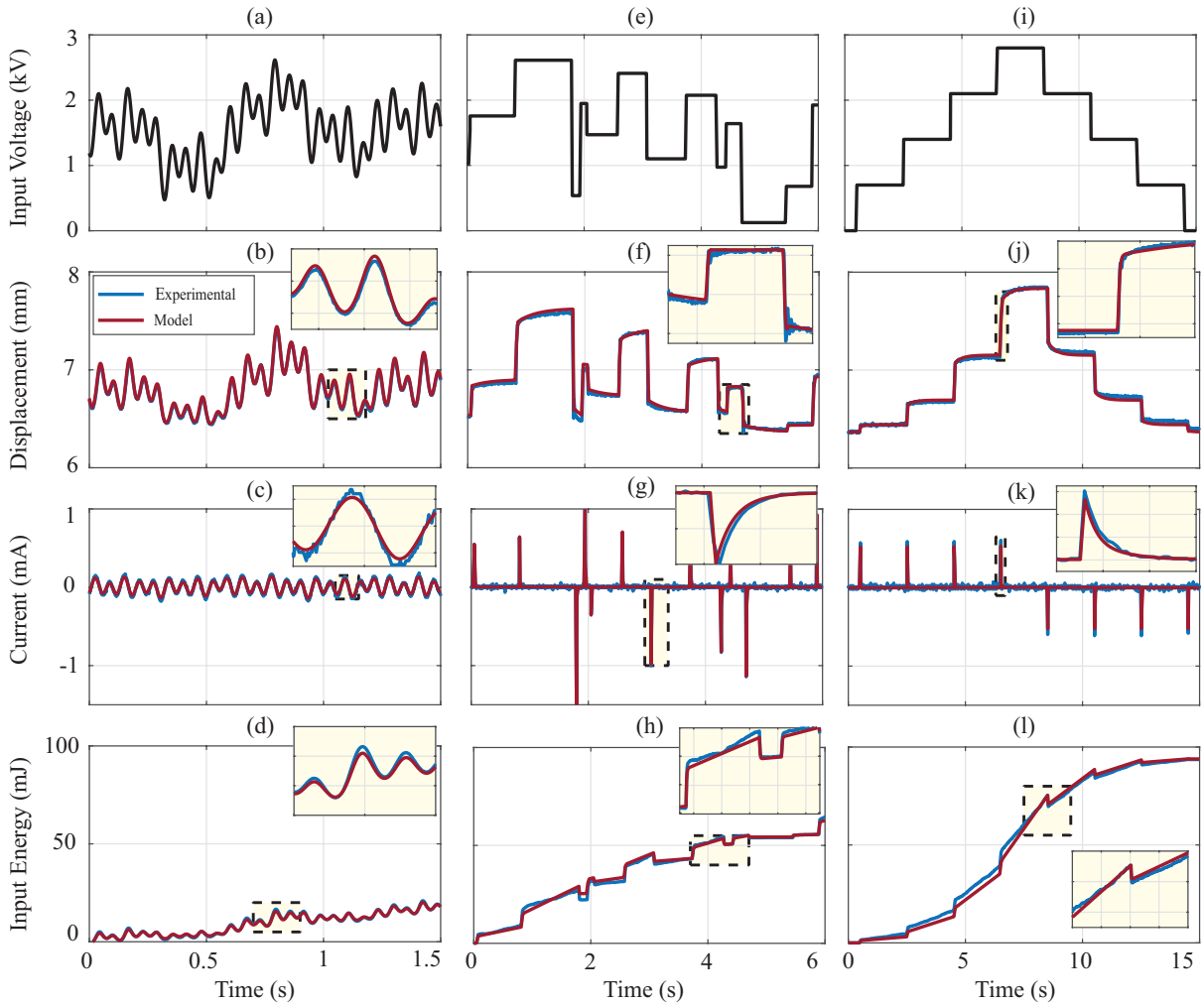

Figure 3.5: Experimental identification results: (a) Sum of sine waves signal; (b) Experimental and predicted displacements with sum of sine waves signal; (c) Experimental and predicted currents with sum of sine waves signal; (d) Experimental and predicted input energies with sum of sine waves signal; (e) APRBS signal; (f) Experimental and predicted displacements with APRBS signal; (g) Experimental and predicted currents with APRBS signal; (h) Experimental and predicted input energies with APRBS signal; (i) Steps signal; (j) Experimental and predicted displacements with validation signal; (k) Experimental and predicted currents with validation signal; (l) Experimental and predicted input energies with validation signal.

Table 3.3: FIT Values

Input Signal	Displacement FIT	Current FIT	Energy FIT
Sine waves sum	91.10	80.96	92.75
APRBS	93.51	75.99	93.47
Step wave	95.20	74.62	94.18

3.4 Energy Minimization via Adaptive Dynamic Programming

The design of an energy optimal control law for the DEA is addresses in this Section. The energy-related utility function of the corresponding energy optimal control problem is firstly defined. Then, and ADP with off-policy learning algorithm solves the intractable HJB equation.

3.4.1 Optimal Control Problem Formulation

Let's consider the following control problem: given an arbitrary initial condition z_0 and a target equilibrium point (z^*, v^*) for model (3.13), find a state feedback controller $v(z)$ which steers the state $z(t)$ from z_0 at $t = 0$ to z^* for $t \to \infty$ and, at the same time, minimizes the input energy consumption. As stated in Section 3.2.2, this problem can be tackled by minimizing the losses that occur during the actuation. Note that the target equilibrium point (z^*, v^*) can be uniquely specified once a desired displacement y^* is known. In fact, for any equilibrium state the corresponding equilibrium momentum p^* and viscoelastic states ε^*_{kj} are always zero. Additionally, once y^* is known the target charge q^* can be found by solving (3.3) at steady state. By plugging q^* and y^* into (3.11) the required voltage v^* can be finally found. The obtained values of y^*, q^*, and v^* allow then to uniquely determine any arbitrary equilibrium state.

Once the target configuration is known, it is convenient to introduce the state and input deviations as $x = z - z^*$ and $u = v - v^*$, respectively. By substituting x and u in (3.13), the following system is obtained

$$\dot{x} = f_e(x) + Bu, \tag{3.23}$$

where $f_e(x) = f(x + z^*) + Bv^*$. In this way, the problem of reaching the target equilibrium point (z^*, v^*) for the original model to can be converted in controlling (3.13) to the origin. This will simplify the formulation of the optimal control problem in the subsequent section. The energy loss function (3.19) can now be expressed with respect to the new set of coordinates in the following way

$$s_e(x, u) = s(x + z^*, u + v^*). \tag{3.24}$$

Analytically, (3.24) can be computed as follows

$$\begin{aligned} s_e(x,u) = {} & \frac{N_l}{R_l(y^*+x_1)C^2(y^*+x_1)}(q^*+x_{M+3})^2 + \frac{N_l}{R_e}\left(v^*+u-\frac{1}{C(y^*+x_1)}(q^*+x_{M+3})\right)^2 \\ & + \frac{\eta_{v0}N_lV_{DE}}{(\varepsilon_1(y^*+x_1)+1)}\left(\frac{1}{m}\left.\frac{d\varepsilon_1(y)}{dy}\right|_{y=y^*+x_1}\right)^2 x_2^2 \\ & + N_l\sum_{j=1}^{M}\frac{k_{vj}^2V_{DE}}{\eta_{vj}}\log\left(\frac{\varepsilon_1(y^*+x_1)+1}{\varepsilon_1(y^*+x_1)-x_{j+2}+1}\right)x_{j+2} \end{aligned} \tag{3.25}$$

where $x_1 = y - y^*$, $x_2 = p$, $x_{j+2} = \varepsilon_{kj}$, and $x_{M+3} = q - q^*$.

The problem of position regulation with energy minimization can be now defined according to the optimal control theory. Given system (3.23) properly defined in such a way its origin

corresponds to a given target displacement y^*, find a stable closed loop control policy $u(x)$ that minimizes for any initial state x_0 the following performance function

$$V(x_0) = \int_0^\infty \underbrace{\left(Q(x) + \alpha(x)u + Ru^2\right)}_{U(x,u)} dt, \tag{3.26}$$

where $Q(x)$, $\alpha(x)$ and R are defined in such a way $U(x,u) = s_e(x,u)$. Function $U(x,u)$ is the utility function, as in (2.3).

Although formally correct, the optimal control problem formulation discussed above has some issues which restrict its usefulness for the given applications. First, it can readily be verified that the selected utility function does not satisfy the condition $U(0,0) = 0$, thus making integral (3.26) diverge to infinity at steady-state. This fact makes sense from the physical point of view, since at steady-state the applied DC voltage v^* results in a non-zero current i^* flowing on resistors R_e and R_l which, in turn, produce a continuous Joule heating dissipation (cf. Fig. 3.3). Since this loss is unavoidable, the ideal function $s_e(x,u)$ is replaced with the following auxiliary dissipation function $s_a(x,u)$

$$\begin{aligned} s_a(x,u) = & \frac{N_l}{R_l(y^*+x_1)C^2(y^*+x_1)} x_{M+3}^2 + \frac{N_l}{R_e}\left(u - \frac{1}{C\left(y^*+x_1\right)}\left(x_{M+3}+q^*\right) + \frac{1}{C\left(y^*\right)}q^*\right)^2 \\ & + \frac{\eta_{v0}N_l V_{DE}}{(\varepsilon_1(y^*+x_1)+1)}\left(\frac{1}{m}\left.\frac{d\varepsilon_1(y)}{dy}\right|_{y=y^*+x_1}\right)^2 x_2^2 \\ & + N_l \sum_{j=1}^{M} \frac{k_{vj}^2 V_{DE}}{\eta_{vj}} \log\left(\frac{\varepsilon_1(y^*+x_1)+1}{\varepsilon_1(y^*+x_1) - x_{j+2} + 1}\right) x_{j+2}. \end{aligned} \tag{3.27}$$

Note that functions (3.25) and (3.27) only differ for the first two terms. It can be easily verified that $s_a(x,u)$ shares the same convexity property of $s_e(x,u)$ with respect to states $x_2, \ldots, x_{M+3}$. Therefore, $s_a(x,u) \geq 0 \ \forall x, u$ in the operating range and, additionally, $s_a(0,0) = 0$. Note also that $s_a(x,u)$ and $s_e(x,u)$ coincide in the limit case in which the DE behaves as a perfect dielectric, i.e., $\rho \to \infty$ so that $i^* = 0$. Since such parameter is commonly very large, it is reasonable to assume that $s_a(x,u)$ will only slightly differ from the ideal $s_e(x,u)$ for typical application scenarios.

The second issue concerns the type of dynamic specification imposed for the optimal control design. In particular, if the control goal is solely expressed in terms of energy consumption, it is expected that the resulting settling time would be unacceptably slow (ideally, in the limit case $\rho \to \infty$ an infinitely slow actuation would result in no dissipation). To properly address the trade-off between energy minimization and transient speed, the overall utility function is modified as follows

$$U(x,u) = s_a(x,u) + \gamma \frac{N_l}{R_e} u^2, \tag{3.28}$$

where $\gamma \geq 0$ is a tuning parameter. Note that (3.28) still has a structure compatible with (3.26). The larger γ, the bigger the penalty on large values of u in the resulting closed loop law. Since the total control input for the real system is expressed in terms of $v = v^* + u$, penalizing u implies that the closed loop system will respond similarly to the uncontrolled system subject to

a steady input voltage v^*. This results into a fast response, but also into a high dissipation. Note that the choice of adding a penalty term proportional to u^2, instead than some squared norm of the states, is motivated by the fact that the former cost function exhibits better numerical convergence properties during the controller design phase.

3.4.2 ADP Solves the Energy-Optimal Control Problem

The energy-optimal control problem defined as the minimization of (3.26), under the system dynamics in (3.23), cannot be solved via traditional methods due to the involved nonlinearities. Thus, the ADP procedure with off-policy learning in Algorithm 2.2 is employed as a tool to solve offline the energy-optimal control problem. Note that the utility function (3.28) can be rewritten in the form of (3.26) as follows

$$U(x,u) = \underbrace{\begin{pmatrix} \frac{N_l}{R_l(y^*+x_1)C^2(y^*+x_1)} x_{M+3}^2 + \frac{\eta_{v0} N_l V_{DE}}{(\varepsilon_1(y^*+x_1)+1)} \left(\frac{1}{m} \left. \frac{d\varepsilon_1(y)}{dy} \right|_{y=y^*+x_1} \right)^2 x_2^2 \\ + N_l \sum_{j=1}^{M} \frac{k_{vj}^2 V_{DE}}{\eta_{vj}} \log\left(\frac{\varepsilon_1(y^*+x_1)+1}{\varepsilon_1(y^*+x_1)-x_{j+2}+1} \right) x_{j+2} + \frac{N_l}{R_e} \left(\frac{x_{M+3}+q^*}{C(y^*+x_1)} \right)^2 \\ + \frac{N_l}{R_e} \left(\frac{q^*}{C(y^*)} \right)^2 - 2\frac{N_l}{R_e} \frac{x_{M+3}+q^*}{C(y^*+x_1)C(y^*)} q^* \end{pmatrix}}_{Q(x)} \tag{3.29}$$

$$+ \underbrace{2\frac{N_l}{R_e}\left(\frac{q^*}{C\left(y^*\right)} - \frac{x_{M+3}+q^*}{C\left(y^*+x_1\right)}\right)}_{\alpha(x)} u + \underbrace{\left(\gamma+1\right)\frac{N_l}{R_e}}_{R(x)} u^2$$

Therefore, the same procedure in Section 2.4.3 can be employed. The result is a set of weights approximating the optimal value function, i.e., the critic weights, and control policy, i.e., the actor weights, as in (2.29) and (2.30).

3.5 Experimental results

The experimental validation of the energy-optimal control strategy discussed in the previous section is presented in the following.

3.5.1 Learning Phase

Prior to evaluating the effectiveness of the optimal controller on the experimental DEA system, the ADP algorithm is first employed in a preliminary learning procedure conducted offline. In particular, the model validated in Section 3.3 and the ADP procedure in Algorithm 2.2 are implemented in the MATLAB/Simulink environment. The ADP approach is used as a tool to solve offline the HJB equation and obtain a nonlinear control policy for a given set of target displacement values. The optimal control policies are then implemented in the microcontroller used to drive the DEA, as discussed in Section 3.3. In order to conduct the data collecting phase in a proper way, a sufficient rich exploratory signal is required. To prevent damaging the real-life system due to intensive training experiments, the ADP algorithm uses the data obtained from

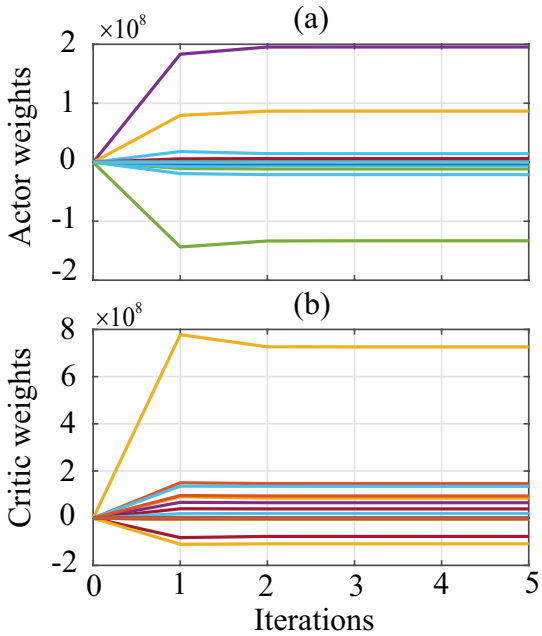

Figure 3.6: Weights convergence for the examined actuator when y^* = 7.73 mm and γ = 0.05: (a) Actor weights convergence; (b) Critic weights convergence.

offline simulations of the validated model instead of physically apply the exploration signal to the DEA.

Given a steady-state value of the input voltage v^*, the design of the mechanical biasing system ensures that the considered DEA exhibits a unique equilibrium displacement y^*. Since the relationship between v^* and y^* is one-to-one, we can alternatively specify y^* and compute the corresponding feedforward control input v^*. Once equilibrium input and output are known, the corresponding equilibrium state can be uniquely determined by exploiting the model equations. By using a passivity argument, we can conclude that this unique equilibrium state is always stable, provided that the initial conditions are chosen in a physically meaningful range. This can be proved by defining a new storage function given by $\Psi_a - v^* q_i$, with $\dot{q}_i = i$, and exploiting equation (3.20) for the case in which $v = v^*$ holds. Based on the above discussion, it can be concluded that, given a target displacement value y^*, the origin of system (3.23) is stable when $u = 0$. This allows to employ $u_0 = 0$ as the initial stabilizing policy in each learning phase conducted on (3.23). The learning noise, $e_L(t)$, is selected as a sum of sine waves with different amplitudes and frequencies. The number of learning steps is $N_L = 10000$, each one of them having a duration of 1 ms. A total of 65 polynomial terms from the second to the fourth degree in the four states are considered as basis function for the critic network, while 34 polynomial terms from the first to the third degree are considered as basis functions for the actor network. As an example, the actor and critic weights convergence is shown in Fig. 3.6(a) and 3.6(b), respectively, for a value of y^* = 7.73 mm and γ = 0.05. Note that the large values of both actor and critic weights are due to the involved physical quantities, since each one of them is characterized by a different unit and range on a different scale of values. For instance, the electric charge error, $x_4 = q - q^*$, and displacement error, $x_1 = y - y^*$, are on the order of μC and mm, respectively, while the applied voltage is in the order of kV.

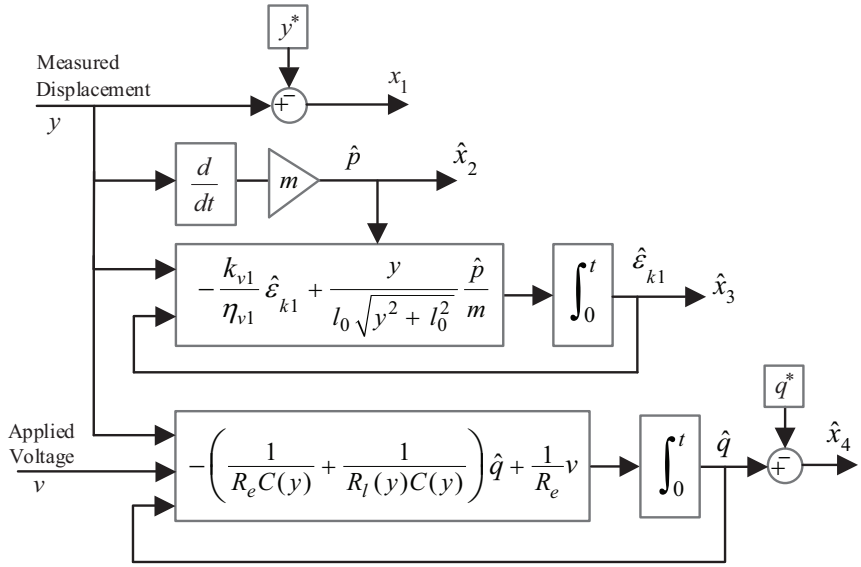

Figure 3.7: Graphical representation of the observer equations.

3.5.2 Results

As stated in Section 3.3, the experimental bench provides displacement, voltage, and current measurements. In order to implement a full state feedback control law, an observer is implemented in the microcontroller. The displacement error, x_1, is measured given the value of y^*. Estimated state variable $\hat{x}_2$, i.e., $\hat{p}$, is easily obtained by exploiting the definition of momentum, through a numerical differentiation. Observed viscoelastic state $\hat{x}_3$, i.e., $\hat{\epsilon}_{k1}$, is obtained by integrating (3.9) once the position feedback and $\hat{x}_2$ are plugged in. Finally, the observed state $\hat{x}_4$ is reconstructed by plugging the measured displacement y, and input voltage v, into the current relation shown in (3.11) and subtracting the resulting observed charge, $\hat{q}$, from the reference value q^*. The observer equations are reported in Fig. 3.7. Convergence of the estimated states to the real values can be easily proven.

In the following, the displacement values y are express as absolute values or as normalized values with respect to the maximum and minimum displacement of the DEA. Let y_{min} and y_{max} be the equilibrium displacement values obtained when applying constant input voltages v_{min} = 0 kV and v_{max} = 3 kV, respectively. Then, for any displacement value $y \in [y_{min},\ y_{max}]$, the following normalized displacement is defined

$$\lambda := \frac{y - y_{min}}{y_{max} - y_{min}} \in [0, 1]\,. \tag{3.30}$$

Experiments are conducted to validate the energy minimization capabilities of the proposed approach. Each optimal control policy is compared to both the open loop behavior and with a Proportional-Integral, hand-tuned in such a way the resulting displacement trajectory has settling time comparable with the other experiments. In particular, the Proportional-Integral controller is combined with a simple nonlinear term which compensates the quadratic nonlinearity of the DEA transduction principle. This allows improved performances if compared with traditional PID controllers, as reported in earlier studies [50, 94]. In order to compensate model uncertainties at steady-state, the actuated control law is smoothly switched from ADP to

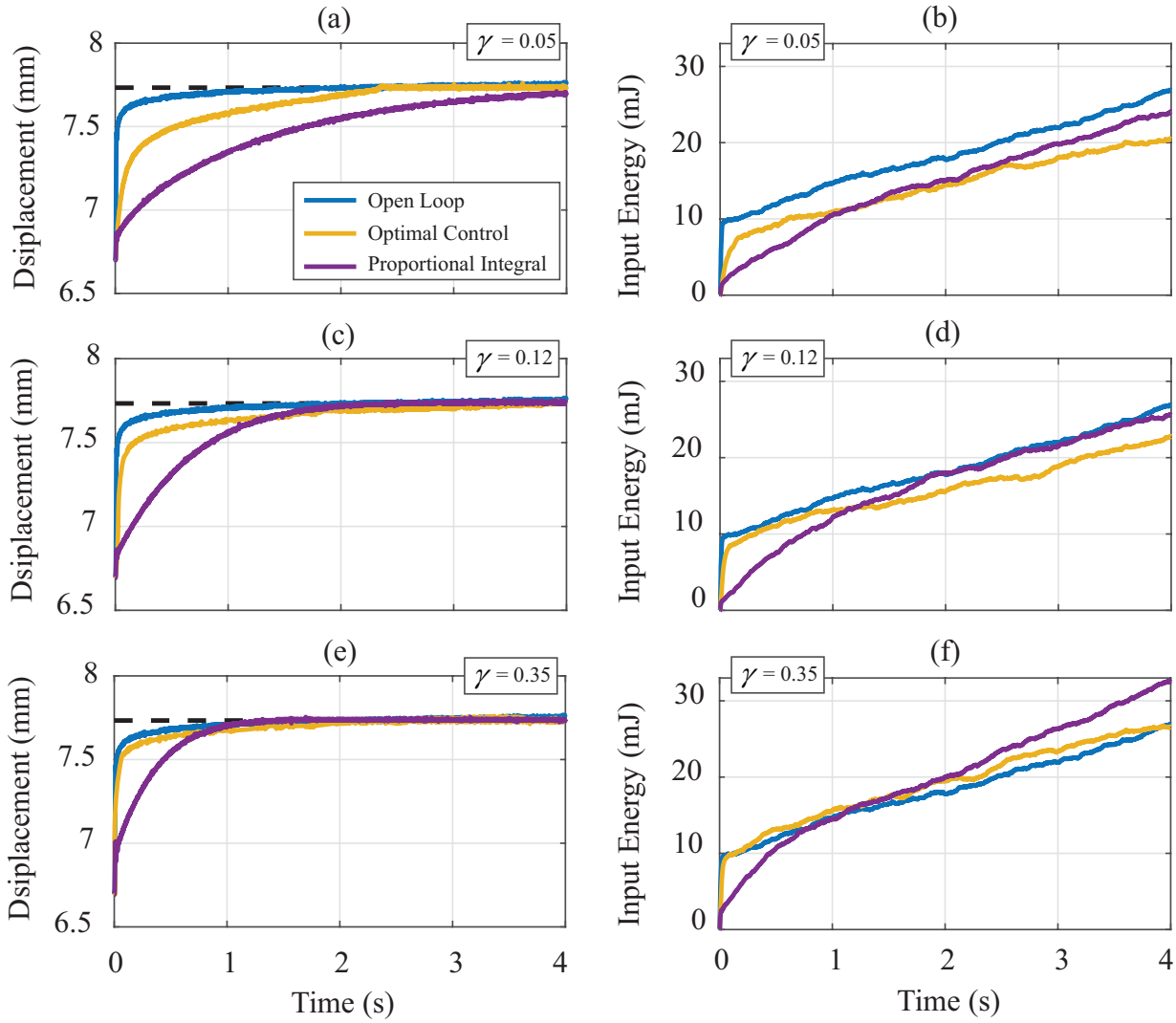

Figure 3.8: Experimental results comparison of open loop, ADP, and Proportional-Integral controllers when $\lambda_0 = 0.2$ and $\lambda^* = 0.8$: (a) Measured displacements when $\gamma = 0.05$; (b) Measured input energies when $\gamma = 0.05$; (c) Measured displacements when $\gamma = 0.12$; (d) Measured input energies when $\gamma = 0.12$; (e) Measured displacements when $\gamma = 0.35$; (f) Measured input energies when $\gamma = 0.35$.

the Proportional-Integral once the position error x_1 gets in a predefined band around the target point, i.e., $|x_1|/y^* \leq 0.01$. As shown in the experimental tests, the switch between the two control policies doesn't affect the energetic performances since they mainly occur during transients. Figure 3.8 shows the results according to an initial displacement $y_0 = 6.67$ mm ($\lambda_0 = 0.2$) and a target displacement $y^* = 7.74$ mm ($\lambda^* = 0.8$), with three different values of $\gamma = 0.05$, $\gamma = 0.12$, and $\gamma = 0.35$. As expected, bigger values of γ imply faster responses (Fig. 3.8(a), 3.8(c), and 3.8(e)) but, at the same time, higher energy consumption evaluated by integrating the product of measured voltages and currents (Fig. 3.8(b), 3.8(d), and 3.8(f)). The measured input energy values are reported in Table 3.4 along with the parameters of each Proportional-Integral controller used for the comparison. In the first two examined cases of Fig. 3.8 ($\gamma = 0.05$, and $\gamma =$

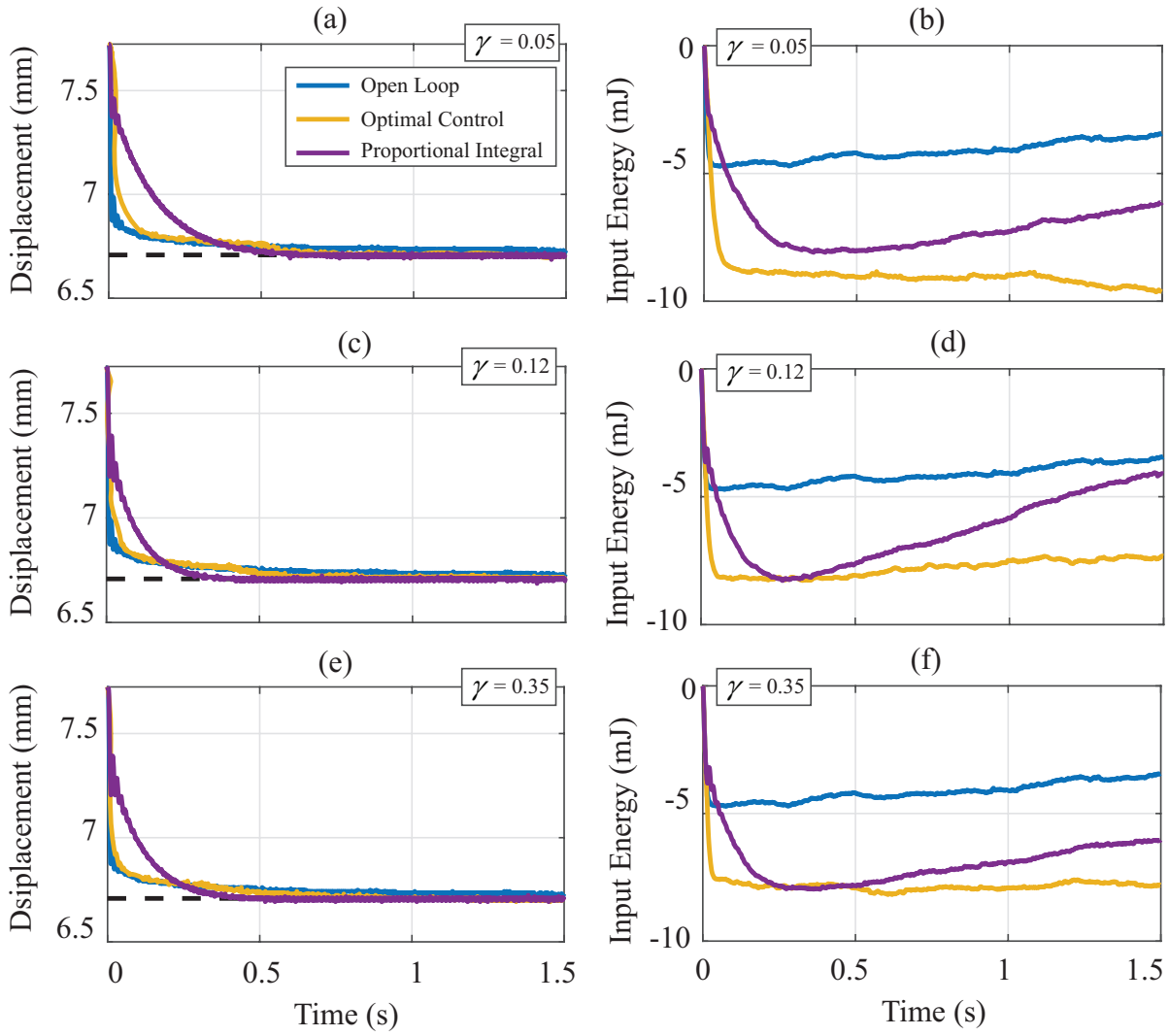

Figure 3.9: Experimental results comparison of open loop, ADP, and Proportional-Integral controllers when $\lambda_0 = 0.8$ and $\lambda^* = 0.2$: (a) Measured displacements when $\gamma = 0.05$; (b) Measured input energies when $\gamma = 0.05$; (c) Measured displacements when $\gamma = 0.12$; (d) Measured input energies when $\gamma = 0.12$; (e) Measured displacements when $\gamma = 0.35$; (f) Measured input energies when $\gamma = 0.35$.

0.12), the open loop controller provides the fastest settling times with the corresponding highest energy consumption values, if compared with both Proportional-Integral and ADP controllers. Proportional-Integral controllers show lower energy consumption values due to their slower dynamics and filtering features. Note also that the shape of the Proportional-Integral closed loop response is significantly different from the one of ADP, due to the strong nonlinearities of the investigated system. For $\gamma = 0.35$, the Proportional-Integral controller shows the highest input energy despite its slower dynamics, if compared with both ADP and open loop. This is mainly due to the higher values of K_p and K_i (see Table 3.4, third row) that result in more aggressive responses. Therefore, despite Proportional-Integral controllers provide feedback policies easy to implement with good performances in steady-state compensations, their energy consumption are unpredictable. Due to the nonlinearities of the DEA, the trade-off between response time

and energy consumption is hard to tune in case of Proportional-Integral controllers. This justify the employment of the proposed ADP approach that provides energy minimization capabilities as well as easy response time tuning through the single parameter γ. Finally, note that ADP controllers show best performances in terms of energy consumption, with settling times always higher than open-loop control.

Figure 3.9 presents the results with inverted values of initial and target displacement, i.e. $\lambda_0 = 0.8$ and $\lambda^* = 0.2$. The same values of γ are used. In this case, since the target displacement is below the initial one, the equivalent capacitor will discharge with a corresponding negative input energy that can be recovered. As in Fig. 3.8, bigger values of γ imply faster responses (Fig. 3.9(a), 3.9(c), and 3.9(e)) and lower energy recovering evaluated by integrating the product of measured voltages and currents (Fig. 3.9(b), 3.9(d), and 3.9(f)). The same considerations of the previous case are still valid, except for γ = 0.12. In fact, in this case while the settling time is slower than the response obtained with γ = 0.35, the resulting restored energy is lower than the value obtained with the same value of γ = 0.35. This is due to the nonlinear behavior of the DEA during discharging phases. However, as in the previous experiments, ADP policies show the best performances in term of recovered energy. For these experiments, the settling times provided by ADP are always higher than the ones obtained in open loop. Note that, in all of the considered scenarios, the experimental tests for both identification and control are performed in a lab environment under repeatable conditions. For this reason, the open loop controllers succeed in reaching the set-point with a remarkably small steady error.

Table 3.4: Experimental Results

					Input Energy [mJ]			
λ_0	λ^*	γ	K_p	K_i	ADP	Prop. Integr.	OL	$ADP/OL\%$
		0.05	0.1	1.5	20.4	24	27	-24.44
0.2	0.8	0.12	1	10	22.9	25.5	27	-15.19
		0.35	2	20	26.7	32.5	27	-1.11
		0.05	2	50	-9.8	-3.9	-2.1	$+366.67$
0.8	0.2	0.12	3	90	-6.5	-2.5	-2.1	$+209.52$
		0.35	3	75	-7.55	-5.21	-2.1	$+259.52$

3.5.3 Robustness Analysis

The approximated energy-optimal control policy is found through offline simulations based on the identified model. Clearly, the learning model depends on the identified parameters shown in Table 3.2. In real-life conditions, some of those parameters may not be exactly known, or may change over time due to several reasons, e.g., changes in environmental conditions, material aging process. To ensure the correct functioning of ADP in real-life settings, a robustness analysis is required to evaluate the performances of the obtained approximated policy when the system parameters are varying.

Simulation studies are conducted to evaluate the performances variation in terms of resulting energy consumption and settling time when identified parameters, i.e., k_{v1}, η_{v1}, η_0, β_1, β_2, β_3, γ_1, γ_2, γ_3, ϵ_r, R_e, and ρ vary from the values reported in Table 3.2. All these parameters are intrinsically related to the constitutive behavior of the DE material, and therefore it is reasonable

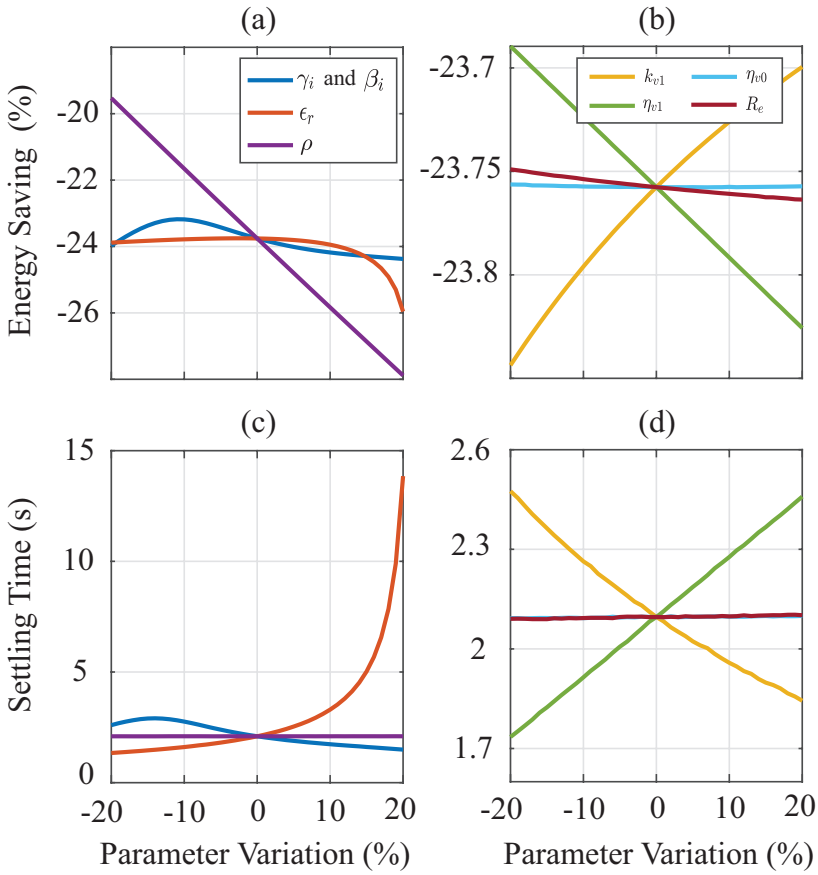

Figure 3.10: Robustness analysis results: (a) Energy saving percentage with respect to open loop control when β_i, γ_i, with $i = 1, 2, 3$, ϵ_r, and ρ are varying; (b) Energy saving percentage with respect to the open loop when k_{v1}, η_{v1}, η_0, and R_e are varying; (c) Resulting settling times when β_i, γ_i, with $i = 1, 2, 3$, ϵ_r, and ρ are varying; (d) Resulting settling times when k_{v1}, η_{v1}, η_0, and R_e are varying.

to assume that they may slightly change over time. Each parameter is changed within the range $[-20\%, +20\%]$ of the identified value, while all the other parameters are kept constant to their nominal values. Parameters β_i and γ_i, with $i = 1, 2, 3$, are all increased or decreased together, since they can be interpreted as a stiffness value (see (3.10)). Resulting performances are evaluated using the same approximated optimal policy used in the previous subsection with $\lambda_0 = 0.2$, $\lambda^* = 0.8$, and $\gamma = 0.05$. Figures 3.10(a) and 3.10(b) show on the horizontal axis the parameter percentage variation, while on the vertical axis the energy saving percentages with respect to the open loop control. The energy savings are computed considering as input energy the integral of the product between simulated current and simulated input voltage. When no parameters are changing the simulated energy saving is -23.76%, in line with the experimental value in Table 3.4, i.e., -24.44%. Changes in the settling time (within 2%) according to each parameter variation are shown in Fig. 3.10(c) and Fig. 3.10(d).

Figures 3.10(a) and 3.10(c) show how performances are mainly affected both in terms of energy saving and settling time by variations in β_i, γ_i, with $i = 1, 2, 3$, ϵ_r, and ρ parameters. In particular, the linear dependency of the energy saving with respect to changes in ρ (Fig.

3.10(a)) highlights the major source of dissipated energy for the considered actuator, i.e., the leakage resistance $R_l(y)$ in (3.12). Clearly, as in (3.19), for increasing (decreasing) values of ρ the losses on the leakage resistance decrease (increase) resulting in increased (decreased) energy savings. When ρ varies, the settling time is not affected (Fig. 3.10(c)) since it mainly depends on the mechanical dynamics. Major changes in both energy saving and settling time are observed when β_i, γ_i, with $i = 1, 2, 3$, and ϵ_r are varying, implying that the approximated optimal policy highly depends on the corresponding values used in the learning stage. Note the increasing value of energy saving when ϵ_r varies more than $+10\%$, as a consequence of the increased settling time. Therefore, it can be stated that the strongest nonlinearities depend on β_i, γ_i, with $i = 1, 2, 3$, and ϵ_r, and, thus, a wider exploration of the system states when those parameters are changing might be necessary in the learning phase. Finally, as shown in Fig. 3.10(b) and Fig. 3.10(d), ADP provides good robustness performances when k_{v1}, η_{v1}, η_0, and R_e are changing, with no substantial variations in both the energy saving and settling time values. Note that when k_{v1} increases (decreases) or η_{v1} decreases (increases), a faster (slower) time response is obtained. This is easily explained since k_{v1}/η_{v1} can be seen as the time constant of the viscoelastic state (see (3.9)).

3.6 Conclusions

This work investigated the minimum energy position control of DEAs. A free-energy model properly describes the energy dissipation in a thermodynamically consistent way. After satisfactory experimental identifications, the energy losses are used as utility function in an optimal control problem. To address nonlinearities in the actuator's model and utility function, an ADP algorithm with off-policy learning solves offline the HJB equation. Experimental validations verified the effectiveness of the proposed approach for different target displacement scenarios. When compared with traditional control techniques, our method provides optimal charging policies, with energy savings up to 20 $\%$, as well as optimal discharging policies, with energy restoring over 300 $\%$. The trade-off between energy consumption and settling time is easier to predict and tune using the proposed approach instead of traditional Proportional-Integral controllers. Finally, a robustness analysis shows how the optimal controller performances are mainly affected by changes in only the Ogden model's stiffness, the material relative permittivity, and the elastomer resistivity.

3.7 Publications

The results presented in this chapter have been published by the author in [104], [105], and [106].

Chapter 4

Distributed Assistive Control of Power Buffers in DC Microgrids

This chapter presents the second application of the Adaptive Dynamic Programming (ADP) approach for the optimal control of complex systems. The distributed control of power buffers in a direct current (DC) microgrids is considered. Modern-day renewable energy sources and loads, such as photo-voltaic generators and electric vehicles find their natural surroundings in DC microgrids. The stability of such distribution systems may result weak due to the absence of damping elements. Power buffers are power electronic converters with large storage devices (e.g. capacitor) used to decouple volatile loads and distribution system, enhancing stability and performances. Normally, the input impedance and the stored energy of power buffers are adjusted in a localized fashion. By letting the power buffers communicate, the effective range of action of each power buffer can be extended to its neighboring loads. In this way, neighboring nodes can assist each other during abrupt load changes.

In this chapter, distributed assistive control policies are developed according to the optimal control theory, with a common objective shared among the network. Such approach enables power buffers to reciprocally assist each other during abrupt load changes in a cooperative fashion. Adaptive dynamic programming (ADP) algorithms with off-policy learning deal with the nonlinear dynamics dictated by the distribution grid.

Based on the configuration of the communication network deployed on top of the physical distribution grid, two different studies are conducted in this chapter. First, the communication topology is fixed and inspired by the physical vicinity of the buffers. The fully nonlinear dynamics is considered and a set of optimal control policies are learned offline and then interpolated in a real-time control scheme according to the the desired network's operating point. Alternatively, a second study considers the communication topology a free parameter subject to optimization. In particular, it is desired to reduce the interactions between different control loops of power buffers while minimizing a closed-loop performance function. In both cases, ADP with off-policy learning algorithms will represent the key tool to solve such problems.

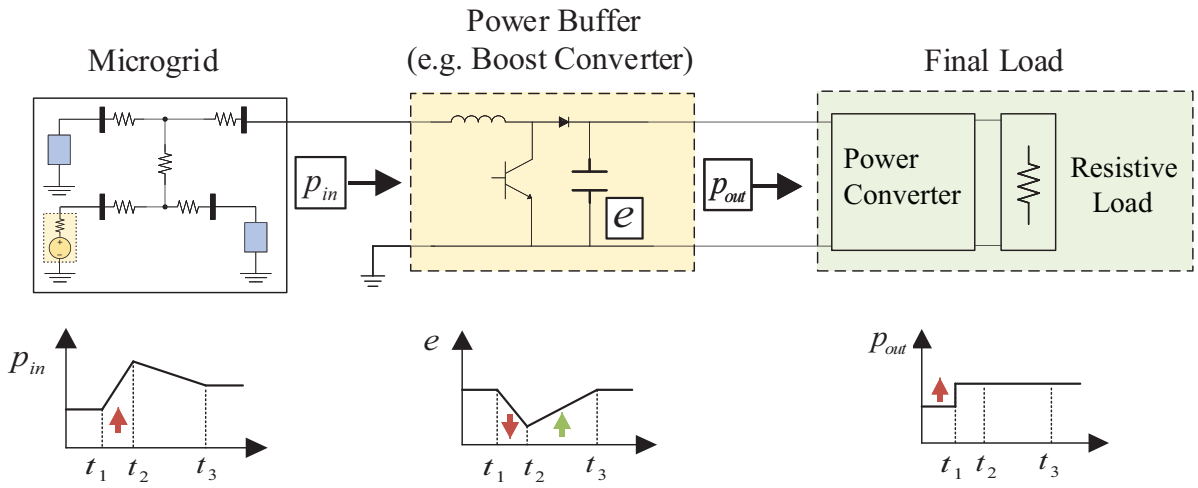

Figure 4.1: Power buffer operating principle.

4.1 Overview and Objectives

4.1.1 Power Buffers for Load Decoupling

The increasing penetration of DC sources and loads, such as photo-voltaic, electrical vehicles, data-centers, and batteries, is naturally integrated within DC microgrids [107], [108]. When compared with their AC counterpart, DC microgrids provide a more reliable and efficient distribution paradigm by avoiding unnecessary conversion stages [109–112]. Furthermore, DC distribution systems are not subject to common ac-related issues such as frequency synchronization, reactive power flows, and transformer inrush currents [113]. Nevertheless, DC microgrids face a compound challenge of handling potentially volatile source and load profiles while having a resistive grid with low damping/generational inertia, leading to stability issues [114], [115]. To tackle these issues, both hardware and control approaches have been studied. While the former require costly central energy storage elements to decouple loads and network [116], [117], the latter make use of both average and hybrid models for the switching converters involved in the network [118], [119].

Power buffers can be used as a damping element to enhance the stability properties of a DC microgrid. Firstly introduced in [120] and [121], power buffers are power converters with large storage elements (e.g. capacitors) able to decouple the load from the distribution grid. During transients, power buffers can shape the input power profile by modifying the load impedance seen by the distribution network. The stored energy is used to compensate the transient mismatch, hence shielding the distribution network from abrupt load changes [55], [122].

Figure 4.1 summarizes the operating principle of a boost converter used as a power buffer. At $t = t_1$ the final load, typically constituted by the series connection of a power converter and a final resistive load, changes the demanded power p_{out}. The power buffer supplies the extra power demand using its stored energy e, smoothing the input power p_{in} drawn from the network. At $t = t_2$, the input power matches the demanded one and the converter switches mode from buffering to energy recovering. In this phase, i.e. $t_2 < t < t_3$, the buffer slightly draws extra power from the network to recover lost energy. The recovering phase ends in $t = t_3$, and for $t > t_3$ the input power equals the demanded one. The stored energy, e, remains at its original

level until the next load change occurs.

Since power buffers are placed at load terminals, they exhibit more efficient behaviors as well as faster responses when compared with central energy storage devices. Moreover, power buffers provide an additional degree of freedom that can be exploited to design control laws that improve the overall network performances.

4.1.2 Existing Control Techniques

The majority of existing control solutions use a game-theoretic control framework with power buffers as players, using different control objectives and solution approaches [123–129]. Control objectives include to meet power or voltage drop requirements in [123], to achieve a constant power characteristics while minimizing network losses in [125], to find optimal controllers with respect to quadratic functionals at each sample time in [126], to conserve as much energy as possible while preventing system collapse in [128], or to simply conserve the buffer's stored energy in [129]. In the absence of a closed-form solution to the game-theoretic problem, a turn-based approach is employed in [123, 128] that could adversely affect the controller performance and stability as the system size increases. Alternatively, the game-theoretic solutions are found in [125, 129] using Pontryagin's minimum principle, with sliding-mode controllers used to actuate the resulting open-loop optimal trajectories. In [126], the solution is found by the means of linear optimal control approaches.

Some of these solutions are implemented in a decentralized fashion, relying on individual objectives with non-cooperative strategies [123, 125, 126, 128, 129]. Communication-based cooperative methods are presented in [124, 127] as an alternative to non-cooperative solutions. In particular, in [124] a Policy Iteration algorithm solves the linear coupled Riccati equations, where the individual objectives are defined with regards to team-aligned and selfish components. In [127], a turn-based approach implements the solution of a leader-follower Stackleberg game to prioritize leader's objective, and finds an optimum set of information to be transmitted.

An assistive control strategy, based on linear distributed approaches, is presented in [130] where, as in [124], the coupling effects of the power distribution grid are considered. However, both [124] and [130] rely on small-signal approximations of power buffers. Although such approach allows to easily obtain a control law, optimality, robustness, and stability are not guaranteed for high load variations. Hence, the controller performances are limited to small load variation ranges. Alternatively, the proposed approach considers the nonlinear dynamics of both loads and distribution network to design distributed closed-loop controllers based on optimal control theory.

4.1.3 Distributed Controllers with Fixed Communication Topology

The communication topology plays a fundamental role in designing and implementing a distributed control policy for the power buffers. The first study is conducted by considering an *a priori* fixed communication topology, inspired by the physical vicinity. The objective is to design a distributed assistive control scheme for power buffers that considers the fully nonlinear dynamics. The control scheme is *distributed* as buffers exchange information through the communication network, *cooperative* as buffers share a common objective, and *assistive* as buffers reciprocally assist each other during abrupt load changes, improving overall network perfor-

mance and stability. Cooperative control techniques are already applied to other domains, e.g., unmanned aerial vehicles [131, 132], robot manipulators [133], and spacecrafts [134], and have recently been extended to DC microgrids (e.g., distributed primary/secondary control [135]).

In the proposed approach, the assistive control problem is formulated using the optimal control theory. Since the fully nonlinear dynamics is considered, an ADP algorithm with off-policy learning is employed to solve the corresponding Hamilton-Jacobi-Bellman equation, as discussed in Chapter 2. ADP has also provided optimal energy management policies in smart grids, e.g., see [136, 137]. In [138] and [139], a discrete-time ADP algorithm solves the optimal energy management problem for microgrids with energy storage elements. In [140], the fair energy scheduling problem for a vehicle-to-grid network is solved via ADP. A self-learning ADP algorithm in [141] considers the real-time electricity price, load demand, and solar energy. Continuous-time on-policy ADP approaches provide reactive power control in wind farms [142] and improve unmatched disturbance rejection in multi-machine power systems [143]. Continuous-time ADP algorithms, based on concurrent-learning, develop droop-free control for DC microgrids [144]. The game-theoretic solution for power buffers in [124] are provided via a policy iteration algorithm.

In the work reported in this chapter, the HJB equation is solved by employing a continuous-time ADP approach with off-policy learning for the purpose of feedback design instead of operational scheduling. A set of distributed optimal control policies able to provide assistance during abrupt load changes is derived. The communication network augments the assisting range of power buffers to nearby loads. Salient features and contributions of this work are

- The control law's weights sets are calculated based on a mesh of reference loads for each power buffer.
- To further reduce both computational requirements and communicated data, the controller is triggered only when a load change occurs, making it suitable for Internet-of-Things (IoT) devices.
- The distributed controller minimizes a shared objective among power buffers in a cooperative fashion, as opposed to non-cooperative strategies in [123, 125, 126, 128, 129].
- The feedback strategy is designed according to the optimal control theory, providing a real-time controller that is known a priori and does not need a turn-based approach as in [123, 127, 128].
- Compared to the work that rely on a small-signal approximation of power buffers [124, 126, 130], the proposed nonlinear optimal control law takes into account the nonlinear dynamics of the power buffers and the coupling power grid, and is valid for large-signal variations.
- It does not solve linear-quadratic regulator (LQR) problems at each sampling instant as in [126].
- The optimal control problem is solved by approximating the solution to HJB equation, instead of employing the Pontryagin's minimum principle as in [125] and [129]. This provides both necessary and sufficient conditions for optimality instead of only the necessary condition. Moreover, it provides a closed-loop control law directly implemented

without the need of other control techniques (e.g., sliding mode), offering simpler designs as well as better performances with small parameter variations or model uncertainty [5,6].

4.1.4 Optimizing the Communication Topology - The Sparsity Promoting Problem

An alternative approach considers the communication topology, i.e., the configuration of the active communication links, as a free parameter subject to optimization. When the communication topology is inspired by physical vicinity [53, 54, 129], the underlying physical interconnection reflects the fixed communication topology with no guarantees that these structures (physical and communication) are optimal with regard to control objectives. Given the limited energy available, co-optimization of control solutions and communication topologies, considering the distribution grid, is important.

Sparsity-promoting algorithms guarantee stability and performance without any *a priori* defined communication topology, i.e., few but crucial communication links are found [145, 146]. Similar to AC systems [58,147], power buffers can benefit from reducing the interactions among feedback loops, minimizing communication costs with a limited impact on the closed-loop performance. Existing sparsity-promoting methods for microgrids mostly rely on linear approaches in AC systems. Based on the linear formulation in [145], decentralized controllers for AC networks with voltage-source converters are designed in [148], while sparse and block-sparse wide-control architectures for AC systems are designed in [58] and [149], respectively. By extending [145] to discrete-time systems, the sparsity-promoting controller in [150] regulates the active power flows and frequency. In [147], decentralized and sparse wide-area controllers are designed to damp inter-area oscillations in AC systems using the convex relaxation of a linear H_∞ problem. Constrained Linear Quadratic Regulator (LQR) formulation finds an optimal controller for predefined communication structures to damp inter-area oscillators in [151]. In [152], a sparsity-promoting linear optimal controller is applied to an AC power system with synchronous machines. However, such formulations are not practical for nonlinear systems as in the case of DC microgrids with power buffers.

In this chapter, the ADP with off-policy learning method is exploited to develop a sparsity-promoting optimal design algorithm for nonlinear systems. The approximated solution of the HJB equation is learned using only system collected data and without the need for the exact knowledge of the system dynamics.In particular, the same set of collected data is repetitively used to find optimal controllers for different communication topologies. Note that stability of optimal designs with sparsity-promoting or structural constraint is not guaranteed even for linear systems [58]. the proposed approach employs Domain-of-Attraction (DoA) estimation methods [66–68] to check the stability of each distributed controller. To deal with the resulting combinatorial problem, a Tabu Search (TS) approach that avoids local minima is used [153, 154]. The main contributions of this work are

- The first attempt to solve nonlinear sparsity-promoting and structured optimal control problems using a data-driven algorithm based on RL and TS methods is developed.

- The obtained controllers are employed in DC microgrids, with a limited impact when comparing incrementally-sparse and fully-connected communication topologies. This

impact is shown to increase if existing techniques used for AC systems, e.g., [58], are applied.

- In contrast to [53, 54], the underling physical interconnection structure dictated by the distribution grid is considered. The communication topology is a free parameter subject to optimization, where the number of active communication links and a closed-loop cost function are simultaneously minimized.
- Sparsity-promoting and communication topology-related studies for power buffers are conducted. The reciprocal assistance among power buffers is shown to increase with a less sparse communication structure.

Finally, Controller/Hardware-In-the-Loop (CHIL) studies validate the effectiveness of the two proposed approaches.

4.1.5 Chapter's Outline

This chapter is organized as follows. Section 4.2 presents the nonlinear model of a DC microgrid with power buffers. Section 4.3 presents the design procedure of distributed assistive controllers based on ADP, when the communication topology is defined a priori and considering the fully nonlinear dynamics. CHIL studies are presented to validate the proposed approach. The sparsity-promoting algorithm and its application to the DC microgrid with power buffers is presented in Section 4.4. As in the case of predefined communication structure, CHIL studies are conducted to verify the effectiveness of the proposed sparsity-promoting approach. Finally, concluding remarks are reported in Section 4.5.

4.2 Nonlinear Dynamic Model of a DC Microgrid

Distribution lines, *active loads*, and DC sources constitute the DC microgrid, as depicted in Fig. 4.2(a). $r_{i,j}$ denotes the resistance between buses i and j. A power buffer connected with a *final load*, i.e., a point-of-load converter (POLC) and a resistive load (Fig. 4.2(b)), constitutes an active load [54]. A localized control approach limits the assistance capabilities of a buffer to its final load. Introducing a communication network among nearby active loads, as shown in Fig. 4.2, allows them to collectively respond to transients. A resistor, r_{si}, in series with a voltage source, v_{si} (Fig. 4.2(c)), model a DC source.

Let the number of active loads and sources be N and M, respectively, and the set of active loads be $\mathcal{L} = \{M+1, ..., M+N\}$. For the ith active load, r_i, v_i, e_i, and p_i are the input impedance, input voltage, stored energy, and power supplied to the final load, respectively. Thus, the energy-balance equation for the generic power buffer i is given as

$$\dot{e}_i = \frac{v_i^2}{r_i} - p_i, \quad i \in \mathcal{L}. \tag{4.1}$$

For the power buffer i, the energy-voltage relation can be expressed as follow

$$e_i = \frac{1}{2} C v_{bi}^2, \quad i \in \mathcal{L}, \tag{4.2}$$

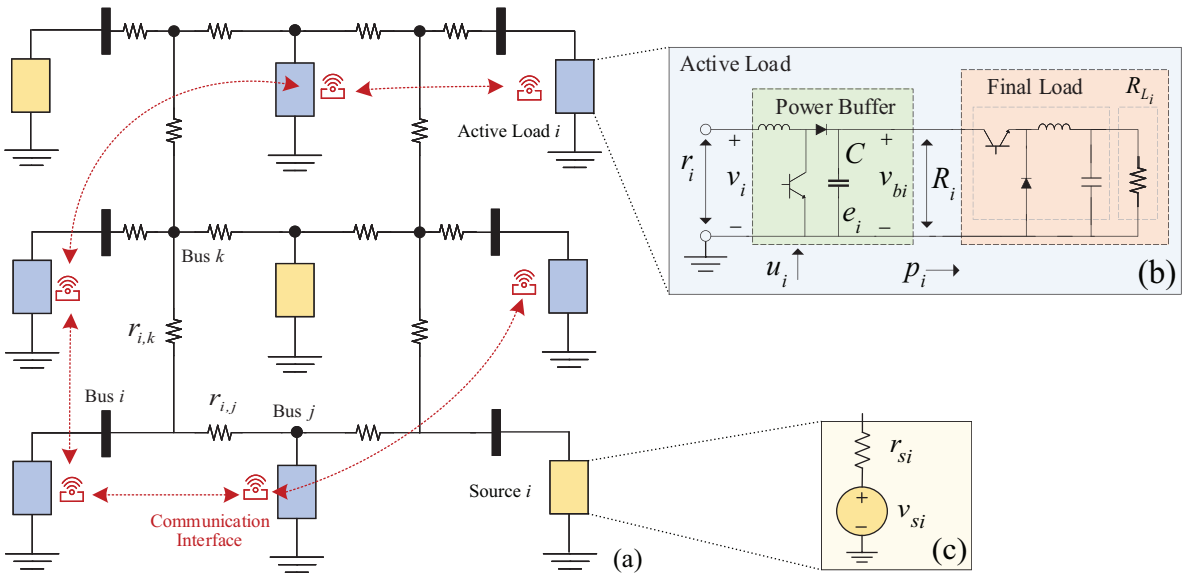

Figure 4.2: DC microgrid and its elements: (a) DC microgrid; (b) Active load consisting of a power buffer and a final load; (c) Model of a DC source.

where C is the capacitance of the power buffer, and v_{bi} is the buffer's output voltage. By defining u_i as the control input that regulates the input impedance of the buffer, r_i, the following state-space model for the ith active load is obtained

$$\begin{cases} \dot{e}_i = \frac{v_i^2}{r_i} - \frac{2e_i}{C}\frac{1}{R_i} \\ \dot{r}_i = u_i \end{cases}, \quad i \in \mathcal{L}. \tag{4.3}$$

R_i is the equivalent resistance of the buffer's output. Given a POLC (e.g., buck converter), in the steady state, R_i can be obtained from the load's resistance, R_{L_i}.

Sources are modeled as a series connection of a voltage source, v_{si}, and a resistor, r_{si}. The admittance matrix of a distribution grid relates its injected nodal currents and the bus voltages. Likewise, the active loads and sources can be related

$$i = \begin{bmatrix} v_{s1}/r_{s1} \\ \vdots \\ v_{sM}/r_{sM} \\ \hline 0 \\ \vdots \\ 0 \end{bmatrix} = Y \begin{bmatrix} v_1 \\ \vdots \\ v_M \\ \hline v_{M+1} \\ \vdots \\ v_{M+N} \end{bmatrix}, \tag{4.4}$$

where i is the vector of injected currents, and Y is the reduced-order admittance matrix [155]. From (4.4), the input voltage of an active load can be related to the input impedances of all active loads:

$$v_i = \zeta_i(r_{M+1}, ..., r_i, ..., r_{M+N}), \quad i \in \mathcal{L}. \tag{4.5}$$

The resulting dynamic model for an active load becomes

$$\begin{cases} \dot{e}_i = \frac{\zeta_i(r_{M+1},\cdots,r_{M+N})^2}{r_i} - \frac{2e_i}{C}\frac{1}{R_i} \\ \dot{r}_i = u_i \end{cases}, \quad i \in \mathcal{L}. \tag{4.6}$$

Given a set of output loads $\left[R_{L_{M+1}} \cdots R_{L_{M+N}}\right]$, the corresponding steady-state values of the buffer's output resistances $\left[R^*_{M+1} \cdots R^*_{M+N}\right]$ can be obtained. Once a set of desired output resistances, R^*_i, is given, corresponding steady-state energy, input resistance, and control input are e^*_i, r^*_i, and $u^*_i = 0$, respectively. r^*_i is found by solving (4.6) at the steady state, while e^*_i is fixed a priori based on the desired steady-state output voltage, v^*_{bi}. By defining the state deviations as $x_{i1} = e_i - e^*_i$ and $x_{i2} = r_i - r^*_i$, (4.6) can be rewritten as

$$\begin{cases} \dot{x}_{i1} = \frac{\zeta_i(x_{(M+1)2}+r^*_{M+1},\ldots,x_{i2}+r^*_i,\ldots,x_{(M+N)2}+r^*_N)^2}{x_{i2}+r^*_i} - \frac{2\left(x_{i1}+e^*_i\right)}{C}\frac{1}{R^*_i} \\ \dot{x}_{i2} = u_i \end{cases}, \quad i \in \mathcal{L}. \tag{4.7}$$

The last equation represents the state-space model of the DC microgrid with power buffers considering the nonlinear dynamics dictated by the distribution grid. The state variables are defined as the deviations with respect to a given steady-state equilibrium point. Such model is used in the subsequent to design distributed optimal control policies via ADP.

4.3 Distributed Assistive Control with Fixed Communication Topology

As in (4.5), each active load's input voltage is affected by its own input impedance, r_i, and by the impedances of all the other active loads. Let's define N_i as the set of all the indexes $k \in \mathcal{L}$ such that active load k is in the neighborhood of the ith active load. In this section, the neighbors set is inspired by the physical vicinity, i.e., for any value of the input resistances $(r_{M+1}, ..., r_i, ..., r_{M+N})$

$$\left|\frac{\partial\zeta_i(r_{M+1},...,r_{M+N})}{\partial r_j}\right| \ll \left|\frac{\partial\zeta_i(r_{M+1},...,r_{M+N})}{\partial r_k}\right| \tag{4.8}$$

for any $j \in \mathcal{L} \setminus N_i$ and for any $k \in N_i$. The communication topology used in the distributed optimal controller will reflect such physical vicinity, and, thus, is fixed a priori. The dependency of (4.5) on the non-neighbors can be neglected by setting the resistances of the non-neighbor loads to infinity, allowing the following approximated relation

$$v_i = \hat{\zeta}_i(r_i, \{r_j\}_{j\in N_i}), \quad i \in \mathcal{L}. \tag{4.9}$$

The dynamics in (4.7) can be thus expressed as follows

$$\begin{cases} \dot{x}_{i1} = \frac{\hat{\zeta}_i(x_{i2}+r^*_i,\{x_{j2}+r^*_j\}_{j\in N_i})^2}{x_{i2}+r^*_i} - \frac{2\left(x_{i1}+e^*_i\right)}{C}\frac{1}{R^*_i} \\ \dot{x}_{i2} = u_i \end{cases}, \quad i \in \mathcal{L}. \tag{4.10}$$

This can be written as

$$\dot{x}_i = f_i(\bar{x}_i) + bu_i, \quad i \in \mathcal{L}, \tag{4.11}$$

with $b = [0 \quad 1]^\intercal$, $x_i = [x_{i1} \quad x_{i2}]^\intercal$, $\bar{x}_i = (x_i^\intercal, \{x_j\}_{j\in N_i}^\intercal)$, and $f_i(\bar{x}_i)$ defined as

$$f_i(\bar{x}_i) = \frac{\hat{\zeta}_i(x_{i2} + r_i^*, \{x_{j2} + r_j^*\}_{j\in N_i})^2}{x_{i2} + r_i^*} - \frac{2\,(x_{i1} + e_i^*)}{C}\frac{1}{R_i^*} \quad i \in \mathcal{L}. \tag{4.12}$$

The dynamics of the entire DC microgrid then becomes

$$\underbrace{\begin{bmatrix} \dot{x}_{M+1} \\ \vdots \\ \dot{x}_{M+N} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} f_i(\bar{x}_{M+1}) \\ \vdots \\ f_i(\bar{x}_{M+N}) \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} b \cdots 0 \\ \vdots \ddots \vdots \\ 0 \cdots b \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_{M+1} \\ \vdots \\ u_{M+N} \end{bmatrix}}_{u}, \tag{4.13}$$

with the origin as an equilibrium, and $f(0) = 0$.

4.3.1 Assistive Control Problem as an Optimal Control Problem

The assistive control problem can be treated as finding an optimal feedback control law for (4.13) that minimizes a cost functional during the transient toward a given setpoint for any initial state. Suppose that the ith active load needs assistance; The cost functional is

$$J_i(x, u) = \int_0^\infty U_i(x, u)dt \quad i \in \mathcal{L}, \tag{4.14}$$

where x_0 is the initial state at $t = 0$. $U_i(\cdot, \cdot)$ is the corresponding *utility function*, defined as

$$U_i(x, u) = x_i^\intercal Q_{ii} x_i + \rho_i u_i^2 + \sum_{j\in N_i} \left(x_i^\intercal Q_{ij} x_j + x_j^\intercal Q_{jj}^{(i)} x_j + \rho_j^{(i)} u_j^2 \right), \tag{4.15}$$

where $Q_{ii} \in \mathbb{R}^{2\times 2}$, $Q_{jj}^{(i)} \in \mathbb{R}^{2\times 2}$, and $Q_{ij} \in \mathbb{R}^{2\times 2}$ are performance matrices weighting the state of active load i, the state of its neighbors, and their product, respectively. $\rho_i^{(i)}$ and $\rho_j^{(i)}$ are scalars weighting the active load's control input and that of its neighbors, respectively. The weighting terms ensure $U_i(x, u) \geq 0 \forall (x, u)$ and $U_i(0, 0) = 0$. Note that (4.15) can also be written as

$$U_i(x, u) = Q_i(x) + u^\intercal R_\rho^{(i)} u, \tag{4.16}$$

with $R_\rho^{(i)} = diag\left(\rho_{M+1}^{(i)}, \cdots, \rho_i, \cdots, \rho_{M+N}^{(i)}\right)$, and $Q_i(x) = x_i^\intercal Q_{ii} x_i + \sum_{j\in N_i} \left(x_i^\intercal Q_{ij} x_j + x_j^\intercal Q_{jj}^{(i)} x_j \right)$.

Intuitively, once the utility function is defined as in (4.16), optimization of (4.14) minimizes the states deviations, x_i and x_j, as well as the control effort u_i of each node involved in the assisting task. In order to guarantee assistance the weights are chosen appropriately: if node i requires more power on the load, the associated utility function, U_i, exhibits greater values of ρ_j than ρ_i, as well as greater weights in the matrix Q_{ii} than those in Q_{ij} and $Q_{jj}^{(i)}$. Note that this last matrix is introduced for converge purposes. By choosing the weights in such a way, the node's individual action is penalized, in favour of a collective action in which neighboring nodes provide support during transients. (4.14) represents a common sub-network objective, shared among the node i and its neighbors. This common objective switches based on the node subject to the load variation, thus, once each optimal control problem is solved, each node's control law switches based on the node that requires assistance.

4.3.2 ADP with Off-policy Learning Solves the Optimal Control Problem

The optimal control problem defined by the cost functional (4.14) and dynamics (4.13) can be solved via the ADP with off-policy learning algorithm discussed in Section 2.4.

In particular, let's consider the following system

$$\dot{x} = f(x) + B(u^{(0)} + e_L(t)), \tag{4.17}$$

where $u^{(0)}$ is a feedback policy that asymptotically stabilizes the system at the origin with a finite associated cost, and $e_L(t) : \mathbb{R} \to \mathbb{R}^N$ is a bounded *exploration noise* for the learning purposes, as seen in (2.26). For each iteration $k \geq 0$, let $u^{(k)'} = u^{(0)} - u^{(k)} + e_L$. Then, (4.17) can become

$$\dot{x} = f(x) + Bu^{(k)} + Bu^{(k)'}. \tag{4.18}$$

The time-derivative of the function $V^{(k)}(x)$, i.e., the value function at iteration k of the Policy Iteration algorithm 2.1, computed along the state trajectory of (4.18), is

$$\dot{V}^{(k)}(x) = \nabla V^{(k)\intercal}(x)\left[f(x) + B(u^{(k)} + u^{(k)'})\right] = -U_i(x, u^{(k)}) - 2\sum_{j\in N_i\cup\{i\}} u_j^{(k+1)}\rho_j^{(i)}u_j^{(k)'}, \tag{4.19}$$

where $u_j^{(k)}$ and $u_j^{(k)'}$ are the jth elements of $u^{(k)}$ and $u^{(k)'}$ vectors, respectively, while $\rho_i^{(i)} = \rho_i$. By following the procedure in Section 2.4, for each $k \geq 0$, the value function $V^{(k)}(x)$ and the control policies $u_j^{(k+1)}$, $j \in N_i \cup \{i\}$, can be approximated in a linear-in-parameters (LIP) fashion, i.e.,

$$\hat{V}^{(k)}(x) = \sum_{l=1}^{N_V} \omega_l^{(k)}\gamma_l(x) = \omega^{(k)\intercal}\Gamma(x), \tag{4.20}$$

$$\hat{u}_j^{(k+1)}(\bar{x}_j) = \sum_{l=1}^{N_U^j} \theta_{jl}^{(k)}\xi_{jl}(\bar{x}_j) = \theta_j^{(k)\intercal}\Xi_j(\bar{x}_j). \tag{4.21}$$

Where $\gamma_l(x) : \mathbb{R}^{2N} \to \mathbb{R}$, with $l = 1, ..., N_V$, and $\xi_{jl}(\bar{x}_j) : \mathbb{R}^{2(|N_i|+1)} \to \mathbb{R}$, with $l = 1, ..., N_U^j$ and $j \in \mathcal{L}$, are the sequences of linearly-independent smooth functions vanishing at the origin, while $\omega^{(k)}$ and $\theta_j^{(k)}$ are the constant row vectors of weights to be determined. Note that the approximating functions in (4.21) depend only on the current state and that of the neighbors. In this way, it is possible to find an approximated optimal control policy that stabilizes the system and, at the same time, is distributed according to the physical vicinity. Replacing $V^{(k)}$ and $u_j^{(k+1)}$ in (4.19) with their approximations, and by integrating both sides over any time interval $[t_n, t_{n+1}]$, the following integral reinforcement learning equation (see (2.31)) is obtained,

$$\begin{aligned}
\omega^{(k)\intercal}\underbrace{[\Gamma(x(t_{n+1})) - \Gamma(x(t_n))]}_{\Delta\Gamma(t_{n+1})} &= -\sum_{j\in N_i\cup\{i\}} \theta_j^{(k-1)\intercal}\underbrace{\left(\int_{t_n}^{t_{n+1}} \Xi_j(\bar{x}_j)\rho_j^{(i)}\Xi_j^{\intercal}(\bar{x}_j)dt\right)}_{\Phi_j(t_{n+1})}\theta_j^{(k-1)} \\
&- 2\sum_{j\in N_i\cup\{i\}} \theta_j^{(k)\intercal}\underbrace{\int_{t_n}^{t_{n+1}} \Xi_j(\bar{x}_j)\rho_j^{(i)}\left(u_j^{(0)} + e_{L_j}(t)\right)dt}_{\Psi_j(t_{n+1})} - \underbrace{\int_{t_n}^{t_{n+1}} Q_i(x)dt}_{Q_I(t_{n+1})} \\
&+ \sum_{j\in N_i\cup\{i\}} \theta_j^{(k)\intercal}\left(\int_{t_n}^{t_{n+1}} \Xi_j(\bar{x}_j)\rho_j^{(i)}\Xi_j^{\intercal}(\bar{x}_j)dt\right)\theta_j^{(k-1)} + \epsilon_n^{(k)}
\end{aligned} \tag{4.22}$$

where $\epsilon_n^{(k)}$ is the approximation error and $\{t_n\}_{n=1}^{N_L}$ is the increasing series of time intervals. As discussed in Chapter 2, starting from an initial stabilizable control policy u^0, sequences $\{\hat{V}^{(k)}\}_{k=0}^{\infty}$ and $\{\hat{u}^{(k+1)}\}_{k=0}^{\infty}$ converge to the optimal values. The weights $\omega^{(k)}$ and $\theta_j^{(k)}$ are obtained by minimizing $\sum_{n=0}^{N_L} \epsilon_n^{(k)^2}$ using a least-squares method.

The following algorithm implements the ADP procedure with off-policy learning, once the approximating functions, learning data, and utility function are given. Note that thanks to the properties of the LIP approximators, the data collecting phase is decoupled from the evaluation of (4.22). The computational efforts are reduced since the same collected data solves several optimal control problems with different utility functions.

Algorithm 4.1 ADP PI Algorithm with off-policy learning

Inputs:

- Utility function parameters, i.e., Q_{ii}, Q_{ij}, $Q_{jj}^{(i)}$, ρ_i, $\rho_j^{(i)}$, with $j \in N_i$;
- Approximating functions $\Gamma(x)$ and $\Xi_j(\bar{x}_j)$, with $j \in N_i \cup \{i\}$;
- Initial stable controller weights $\theta_j^{(0)}$ for any power buffer;
- System's collected data, i.e., x, $u^{(0)} + e_L(t)$, and $\Delta\Gamma(t_n), n = 1, ..., N_L$, recorded from (4.17);
- A stopping threshold δ.

Outputs: approximated optimal weights $\hat{\omega}$ and $\hat{\theta}_j, j \in N_i \cup \{i\}$.

1. **Initialization:** Set the initial iteration number as $k = 1$.
2. **Data Evaluation:** By using the previously collected data, evaluate $Q_I(t_n)$, $\Psi_j(t_n)$, and $\Phi_j(t_n)$, with $j \in N_i \cup \{i\}$, and $n = 1, ..., N_L$.
3. **Policy Improvement:** Evaluate the following matrices

$$X = \begin{bmatrix} \Delta\Gamma(t_1) & 2\left(\Psi_{j1}(t_1) - \Phi_{j1}(t_1)\theta_{j1}^{(k-1)}\right)^{\intercal} & \cdots & 2\left(\Psi_{jz}(t_1) - \Phi_{jz}(t_1)\theta_{jz}^{(k-1)}\right)^{\intercal} \\ \vdots & \vdots & \vdots & \\ \Delta\Gamma(t_{N_L}) & 2\left(\Psi_{j1}(t_{N_L}) - \Phi_{j1}(t_{N_L})\theta_{j1}^{(k-1)}\right)^{\intercal} & \cdots & 2\left(\Psi_{jz}(t_{N_L}) - \Phi_{jz}(t_{N_L})\theta_{jz}^{(k-1)}\right)^{\intercal} \end{bmatrix}$$

$$B_\Phi = -\begin{bmatrix} Q_I(t_1) + \sum\limits_{j \in N_i \cup \{i\}} \theta_j^{(k-1)\intercal} \Phi_j(t_1)\theta_j^{(k-1)} \\ \vdots \\ Q_I(t_{N_L}) + \sum\limits_{j \in N_i \cup \{i\}} \theta_j^{(k-1)\intercal} \Phi_j(t_{N_L})\theta_j^{(k-1)} \end{bmatrix}$$

with $j1, ..., jz \in N_i \cup \{i\}$. Then find unknown weights by solving the following

$$X \begin{bmatrix} \omega^{(k)\intercal} & \theta_{j1}^{(k)\intercal} & \cdots & \theta_{jz}^{(k)\intercal} \end{bmatrix}^{\intercal} = B_\Phi.$$

4. **Off Policy Iteration:** If $\left|\left|\omega^{(k)} - \omega^{(k-1)}\right|\right| \geq \delta$ set $k = k + 1$ and repeat Step 3. Otherwise, stop and return the approximated optimal value function and control policy, i.e., $\hat{\omega} = \omega^{(k)}$ and $\hat{\theta}_j = \theta_j^{(k)}$, with $j \in N_i \cup \{i\}$.

4.3.3 Learning Phase

Algorithm 4.1 is exploited to obtain a set of near-optimal policies with respect to all the active loads, considering several desired setpoints. The obtained weight sets are then aggregated into look-up tables to compose a control scheme able to provide a near-optimal policy working in all scenarios. Algorithm 4.2 summarizes the learning procedure. For a set of loads for the ith active load, $P_{R_{L_i}} = \{R_{L_i}^{*(1)}, .., R_{L_i}^{*(S_i)}\}$, the corresponding set $P_{R_i} = \{R_i^{*(1)}, .., R_i^{*(S_i)}\}$ can be found. A learning grid of different loads is defined as $P_{R_{M+1}} \times ... \times P_{R_{M+N}}$. For each element of this learning grid, a set of optimal control problems is defined corresponding to each active load that needs assistance for that specific setpoint. Then, given an N-tuple $(\bar{R}^*_{M+1}, ..., \bar{R}^*_{M+N}) \in P_{R_{M+1}}$, corresponding input impedances $(\bar{r}^*_{M+1}, ..., \bar{r}^*_{M+N})$ are found by solving (4.10) in the steady state. The input impedances are stored in a map, $M_r(\bar{R}^*_{M+1}, ..., \bar{R}^*_{M+N})$, to compute the states of each active load fed to the controller. Once all the reference values are given, the data collection phase can be performed. For each setpoint, N corresponding optimal control problems are solved by means of Algorithm 4.1. The obtained control weights for the ith problem are stored in a map, $M^i_{\theta_j}(\bar{R}^*_{M+1}, ..., \bar{R}^*_{M+N})$, for each $j \in N_i \cup \{i\}$. This map defines the control policy of active load j that is triggered through a load change in the active load i.

Algorithm 4.2 Assistive Control Learning Procedure

Inputs:

- Buffer's output resistances set for each active load $P_{R_i} = \{R_i^{*(1)}, .., R_i^{*(S_i)}\}$;
- Utility function parameters, i.e., Q_{ii}, Q_{ij}, $Q^{(i)}_{jj}$, ρ_i, $\rho^{(i)}_j$, for any $i \in \mathcal{L}$ and $j \in N_i$;
- Approximating functions and initial stable controller weights $\theta^{(0)}_j$ for any power buffer, as in Algorithm 4.1.

Outputs:

- Input impedance references map $M_r(R_{M+1}, ..., R_{M+N})$;
- Near-optimal control policies map $M^i_{\theta_j}(R_{M+1}, ..., R_{M+N})$, with $i \in \mathcal{L}$ and $j \in N_i \cup \{i\}$.

1. **for** each $(\bar{R}^*_{M+1}, ..., \bar{R}^*_{M+N}) \in P_{R_{M+1}} \times ... \times P_{R_{M+N}}$ **do**
2. Find corresponding $(\bar{r}^*_{M+1}, ..., \bar{r}^*_{M+N})$ by solving (4.10) in the steady state, with $R^*_i = \bar{R}^*_i$, and set
$$M_r(\bar{R}^*_{M+1}, ..., \bar{R}^*_{M+N}) = (\bar{r}^*_{M+1}, ..., \bar{r}^*_{M+N}).$$
3. Define system (4.13) by setting reference values found in Step 2, and collect corresponding learning data using the initial controller.
4. **for** each active load i **do**
5. Solve the optimal control problem using Algorithm 4.1 with learning data from Step 3 and terms Q_{ii}, $\rho^{(i)}_i$, Q_{ij}, $Q^{(i)}_{jj}$ and $\rho^{(i)}_j$, with $j \in N_i \cup \{i\}$, and set
$$M^i_{\theta_j}(\bar{R}^*_{M+1}, ..., \bar{R}^*_{M+N}) = \hat{\theta}_j.$$
6. **end for**
7. **end for**

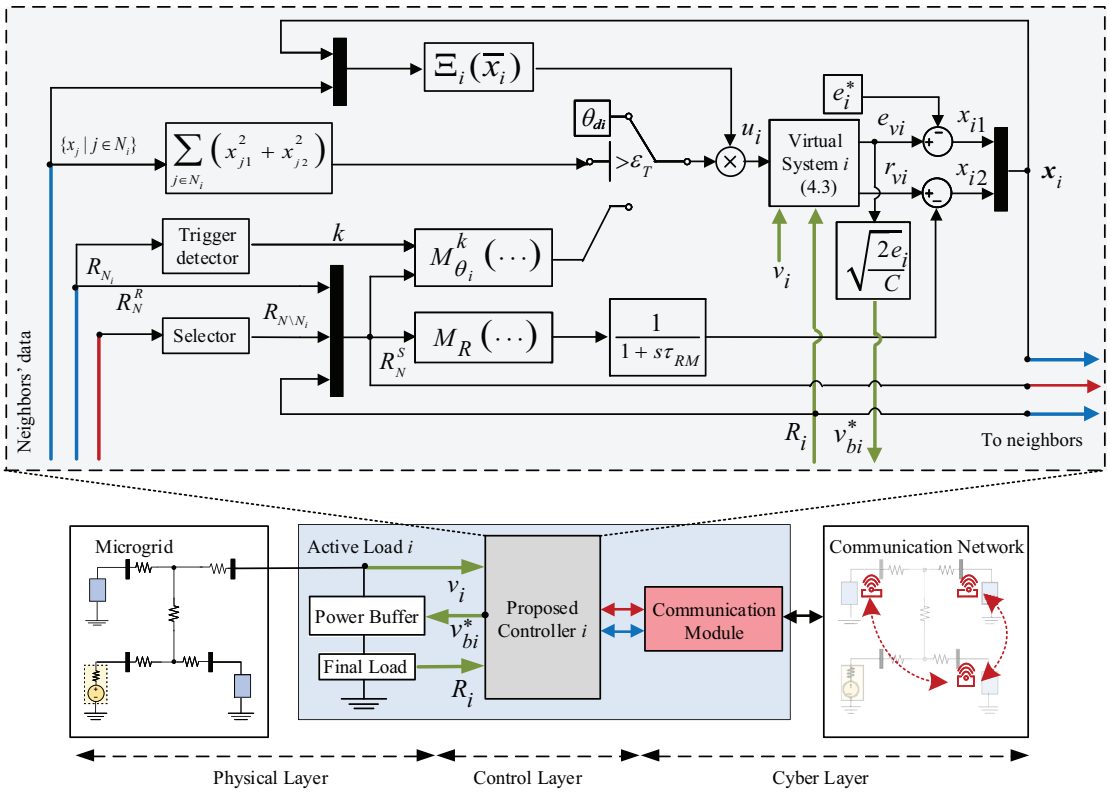

Figure 4.3: Proposed control scheme. Green, blue, and red lines refer to local data, incoming/outgoing real-time data, and incoming/outgoing high-latency data.

4.3.4 Assistive Control Scheme

As the first control objective, the buffer's output voltage should be fixed, in the steady state, on the rated value of v_{bi}^*, which corresponds to e_i^* as in (4.2). As the second objective, the input impedance profile varies during transients according to the assistive control policy. The proposed scheme employs the voltage tracker embedded into the power buffer to handle both objectives. The assistive policy acts on the software implementation of system (4.3), defined as the *virtual system* in Fig. 4.3, and is connected to the physical buffer through the input voltage, v_i. e_{vi} and r_{vi} in Fig. 4.3 denote the states of the virtual system synced with the physical one through the input voltage, v_i. The real-time controller uses the states of the virtual system in its feedback policy. In particular, the real-time value of r_{vi} is obtained by integrating the control input, u_i. The controller drives the input impedance of the virtual system, providing a desired energy profile translated into the reference of the voltage tracker of the power buffer.

Each power buffer sends to its neighbors its own state and output resistance, i.e., x_i and R_i, respectively. The distributed assistive control policy is triggered when an active load detects a change in its neighborhood. Otherwise, it uses a default local stabilizing controller $u_{di}(x_i) = \theta_{di}\Xi_i(\bar{x}_i)$; Where θ_{di} is chosen such that u_{di} depends only on local states. After each transient, i.e., when $\sum_{j\in N_i}(x_{j1}^2 + x_{j2}^2)$ is lower than a defined threshold ϵ_T, the control weights switch to the default ones. Thus, the communication module is used only during the assistive task. This makes the proposed method suitable for energy-constrained devices (e.g., IoT devices),

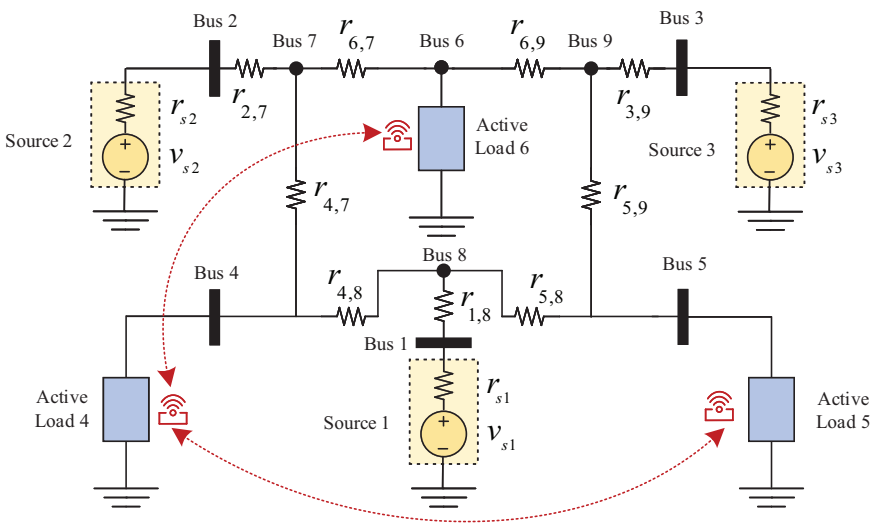

Figure 4.4: Considered DC microgrid structure.

and keeps the system stable in case of communication fails.

The reference and weights map in Algorithm 4.2 are queried by the complete N-tuple $(R^*_{M+1}, ..., R^*_{M+N})$. To correctly query the maps, active load i has to know its resistance, the set of neighbors, and the set of non-neighbors resistances, i.e., R_i, $R_{N_i} = \{R_j\}_{j \in N_i}$, and $R_{N \setminus N_i} = \{R_j\}_{j \in N \setminus N_i}$, respectively. Thus, the control mechanism is enhanced with a communication protocol to broadcast each routing active load's vector $R^S_N = (R_i, R_{N_i}, R_{N \setminus N_i})$ to its neighbors. This protocol ensures consensus among active loads if the communication graph features a spanning tree [156]. Once a load change occurs, the neighbors state data is sent in real time, while the information R^S_N is sent with a higher latency. The maximum latency has to be lower than the minimum rate of load change for each active load. Once the ith active load detects a change in R_{N_i}, the non-neighbors resistances are selected from R^R_N, which is the received counterpart of R^S_N. Hence, the active load can correctly query both the weights and reference maps.

Assuming that the learning procedure has been properly conducted, and given the stabilizing properties of the default policy, a switch between asymptotically-stable controllers occurs once the transient effects are dissipated. The output of the reference map is filtered to avoid states jump and preserve system stability during the switching phase [157]. This filter's time constant, τ_{RM}, is chosen faster than the communication sampling time.

4.3.5 CHIL Validation

a. System Setup

The proposed control scheme is verified on a 48V DC microgrid, with its structure shown in Fig. 4.4. Line resistances are set as follows

$$\begin{aligned} &r_{4,8} = 0.2\Omega, r_{1,8} = 0.35\Omega, r_{6,7} = 0.6\Omega, \\ &r_{5,8} = r_{5,9} = r_{6,9} = 0.3\Omega, \\ &r_{4,7} = r_{2,7} = r_{3,9} = 0.5\Omega, \end{aligned} \tag{4.23}$$

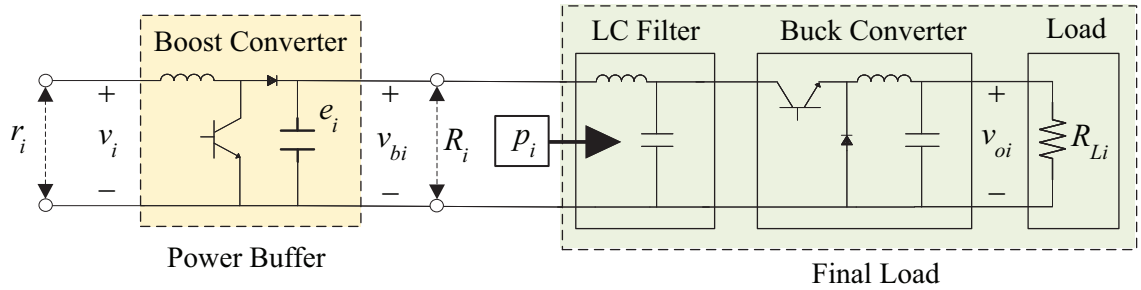

Figure 4.5: Power buffer and final load architecture.

Table 4.1: Power Buffer and Final Load Parameters

Power Buffer		Final Load	
Parameter	Value	Parameter	Value
Converter input inductor	4.00mH	LC filter inductor	300μH
Converter output capacitor	4.4mF	LC filter capacitor	2.2mF
Converter input inductor ESR	520mΩ	Converter output inductor	2.65mH
Converter output capacitor ESR	25mΩ	Converter output capacitor	2.2mF
Switching frequency	60kHz	LC filter inductor ESR	100mΩ
Proportional gain (voltage controller)	1	LC filter capacitor ESR	50mΩ
Integral gain (voltage controller)	3.5	Converter output inductor ESR	520mΩ
Rated output voltage	100V	Converter output capacitor ESR	50mΩ
		Switching frequency	60kHz
		Proportional gain	0.09
		Integral gain	1.08
		Output voltage set point	48V

while every DC source is modeled as a series connection of a $50V$ ideal voltage source and a 0.1Ω resistor. Each active load consists of a power buffer (boost converter) and a final load composed of a buck converter with an LC filter interposed in between, as shown in Fig. 4.5. Power buffer and final load parameters are reported in Table 4.1. The boost converter features a fast voltage tracker to follow the voltage profile defined by the assistive control scheme in Fig. 4.3. Its rated output voltage is $v_{bi}^* = 100V$. The fast voltage regulator of the buck converter is regulated at 48V. Both voltage trackers employ Proportional-Integral (PI) controllers.

The relationship between the control input, u_i, and the switching state of the solid-state switch of the boost converter can be derived as follows. Given the initial value of the stored energy, e_{i0}, input impedance r_{i0}, and load value R_i, the control input profile, u_i, is translated into the energy profile, e_{vi}, by integrating the equations of the virtual system as in Fig. 4.3

$$e_{vi}(t) = e^{-\frac{2}{C}\frac{1}{R_i}} \left(e_{i0} + \int_0^t \frac{e^{\frac{2}{C}\frac{1}{R_i}\tau} v_i(\tau)^2}{r_{i0} + \int_0^\tau u_i(\zeta) d\zeta} d\tau \right), \tag{4.24}$$

where v_i is the measured input voltage of power buffer i. Using (4.2), e_{vi} is translated into the reference of the fast voltage tracker for the boost converter, i.e., $v_{bi}^*(t) = \sqrt{(2/C)e_{vi}(t)}$. The output of the Proportional-Integral controller of the i-th boost converter is denoted by y_i^{PI}, while its input is the error between the reference voltage, $v_{bi}^*(t)$, and the measured output voltage,

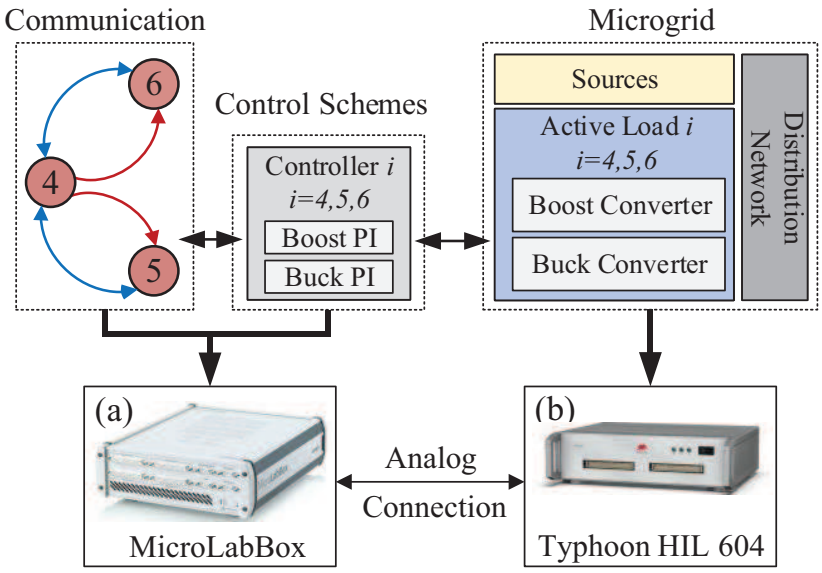

Figure 4.6: Controller/Hardware-in-the-loop setup: (a) dSPACE MicroLabBox Controller (handles the control and communication routines); (b) Typhoon HIL 604 (emulates the physical components of the underlying microgrid).

v_{bi}. y_i^{PI} is used along with the measured input current, i_i, in an hysteresis-band controller to determine the switching state of the solid-state device,

$$d_i(t) = \begin{cases} 1 & if \quad y_i^{PI} - i_i > hb_i \\ 0 & if \quad y_i^{PI} - i_i < -hb_i \end{cases} , \tag{4.25}$$

where hb_i is the hysteresis band herein set as 0.2. The switch status is kept constant for values between the thresholds.

The physical microgrid is emulated on a Typhoon HIL 604, and the communication network and the control scheme run on a dSPACE MicroLabBox controller board, as shown in Fig. 4.6. The sampling times of the controller and the communication module are $0.1ms$ and $1ms$, respectively. The time constant of the filter placed after the resistances map is $\tau_{RM} = 0.2ms$.

b. Learning Stage

The assistive control scheme requires a learning phase via Algorithm 4.2. According to Fig. 4.4, neighborhood sets are $N_4 = \{5, 6\}$, $N_5 = 4$, and $N_6 = 4$. The output load for each active load varies from 10Ω to 100Ω, in steps of 10Ω. Mixed linear-independent polynomial terms, up to 4th degree, are used as approximating functions, with a corresponding $N_V = 166$, $N_U^4 = 83$, and $N_U^5 = N_U^6 = 34$. The structure of approximating functions, for both critic and

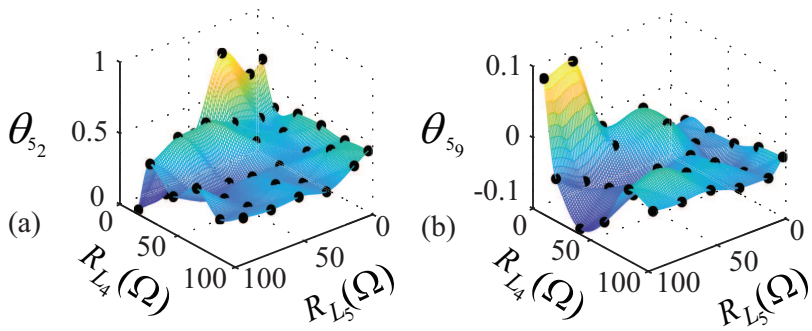

Figure 4.7: Two policy weights of the active load 5, when the active load 4 is in need: (a) weights for approximating function x_{5_1}; (b) Weights for approximating function $x_{4_1}x_{5_1}$.

actor networks, is

$$\begin{cases} \hat{V}^{(k)}(x) = \sum\limits_{\substack{l=1,\dots,166 \\ i_1,\dots,i_6\geq 0 \\ 2\leq i_1+\dots+i_6\leq 4}} c_l x_{4_1}^{i_1} x_{4_2}^{i_2} x_{5_1}^{i_3} x_{5_2}^{i_4} x_{6_1}^{i_5} x_{6_2}^{i_6} \\ \hat{u}_4^{(k+1)}(\bar{x}_4) = \sum\limits_{\substack{l=1,\dots,83 \\ i_1,\dots,i_6\geq 0 \\ 1\leq i_1+\dots+i_6\leq 3}} w_{4_{k_l}} x_{4_1}^{i_1} x_{4_2}^{i_2} x_{5_1}^{i_3} x_{5_2}^{i_4} x_{6_1}^{i_5} x_{6_2}^{i_6} \\ \hat{u}_5^{(k+1)}(\bar{x}_5) = \sum\limits_{\substack{l=1,\dots,34 \\ i_1,\dots,i_4\geq 0 \\ 1\leq i_1+\dots+i_4\leq 3}} w_{5_{k_l}} x_{4_1}^{i_1} x_{4_2}^{i_2} x_{5_1}^{i_3} x_{5_2}^{i_4} \\ \hat{u}_6^{(k+1)}(\bar{x}_6) = \sum\limits_{\substack{l=1,\dots,34 \\ i_1,\dots,i_4\geq 0 \\ 1\leq i_1+\dots+i_4\leq 3}} w_{6_{k_l}} x_{4_1}^{i_1} x_{4_2}^{i_2} x_{6_1}^{i_3} x_{6_2}^{i_4} \end{cases} \tag{4.26}$$

The learning sequence $\{t_n\}$, with $N_L = 10000$ intervals of $10ms$ and 3 filtered white noises, are used as exploration signals. For each active load, the initial stabilizing controller is $\hat{u}_j^{(0)} = 2x_{j1}, j = 4, 5, 6$. A trial and error approach finds the initial controller whose stability has been checked through simulations conducted over several loading scenarios. The same steady-state stored energy were considered for all the buffers. The weighting terms are set as

$$\begin{cases} Q_{44} = Q_{55} = Q_{66} = diag(8, 8), \\ Q_{44}^{(5)} = Q_{44}^{(6)} = Q_{55}^{(4)} = Q_{55}^{(6)} = diag(1, 1), \\ Q_{45} = diag(-2, 0), \\ Q_{46} = diag(-1, 0), \\ Q_{54} = Q_{64} = diag(-5, 0), \\ \rho_4 = \rho_5 = \rho_6 = 1, \rho_5^{(4)} = \rho_6^{(4)} = \rho_4^{(5)} = \rho_4^{(6)} = 0.1. \end{cases} \tag{4.27}$$

Once the learning phase is complete, the near-optimal control policy maps are interpolated to obtain different control weights surfaces for each active load with respect to each neighbor in need. As an example, Fig. 4.7 shows two surfaces actuated by the active load 5 and triggered when the active load 4 needs assistance. Note that the weights depend on the desired setpoint (here, $R_{L_6} = 70\Omega$).

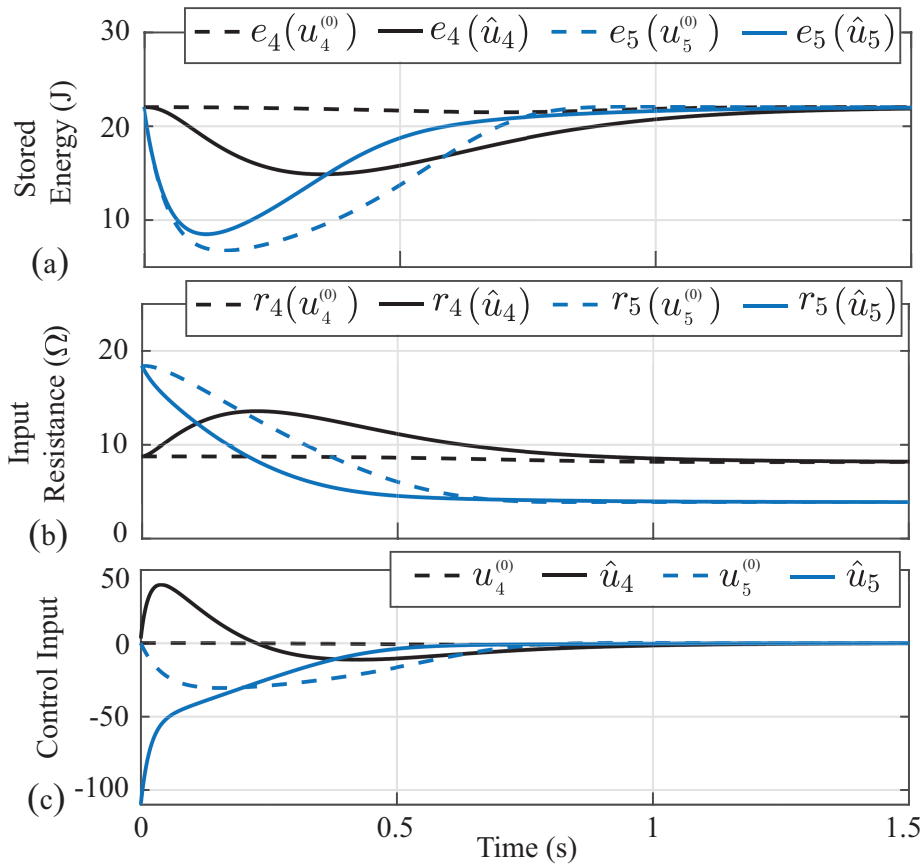

Figure 4.8: Learning results when power buffer 5 is in need: (a) stored energies for initial and near-optimal controllers; (b) Input resistances for initial and near optimal controllers; (c) initial and near-optimal control inputs.

Example studies from Algorithm 4.1 in Fig. 4.8 show how the near-optimal control policy provides assistance among neighboring power buffers. Using formulation (4.13) for the underlying DC microgrid, a single control policy, $\hat{u}$, was obtained to assist power buffer 5 during transients with the same weighting terms described above. In this example, R_5 changes from 80Ω to 10Ω at $t = 0$, while R_4 and R_6 are set as 40Ω and 30Ω, respectively. Figures 4.8(a) and 4.8(b), respectively, show the trajectories of e_4, e_5 and r_4, r_5 both with the initial control policies $u_4^{(0)}$, $u_5^{(0)}$, and with the near-optimal control policies $\hat{u}_4$, $\hat{u}_5$. These control policies are compared in Fig. 4.8(c). The initial control policy of power buffer 4 did not provide assistance to the power buffer 5, while the near-optimal control policy of power buffer 4 uses its stored energy to help power buffer 5 during transients, reducing both the energy and input impedance variations for power buffer 5.

The controller stability depends on the approximation domain of the employed neural networks in the learning stage [65]. The exploration signal allows the system states to span the region for the considered loading scenarios. Thus, the near-optimal control policy becomes stable, providing an approximated optimal value function $\hat{V}$, that acts as a Lyapunov function, as shown in the left and central parts of Fig. 4.9(a). The performances of the two control poli-

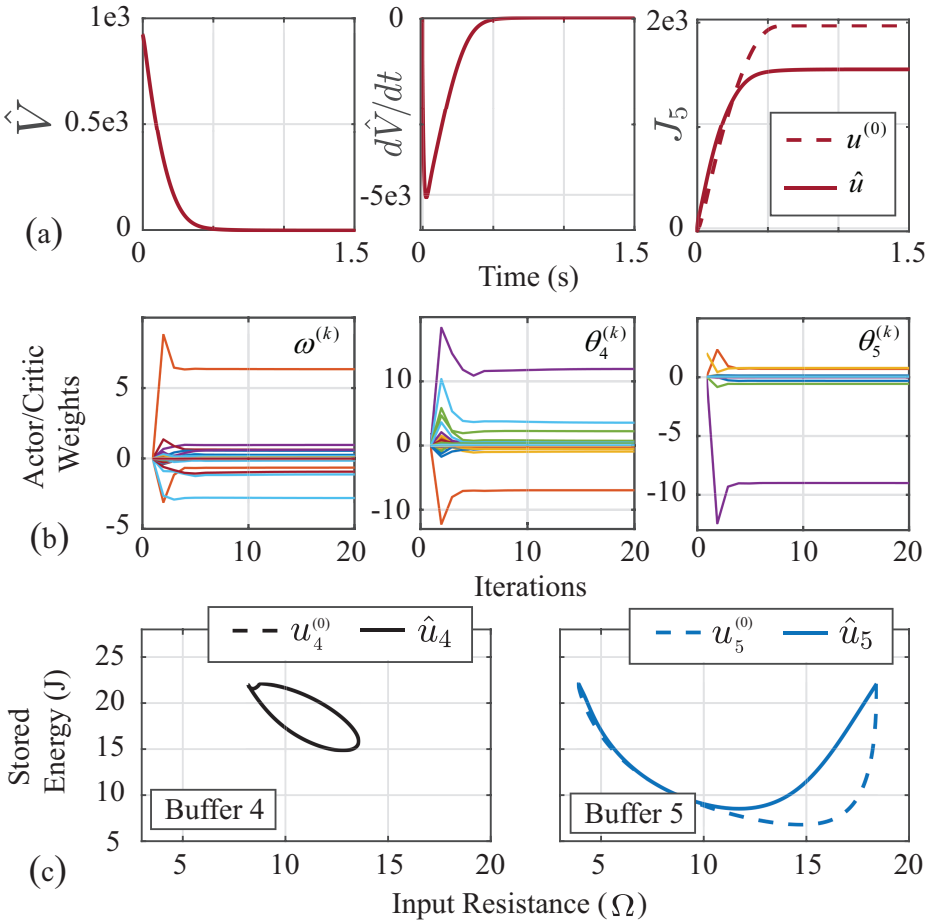

Figure 4.9: Learning results when power buffer 5 is in need: (a) Time trajectories of the learned value function (left), derivative of the learned value function (center), and performance comparison (right); (b) Weights convergence for the critic network (left), actor network of power buffer 4 (center), and actor network of power buffer 5 (right); (f) Energy-impedance trajectories for the initial and near-optimal control policies.

cies, in minimizing the cost function, is shown in the right part of Fig. 4.9(a). Clearly, the near-optimal controller, $\hat{u}$, provides a lower value for the shared objective function, J_5. The weights convergence for this scenario is depicted in Fig. 4.9(b). Finally, Fig. 4.9(c) shows the energy-impedance trajectories for the two power buffers, with both the initial controller and the near-optimal one. As seen, the initial control policy for power buffer 4 doesn't change its stored energy, while its approximated optimal policy assists power buffer 5, by using the buffering capabilities of power buffer 4.

c. Deactivated Power Buffers

Figures 4.10 and 4.11 show the system performance when power buffers are inactive. The initial loads of active loads 4, 5 and 6 are 80Ω, 100Ω and 70Ω, respectively. The load attached to the power buffer 5 changes to 20Ω at $t = 2s$. The load attached to the power buffer 4 goes to 15Ω at $t = 9s$. Loads 4 and 5 regain their original values at $t = 15s$ and $t = 25s$, respectively.

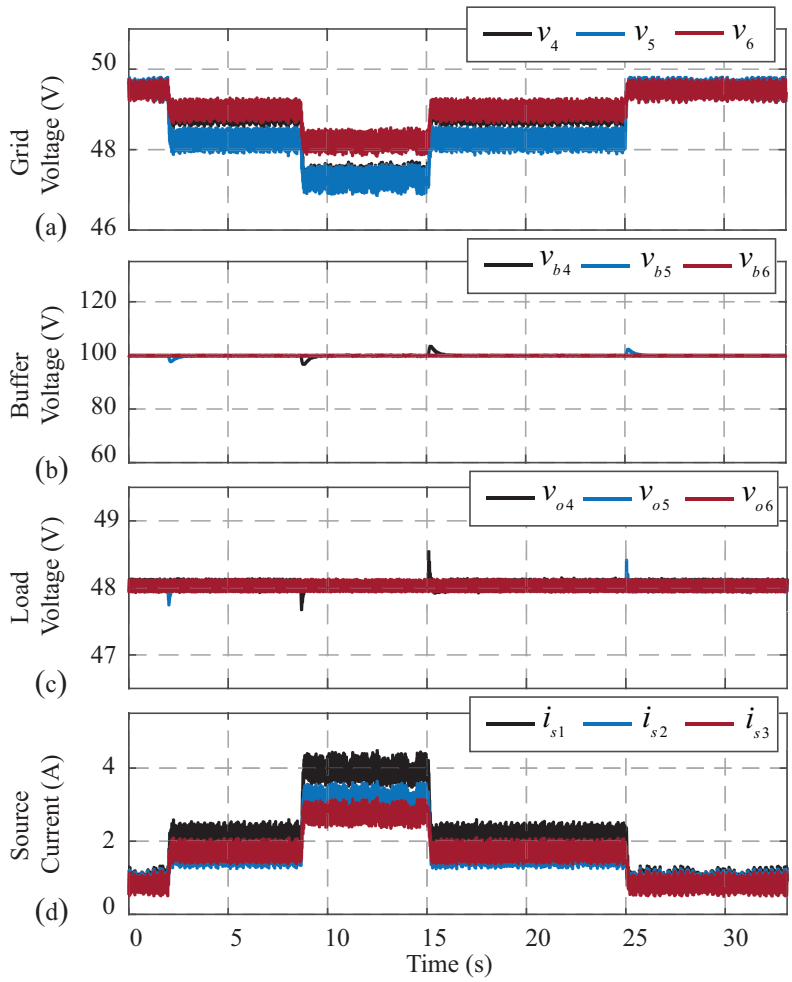

Figure 4.10: Microgrid performance in response to two load changes at terminal 5 and terminal 6 with deactivated power buffers: (a) Distribution bus voltages observed at the load terminals; (b) Output voltage of the power buffers; (c) Output voltage across the resistive loads; (d) Source currents.

Bus voltages and source currents exhibit step-change behaviors in Figs. 4.10(a) and 4.10(d), respectively. Note that when slow or stochastic (renewable) sources are present, such abrupt demands on the source currents are highly undesired. The energy-impedance trajectories of active loads are shown in Fig. 4.11(d). The trajectories corresponding to the first, second, third, and fourth load changes are represented by red, green, orange, and violet lines, respectively. The operating points of buffers 4 and 5 form an almost straight line, while buffer 6 doesn't show any change.

d. Activated Power Buffers with Communication Delays

The proposed control scheme is activated, and the communication network links the neighboring active loads. Some studies report IEEE 802.11 (WiFi) or Bluetooth Low Energy (BLE) as communication protocols mostly suited for low-power IoT devices [158]. During the assistive task, a data packet with 3 doubles (R_{L_i}, x_{i1}, and x_{i2}) is communicated, which would require

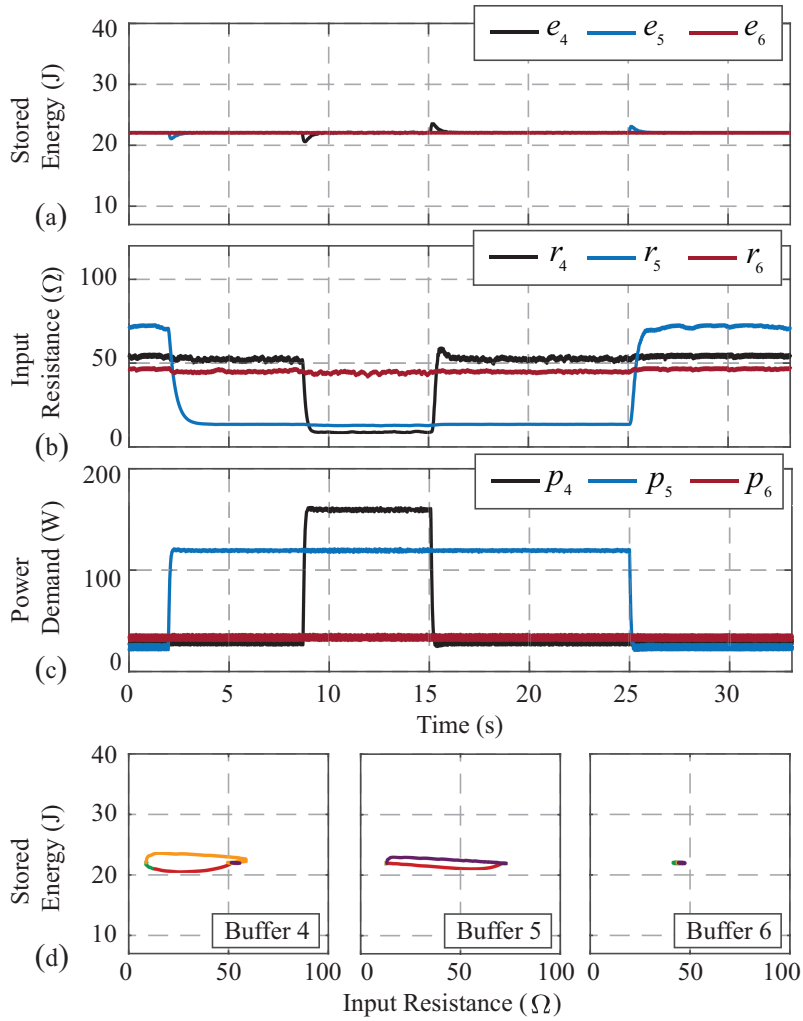

Figure 4.11: Microgrid performance in response to two load changes at terminal 5 and terminal 6 with deactivated power buffers: (a) Stored energy in power buffers; (b) Input impedance of the power buffers; (c) Output of the active loads; (d) Energy-impedance trajectories of the power buffers.

a single link capacity of 192 kbps. Maximum data rates for WiFi and BLE are 54Mbps and 1 Mbps, respectively [159]. Thus, BLE is suitable for microgrids with up to 5 neighbors for each active load; Otherwise, WiFi is preferred. For both protocols, the maximum transport delay is less than $100ms$ [160, 161]. Communication delays of $125ms$, $120ms$, and $130ms$ are introduced in the links between active loads 4 and 5, active loads 5 and 4, and active loads 4 and 6, respectively.

For $0 < t < 2s$, all the power buffers run a default control law, the same as the $u_i^{(0)}$ in Section b.. At $t = 2s$, the active load 5 changes from 100Ω to 20Ω, while active loads 4 and 6 stay at 80Ω and 70Ω, respectively. The active load 4 receives the load-change signal after $120ms$, triggering the assistive control law by querying the references map and the weight surfaces. Once the transient is over, active load 4 switches to the default control law and updates the active load 6. Thus, the active load 6 can correctly query the references map at the next event. At $t = 9s$, the active load 4 changes to 15Ω, triggering its own near-optimal policy. After

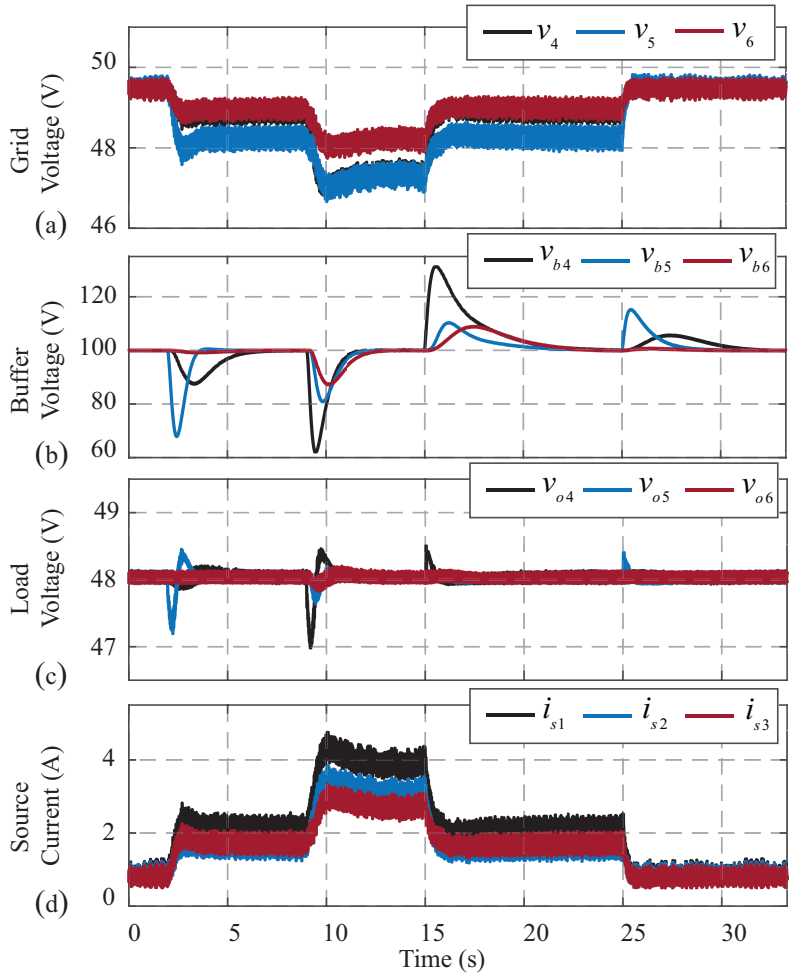

Figure 4.12: Microgrid performance in response to two load changes at terminal 5 and terminal 6 with activated power buffers: (a) Distribution bus voltages observed at the load terminals; (b) Output voltage of the power buffers; (c) Output voltage across the resistive loads; (d) Source currents.

$125ms$ and $130ms$, respectively, active loads 5 and 6 receive the information and trigger their control policies to assist the active load 4. At $t = 15s$ and $t = 25s$, active loads 4 and 5 are changed back to their initial values, respectively.

Validation results are showed in Fig. 4.12 and Fig. 4.13. After the first load change event, the active load 4 is only supporting the active load 5. The second event requires that both active loads 5 and 6 help smooth the transients. As shown in Fig. 4.13(a), once active load 5 abruptly changes, the stored energy of the active load 4 changes according to the assistive control law. The same happens to the stored energies of active loads 5 and 6, after the active load 4 changes. Red curves in Fig. 4.13(d) show impedance-energy trajectories during the first event, the green curves show those trajectories after the second event. In both cases, assistance is provided by dropping the stored energy and increasing the input impedance of the corresponding buffer. Orange and violet curves in Fig. 4.13(d) refer to the third and fourth load change events, respectively. Energy-impedance trajectories of buffers 4 and 5 go back to their initial points. Violet trajectory of buffer 4, and orange trajectories of buffers 5 and 6, denote how the stored

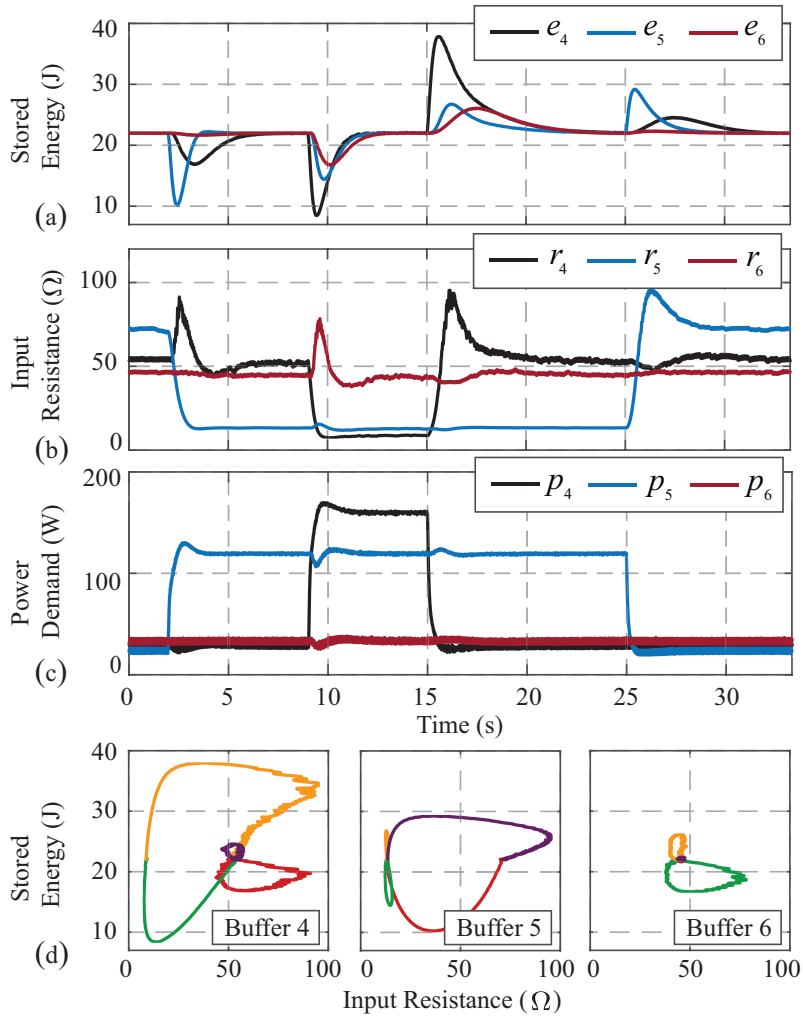

Figure 4.13: Microgrid performance in response to two load changes at terminal 5 and terminal 6 with activated power buffers: (a) Stored energy in power buffers; (b) Input impedance of the power buffers; (c) Output of the active loads; (d) Energy-impedance trajectories of the power buffers.

energy and impedance exhibit smaller variations. This asymmetric behavior is due to the non-linearity of the control law as well as the choice of weighting terms. As shown in Fig. 4.12(a), Fig. 4.12(d), and Fig. 4.13(c), the group action of power buffers smooth, respectively, the input bus voltages, source currents, and power demands.

A comparison with the distributed algorithm presented in [130] is shown in Fig. 4.14. At $t = 0.7s$, buffer 4 observes an abrupt change of its load from 80Ω to 15Ω. Using ADP, the energy stored in the buffer 4 recovers faster, as seen in Fig. 4.14(a). The energies stored in buffers 5 and 6 show higher deviation with comparable (active load 6) or slower (active load 5) settling times. This shows how the proposed method penalizes the individual action of the active load 4, enhancing the collective assistance provided by the active loads 5 and 6. Power demands are kept smooth, with a smaller initial derivative, as seen in Fig. 4.14(b). A faster dynamic response could be attained by adjusting the control gains in [130]. Therein, the controller design was based on a small-signal approximation of power buffers, making the

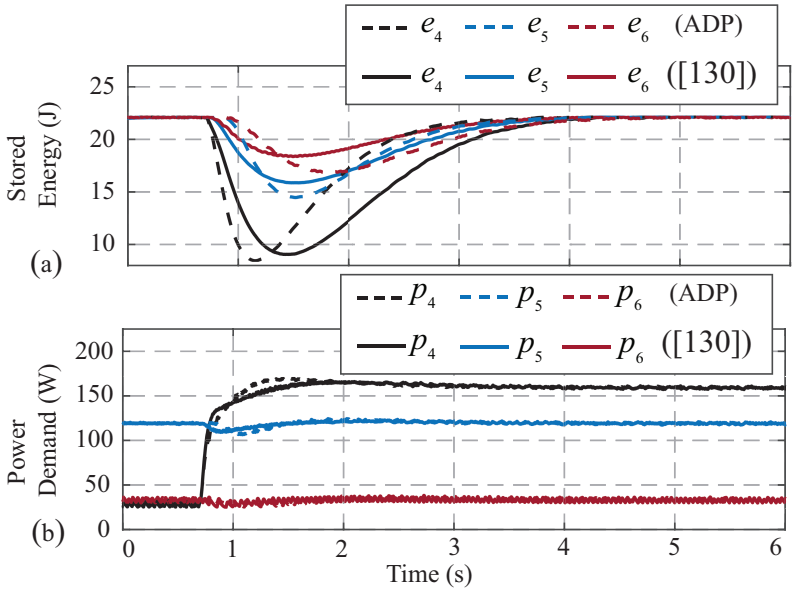

Figure 4.14: Proposed controller performances against the linear controller in [130].

controller valid only for a single operating point without guaranteeing its performance for larger load variations. By contrast, the method proposed here is based on a nonlinear formulation of the microgrid in (4.13). So long as the learning phase spans a sufficiently-large loading space, and the PE condition is valid, controller stability is guaranteed for higher deviations. Finally, individual controllers are designed through Algorithm 3 for each load variation. The optimal control formulation guarantees semi-optimal performance in every learned scenario.

4.4 Distributed Assistive Control with Sparsity Promoting

Section 4.3 considered a communication graph defined a priori and dictated by the physical vicinity of the power buffers, i.e., a communication structure reflecting the underlying dependency between each active load's input voltage and the other active loads input impedances, as in (4.8). However, there is no guarantee that these structures, i.e., the configuration of the underlying physical dependency and the topology of the communication graph, are optimal with regards to the defined control objectives.

Alternatively, the communication topology can be considered a free parameter subject to optimization. In particular, it is desired to minimize the number of active communication links with a limited impact on the closed loop cost functional defined according to the optimal control theory. The co-optimization of the two quantities provides a good trade-off between closed loop performances and computational costs associated to the communication. Clearly, if the nonlinear dynamics is considered, better performances are obtained, as seen in the Section 4.3.

However, the approach developed in Section 4.3 strictly depends on the knowledge of the desired overall target operating point. In fact, different optimal control problems are solved in Algorithm 4.2 according to a defined mesh of desired loads, for any power buffer. This is necessary since each different operating point defines a different nonlinear system, and, thus, a different optimal controller. Therefore, such approach is clearly not suitable when the com-

munication topology is considered as a free parameter: it would be necessary to solve a co-optimization problem for each set-point obtaining different communication topologies for each problem. Therefore, in this section a second-order linearization around the half-load operations of the DC microgrid with power buffers is considered. In this way, the communication topology no longer depends on the target set-point, and, at the same time, a good trade-off between linear and fully nonlinear formulations is obtained. Experimental results show the effectiveness of the obtained control structure on loading scenarios different from the one employed for the second-order linearization.

To start with, let's derive the second order linearization of the DC microgrid with power buffers. Let's consider the dynamics in (4.6), and, as done in 4.3, once a set of desired output resistances, R_i^*, is given, corresponding steady-state energy, input resistance, and control input are e_i^*, r_i^*, and $u_i^* = 0$, respectively. r_i^* is found by solving (4.6) at the steady state. The second-order approximation of (4.6), for each $i \in \mathcal{L}$, around an equilibrium point is

$$\begin{cases} \dot{x}_{i_1} = \sum\limits_{j=M+1}^{M+N} \left(\frac{\partial}{\partial r_j} \frac{\zeta_i(r)^2}{r_i} \bigg|_{r^*} x_{j_2} + \frac{1}{2} \frac{\partial^2}{\partial r_j^2} \frac{\zeta_i(r)^2}{r_i} \bigg|_{r^*} x_{j_2}^2 \right) - \frac{2}{CR_i^*} x_{i_1} \\ \dot{x}_{i_2} = u_i, \end{cases} \tag{4.28}$$

where $r = [r_{M+1} \cdots r_{M+N}]^\intercal$, and $x_{i_1} = e_i - e_i^*$, $x_{i_2} = r_i - r_i^*$.

Note that both (4.6) and (4.28) are nonlinear systems. As shown in [130] and [124], first-order linearization around the half-load loading scenario provides a satisfactory performance. Better performances are obtained with the nonlinear switching control policies developed in Section 4.3. The second-order approximation in (4.28) provides a good trade-off between those two approaches, as will be shown through the experimental results. Moreover, note that in (4.28) no further assumptions are made on the dependency of the input voltage of each power buffers, as in (4.9).

Each active load in (4.28) can be generally expressed as

$$\dot{x}_i = f_i(x) + g_i(x) u_i, \quad i \in \mathcal{L}. \tag{4.29}$$

$x_i \in \mathbb{R}^{n_i}$ is the state vector of active load i. Note that the sparsity-promoting approach proposed in the subsequent is valid for any nonlinear system in form (4.29), thus general expressions for n_i and $g_i(x)$ are used. In case of the DC microgrid with power buffers it results that, for each $i \in \mathcal{L}$, $n_i = 2$, $x_i = [x_{i_1} \quad x_{i_2}]^\intercal$, and $g_i(x) = [0 \quad 1]^\intercal$. The overall system's state is $x = \left[x_{M+1}^\intercal, ..., x_{M+N}^\intercal\right]^\intercal \in \mathbb{R}^{\bar{N}}$, where $\bar{N} = \sum_{i=M+1}^{M+N} n_i$. The interconnection of the N subsystems in (4.29) gives the overall microgrid dynamics,

$$\dot{x} = f(x) + g(x) u, \tag{4.30}$$

where $f(x) = [f_{M+1}(x)^\intercal, ..., f_{M+N}(x)^\intercal]^\intercal \in \mathbb{R}^{\bar{N}}$, $g(x) = \mathbf{diag}(g_{M+1}(x), ..., g_{M+N}(x)) \in \mathbb{R}^{\bar{N} \times N}$, and $u = [u_{M+1}, ..., u_{M+N}] \in \mathbb{R}^N$.

The goal is to minimize, at the same time, the number of the communication links (sparsity-promoting objective) and a closed-loop performance index (optimal control objective). To define the overall objective function, the first step is to solve the optimal control problem, whose cost function is

$$J(x, u) = \int_0^\infty U(x, u) dt, \tag{4.31}$$

where $U(x, u)$ is the utility function defined as

$$U(x, u) = Q(x) + \sum_{i \in \mathcal{L}} \rho_i(x) u_i^2. \tag{4.32}$$

with $Q(x)$ and $\rho_i(x)$, $i \in \mathcal{L}$, being positive definite functions weighting the convergence dynamics. Note that unlike (4.14), (4.31) represents an objective shared among all the subsystems. The optimal sparsity-promoting objective function and an algorithmic procedure that optimizes it are provided in the next Section.

4.4.1 Proposed Optimal Sparsity Promoting Methodology

a. Structured Controllers with ADP and Off-policy Learning

The features of the ADP algorithm with off-policy learning (Algorithm 2.2) are exploited in order to obtain a structured optimal controller, i.e., with a communication structure defined a priori, for system (4.30).

As in (4.17), let's start by considering the following system

$$\begin{aligned} \dot{x} &= f(x) + g(x)(u^{(0)}(x) + e_n(t)) = \\ &= f(x) + g(x)(u^{(k)}(x) + u^{(k)'}(x)), \quad \forall k \geq 0, \end{aligned} \tag{4.33}$$

where $u^{(0)}$ is an asymptotically-stable control policy, $e_n(t) : \mathbb{R} \to \mathbb{R}^N$ is the bounded noise injected for learning and exploration purposes, while $u^{(k)'} = u^{(0)} - u^{(k)} + e_n$.

The time-derivative of the function $V^{(k)}(x)$, i.e., the value function at iteration k of the Policy Iteration algorithm 2.1, computed along the state trajectory of (4.33), is

$$\dot{V}^{(k)}(x) = -U(x, u^{(k)}) - 2 \sum_{i \in \mathcal{L}} u_i^{(k+1)} \rho_i(x) u_i^{(k)'}. \tag{4.34}$$

The value function, $V^{(k)}$, and the policies, $u_i^{(k+1)}$, are approximated using LIP approximators,

$$\hat{V}^{(k)}(x) = \sum_{l=1}^{N_V} \omega_l^{(k)} \gamma_l(x) = \omega^{(k)\intercal} \Gamma(x), \tag{4.35}$$

$$\hat{u}_i^{(k+1)}(x) = \sum_{l=1}^{N_U} \theta_{i_l}^{(k)} \xi_l(x_l^{\xi}) = \theta_i^{(k)\intercal} \Xi(x), \tag{4.36}$$

where $\gamma_l(x)$, with $l = 1, ..., N_V$, and $\xi_l(x_l^{\xi})$, with $l = 1, ..., N_U$, are the set of smooth linearly-independent functions returning zero at the origin, with N_V and N_U as integers. The lth basis function $\xi_l(x_l^{\xi})$ depends on a subset of the overall system state, e.g., if $\xi_{\bar{l}}(x_{\bar{l}}^{\xi}) = \xi_{\bar{l}}(x_{M+1}, x_{M+2}, x_{M+4})$, then $x_{\bar{l}}^{\xi} = \{x_{M+1}, x_{M+2}, x_{M+4}\}$. For each $\xi_l(x_l^{\xi}) \in \Xi(x)$, the set $N_l^{\xi} = \{j | x_j \in x_l^{\xi}\}$ is defined. The basis functions set $\Xi(x)$ is the same for each buffer. $\omega^{(k)} \in \mathbb{R}^{N_V}$ and $\theta_i^{(k)} \in \mathbb{R}^{N_U}$, $i \in \mathcal{L}$, are the usual constant weights to be determined, as in (4.20) and (4.21). Now integrating (4.34)

over any time interval, and replacing $V^{(k)}$ and $u_i^{(k+1)}$ with their approximations, the following integral reinforcement learning equation is obtained

$$\begin{aligned}\omega^{(k)\intercal}\underbrace{[\Gamma(x(t_{n+1}))-\Gamma(x(t_n))]}_{\Delta\Gamma(t_{n+1})\in\mathbb{R}^{N_V}} &= -\sum_{i\in\mathcal{L}}\theta_i^{(k-1)\intercal}\left(\int_{t_n}^{t_{n+1}}\Xi(x)\rho_i(x)\Xi^\intercal(x)dt\right)\theta_i^{(k-1)}\\ &-\underbrace{\int_{t_n}^{t_{n+1}}Q(x)dt}_{Q_I(t_{n+1})\in\mathbb{R}}-2\sum_{i\in\mathcal{L}}\theta_i^{(k)\intercal}\underbrace{\int_{t_n}^{t_{n+1}}\Xi(x)\rho_i(x)(u_i^{(0)}+e_{n_i})dt}_{\Psi_i(t_{n+1})\in\mathbb{R}^{N_U}}\\ &+2\sum_{i\in\mathcal{L}}\theta_i^{(k)\intercal}\underbrace{\left(\int_{t_n}^{t_{n+1}}\Xi(x)\rho_i(x)\Xi^\intercal(x)dt\right)}_{\Phi_i(t_{n+1})\in\mathbb{R}^{N_U\times N_U}}\theta_i^{(k-1)}+\epsilon_{k_n}.\end{aligned}\tag{4.37}$$

where ϵ_{k_n} is the approximation error and $\{t_n\}_{n=1}^{N_L}$ is an increasing series of time intervals, with $N_L > 0$ as a sufficiently-large number. By collecting system data for the N_L intervals, the weights $\omega^{(k)}$ and $\theta_i^{(k)}$ are found by minimizing $\sum_{n=0}^{N_L}\epsilon_{k_n}^2$ using least squares. Starting from $u^{(0)}$, sequences $\{\hat{V}^{(k)}\}_{k=0}^{\infty}$ and $\{\hat{u}^{(k+1)}\}_{k=0}^{\infty}$ converge to the optimal values. With a finite number of iterations, near optimal cost function, $\hat{V}(x)$, and control policies, $\hat{u}_i(x)$, $i\in\mathcal{L}$, are obtained.

The distributed near optimal feedback policies, $\hat{u}_i(x)$, that minimize (4.31) depend on the whole system's state, x, i.e., with a fully-connected communication topology. The objective is to minimize the communication links, thus finding a sparse control law that keeps the system stable and minimize (4.31). Let's define a binary decision matrix, $A_d\in\mathbb{R}^{N\times N}$, such that $(A_d)_{ij}=1$ if system j is allowed to send its own state to subsystem i, otherwise, $(A_d)_{ij}=0$. Given a fixed A_d, the matrix $P_i(A_d)\in\mathbb{R}^{N_U\times N_U}$, for each $i\in\mathcal{L}$, is defined as

$$P_i(A_d)=\mathbf{diag}\left(\prod_{j\in N_1^\xi}(A_d)_{ij}\quad\dots\quad\prod_{j\in N_{N_U}^\xi}(A_d)_{ij}\right).\tag{4.38}$$

Therefore, for $(A_d)_{ij}=0$, the $l-th$ diagonal element of $P_i(A_d)$ is zero if the $l-th$ approximating function $\xi_l(x_l^\xi)$ depends on x_j. Given an arbitrary A_d, by constraining the appropriate corresponding weights, $\theta_{i_l}^{(k)}$, a near optimal control policy, $\hat{u}$, with the corresponding underling connectivity pattern can be obtained. To this end Algorithm 4.3 is proposed, the same data collected for the fully-connected communication topology is used to find, if there exists, approximated optimal control policies in line with the communication topology defined by A_d. By exploiting the LIP approximators properties, Algorithm 4.3 makes use of the same training data recorded by applying the input $(u^{(0)}+e_n(t))$ to system (4.30).

The main advantages introduced by using the ADP with off-policy learning approach can be summarized as follows. An approximated optimal feedback controller, that does not require the explicit solution of the HJB equation, is obtained. Using collected system data, the full knowledge of the system dynamics is not required. Finally, the same collected data can be repeatedly used to find the approximated optimal control policies for different communication topologies, hence significantly reducing the computational requirements.

Algorithm 4.3 Off-Policy IRL Algorithm for an Arbitrary A_d

Inputs:

- Initial weights $\omega^{(0)}$, $\theta_i^{(0)}$,
- Recorded system data $\Delta\Gamma(t_n)$, $\Psi_i(t_n)$, $\Phi_i(t_n)$, and $Q_I(t_n)$, with $n = 1, ..., N_L$
- Matrices $P_i(A_d)$, with $i \in \mathcal{L}$;
- A stopping threshold δ.

Outputs: Near-optimal cost function and policies $\hat{\omega}_{A_d}$ and $\hat{\theta}_{i_{A_d}}$, with $i \in \mathcal{L}$.

1. **Initialization:** Set $k = 1$; Evaluate $X_\Gamma = [\Delta\Gamma^\intercal(t_1) \ ... \ \Delta\Gamma^\intercal(t_{N_L})]^\intercal \in \mathbb{R}^{N_L \times N_V}$, and $B_Q = -[Q_I(t_1) \ ... \ Q_I(t_{N_L})]^\intercal \in \mathbb{R}^{N_L}$.
2. **Data Evaluation:** Compute the following matrices

$$X_i = \left[2\left(\Psi_i^\intercal(t_1) - \theta_i^{(k-1)^\intercal}\Phi_i^\intercal(t_1)\right)P_i(A_d) \quad \cdots \quad 2\left(\Psi_i^\intercal(t_{N_L}) - \theta_i^{(k-1)^\intercal}\Phi_i^\intercal(t_{N_L})\right)P_i(A_d)]\right]$$
$$B_\Phi = -\left[\sum_{i\in\mathcal{L}}\theta_i^{(k-1)^\intercal}\Phi_i(t_1)\theta_i^{(k-1)} \quad \cdots \quad \sum_{i\in\mathcal{L}}\theta_i^{(k-1)^\intercal}\Phi_i(t_{N_L})\theta_i^{(k-1)}\right].$$

3. **Policy Improvement:** Find $\omega^{(k)}$ and $\theta_i^{(k)}$, $i \in \mathcal{L}$ from the following least square problem

$$[X_\Gamma \ X_{M+1} \ ... \ X_{M+N}]\left[\omega^{(k)^\intercal} \quad \theta_{M+1}^{(k)^\intercal} \quad ... \quad \theta_{M+N}^{(k)^\intercal}\right]^\intercal = B_Q + B_\Phi.$$

4. **Off-policy Iteration:** If $||\omega^{(k)} - \omega^{(k-1)}|| \geq \delta$, then set $k = k + 1$ and repeat Step 2. Otherwise, stop and return $\hat{\omega}_{A_d} = \omega^{(k)}$, $\hat{\theta}_{i_{A_d}} = \theta_i^{(k)}$, with $i \in \mathcal{L}$.

b. Domain of Attraction Estimation

The stability of the approximated optimal policies depends on the given structure of A_d, as well as on a compact set $\Omega_L \subset \mathbb{R}^{\bar{N}}$ where the data collecting phase has been done. In fact, NNs approximate nonlinear functions on compact sets, and not on the entire $\mathbb{R}^{\bar{N}}$ [65].The stability is verified by quantifying the DoA of the origin in the resulting closed-loop system, i.e.,

$$\mathcal{H} = \{x_0 \in \mathbb{R}^{\bar{N}} | \lim_{t\to\infty} x(t, x_0) = 0\}. \tag{4.39}$$

Once approximated policies are obtained, the function $\hat{V}_{A_d}(x) = \hat{\omega}_{A_d}^\intercal\Gamma(x)$ is employed as a candidate Lyapunov function, whose sub-level set is defined, for any $l \in \mathbb{R}$, as

$$\mathcal{H}_{\hat{V}}(l) = \{x \in \mathbb{R}^{\bar{N}} | \hat{V}_{A_d}(x) \leq l\}. \tag{4.40}$$

Given the difficulty in finding closed forms of the DoA, an estimation is found using data-driven methods. Any sublevel set provides an estimation of the DoA if $\hat{V}_{A_d}(x)$ is positive definite and $\dot{\hat{V}}_{A_d}(x)$ is negative definite within the sub-level set [102]. The goal is to find the

largest invariant set, $\mathcal{H}_{\hat{V}}(l^*)$, representing the largest estimate for the DoA. In the work, the memory-based algorithm presented in [66], adapted to our needs and reported in Algorithm 4.4, is employed. It relies on the evaluation of the candidate Lyapunov function and its derivative on randomly selected data during the learning phase.

Algorithm 4.4 DoA Estimation Algorithm modified from [66]

Inputs:

- Approximated optimal cost function and control policies, $\hat{V}_{A_d}(x) = \hat{\omega}_{A_d}^{\intercal}\Gamma(x)$, $\hat{u}_{i_{A_d}}(x) = \hat{\theta}_{i_{A_d}}^{\intercal}\Xi(x)$, $i \in \mathcal{L}$;
- Sampled data S_{ue}, S_x, and S_{dx};
- Function $g(x)$.

Outputs:

- DoA estimation d_L;
- Maximum/estimated ratio η_V;
- Failed ratio η_F;
- Average cost $\bar{V}$.

Initialization: Set $d_L = 0$, $d_U = \infty$, $N_F = 0$, $M_E = \{0\}$.
for $k = 1, ..., N_{Rs}$ **do**
  Compute $\tau_k = \hat{V}_{A_d}(x(t_{R_k}))$, and $\dot{\tau}_k = \dot{\hat{V}}_{A_d}(x(t_{R_k}))$ as in (4.41).
  if $\dot{\tau}_k < 0$ **and** $\tau_k \geq 0$ **then**
    store $\hat{V}_{A_d}(x(t_{R_k}))$ in M_E; **else** $N_F = N_F + 1$
  end if
  if $\dot{\tau}_k < 0$ **and** $0 \leq d_L < \tau_k < d_U$ **then**
    $d_L = \hat{V}_{A_d}(x(t_{R_k}))$
  else if $\dot{\tau}_k \geq 0$ **and** $0 \leq \tau_k < d_U$ **then**
    $d_U = \hat{V}_{A_d}(x(t_{R_k}))$
    if $d_L \geq d_U$ **then** $d_L = argmax\{e \in M_E | e < d_U\}$
  end if
end for
Compute V_{max} and $\bar{V}$ as the maximum and the average value of M_E, respectively.
return d_L, $\eta_V = d_L/V_{max}$, $\eta_F = N_F/N_{Rs}$, and $\bar{V}$.

Let $T_{Rs} = \{t_{R_i}, i = 1, ..., N_{Rs}\}$ be the set of randomly-selected sampling times. The following sets of sampled data are collected during the learning phase: $S_x = \{x(t_{R_i}), i = 1, ..., N_{Rs}\}$, $S_{dx} = \{\dot{x}(t_{R_i}), i = 1, ..., N_{Rs}\}$, and $S_{ue} = \{s_{ue}(t_{R_i}), i = 1, ..., N_{Rs}\}$, where $s_{ue}(t_{R_i}) = \big(u^{(0)}(x(t_{R_i})) + e_n(t_{R_i})\big)$. For each sampled state, $x(t_{R_k})$, and any weights set, ω_{A_d},

$\theta_{i_{A_d}}$, the following holds

$$\dot{\hat{V}}_{A_d}(x(t_{R_k})) = \omega_{A_d}^{\intercal} \nabla\Gamma(x(t_{R_k}))\left[\hat{f}(x(t_{R_k})) + g(x(t_{R_k}))\left[\theta_{M+1_{A_d}}^{\intercal} \cdots \theta_{M+N_{A_d}}^{\intercal}\right]^{\intercal} \Xi(x(t_{R_k}))\right], \tag{4.41}$$

where $\hat{f}(x(t_{R_k})) = \dot{x}(t_{R_k}) - g(x(t_{R_k}))s_{ue}(t_{R_k})$ is the estimated value of $f(x(t_{R_k}))$. Algorithm 4.4 requires the knowledge of $g(x)$ to compute (4.41) and $\hat{f}(x(t_{R_k}))$. The number of failed trials among the sampled data is N_F. Algorithm 4.4 updates the upper and lower bounds of l^*, i.e., d_U and d_L, respectively, starting from the initial values of $d_L = 0$, $d_U = \infty$.

For each sampled state, $x(t_{R_k})$, the potential estimate for the DoA, i.e., $\hat{V}_{A_d}(x(t_{R_k}))$, is stored in the memory M_E if stability conditions are verified. Then, if it results that $d_L < \hat{V}_{A_d}(x(t_{R_k})) < d_U$ then d_L is updated with the current value of $\hat{V}_{A_d}$ (i.e., the DoA is increasing its radius). Otherwise, if $\dot{\hat{V}}_k(x(t_{R_i})) \geq 0$ then the upper estimate d_U is replaced with the current $\hat{V}_{A_d}$ (i.e., the DoA is decreasing its radius). Then, if $d_L \geq d_U$, the new (lower) estimation is updated as the maximum value among the previously evaluated values stored in M_E such that $d_L < d_U$. After a large number of samples, the estimation d_L increases and provides a conservative estimation of the DoA.

Algorithm 4.4 returns the parameters d_L, η_V, η_F, and $\bar{V}$. η_V is the ratio between d_L and the maximum evaluated cost function. It provides a measure of how small the resulting DoA is compared to the state space spanned during the training phase; E.g., if $\eta_V = 1$, then the DoA is the whole training space. η_F is the ratio between the failed and total trials. Finally, the average cost, $\bar{V}$, provides a performance measure of the resulting controllers in terms of (4.31).

c. Tabu Search

The sparsity-promoting problem can now be defined as

$$\underset{A_d}{\text{minimize}} \quad \beta||A_c \circ A_d||_F^2 + \alpha(A_d) \tag{4.42}$$

where $A_c \in \mathbb{R}^{N\times N}$, $(A_c)_{ij} > 0$ is the cost of the communication link between buffers i and j, $\circ$ denotes the Hadamard product, $||\cdot||_F$ is the Frobenius norm, β is a weighting factor, and $\alpha(A_d)$ is defined in Algorithm 4.5. Due to possible numerical errors in (4.41), the DoA obtained by Algorithm 4.3 may be still valid if η_F is below a given threshold, δ_F, which is a design parameter. If $\eta_F > \delta_F$ or Algorithm 4.3 does not converge, the control policy is considered unstable, with the penalty set to ∞. Otherwise, $\alpha(A_d)$ provides a penalty term proportional to the average performance, $\bar{V}$, and to the reduction of the DoA regarding its maximum span, i.e., $\alpha(A_d) = \bar{V}/\eta_V$. Thus, the compromise between the resulting averaged performances and the number of active communication links is minimized. Clearly, dense communication structures lead to better performances in term of $\bar{V}$. Note that $\alpha(A_d)$ is computed for every A_d using the data collected for the fully-connected structure.

Each solution to problem (4.42) is constituted by the decision variable set, i.e., the communication links status, and the resulting cost. Structural constraints are embedded in the decision variables by defining four different structures as follows. The decision variables are: 1) The diagonal and upper diagonal elements of A_d, i.e., only symmetric links are considered and self loops are optional (elements $(A_d)_{ii}$ can be zero). 2) All the elements of A_d, i.e., non symmetric

communication links are allowed as well as neglected self loops. 3) Only the upper diagonal elements of A_d, with the diagonal elements fixed at 1, i.e., symmetric links and self loops always present. 4) The extra-diagonal elements of A_d, i.e., self loops always present and non symmetric structures allowed.

The TS algorithm reported in Algorithm 4.6 solves the combinatorial optimization problem in (4.42). TS uses a flexible search history to avoid local minimum entrapment [153, 154]. The main features of TS are the moves and the tabu list. Each move m in the moves set, $\mathcal{M}$, generates a new solution when applied to the current one. Implemented moves include: 1) Swap moves where two different decision variables are swapped; 2) Reversion moves where consecutive decision variables are reversed; 3) Insertion moves where each decision variable is changed from 0 to 1 and viceversa.

Algorithm 4.5 $\alpha(A_d)$ Function

Inputs: Matrix A_d; Threshold parameter δ_F.
Outputs: Cost $\alpha(A_d)$.

1. **Off-Policy IRL Convergence Check:** Run Algorithm 4.3 and, if converges, obtain approximated optimal weights and go to Step 2; Otherwise, return $\alpha(A_d) = \infty$.
2. **DoA Estimation:** Run Algorithm 4.4 and obtain η_F, η_V, and $\bar{V}$ parameters. If $\eta_F < \delta_F$, go to Step 3; Otherwise, return $\alpha(A_d) = \infty$.
3. **Cost Evaluation:** Return $\alpha(A_d) = \bar{V}/\eta_V$.

Algorithm 4.6 Tabu Search Algorithm

Inputs: Initial solution S_0; Tabu length T_L; Set of moves $\mathcal{M}$.
Outputs: Best solution S_B^*.

1. **Initialization:** For every move $m \in \mathcal{M}$, initialize the corresponding tabu counter, $T_C(m)$, to zero; Set the initial best solution $S_B^* = S_0$; Set the best candidate solution $S_B = S_0$.
2. **Best candidate solution evaluation:**
   a. **for each** $m \in \mathcal{M}$ **do**
   b. **if** $T_C(m) = 0$ **then**
   c. Apply move m to S_B and obtain solution $S_{B,m}$
   d. **if** $S_{B,m}$ is better than S_B **then** Set $S_{B,m} = S_B$ and set the best move, m_B, to m.
   e. **end if**
   f. **end for**
3. **Best solution evaluation:**
   if S_B is better than S_B^* **then** update $S_B^* = S_B$.
4. **Tabu list update:** Add m_B to the tabu list by setting $T_C(m_B) = T_L$. For each $m \in \mathcal{M}$, $m \neq m_B$, decrease $T_C(m)$ by 1 if greater than 0.
5. **Stopping criterion:** Go to Step 2 until the maximum number of iteration is reached.

In summary, the best solution is initialized with a fully-connected feedback. Each TS iteration seeks the best non-tabu move that improves the current best solution. Then, the best move is inserted in the tabu list whose length provides the number of TS iterations in which the move is forbidden, allowing better exploration and escaping the local minimum. Finally, the

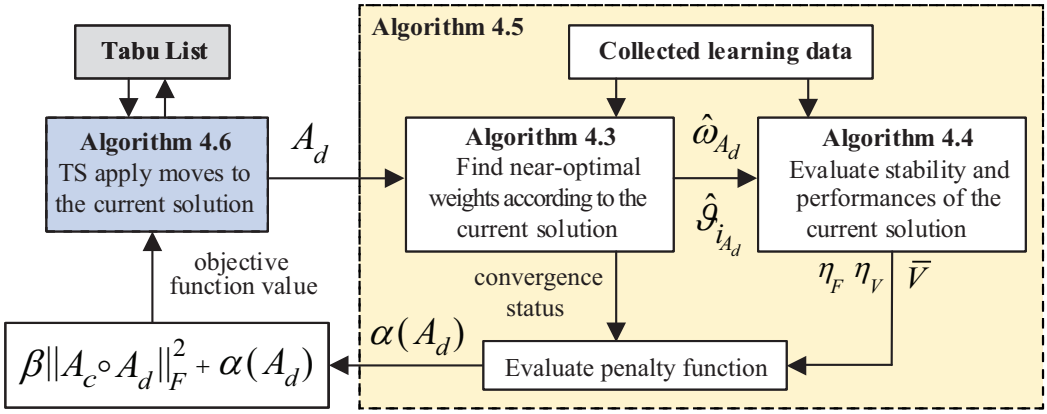

Figure 4.15: Information flow between algorithmic components in the proposed sparsity-promoting approach.

relationships between the algorithmic components of the proposed approach are graphically represented in Fig. 4.15.

4.4.2 CHIL Validation

a. System Setup

Verification studies are conducted on the 48V DC microgrid depicted in Fig. 4.16, where $M = 5$ and $N = 6$, with $v_{si} = 50V$ and $r_{si} = 0.1\Omega$. Line resistances are set as follows

$$\begin{aligned}
&r_{11,12} = r_{14,15} = r_{20,4} = r_{15,16} = r_{16,7} = r_{17,5} = 0.5\Omega,\\
&r_{12,13} = r_{8,18} = r_{9,20} = r_{16,21} = r_{21,10} = 0.6\Omega,\\
&r_{13,6} = r_{6,14} = r_{19,3} = r_{20,21} = 0.3\Omega,\\
&r_{12,1} = r_{11,8} = r_{15,2} = r_{7,17} = 0.4\Omega,\\
&r_{19,9} = r_{14,9} = r_{10,17} = 0.7\Omega,\\
&r_{18,19} = 0.2\Omega,\\
&r_{13_{18}} = 0.9\Omega.
\end{aligned} \tag{4.43}$$

The architecture of each active load is as in Fig. 4.5 with th same parameters reported in 4.1. CHIL validations are conducted using a setup similar to Fig. 4.6, where the communication network and control schemes are emulated on a dSpace MicroLabBox system, while the physical microgrid is emulated on a Typhoon HIL604 hardware. Communication and controller sampling times are $1ms$ and $0.1ms$, respectively.

The control objectives are two fold: 1) Regulate the output voltage of each buffer in the steady state at $v_{bi}^* = 100V$, with a corresponding $e_i^* = 22J$; 2) Vary the input impedance, r_i, according to the sparse distributed policy. Both objectives are addressed using the fast voltage tracker of each boost converter. The resulting scheme is shown in Fig. 4.17. The ith active load receives the states $\{x_j\}_{j\in N_i}$, where herein N_i denotes the set of other buffers that communicate with the buffer i, i.e., $N_i = \{j|(A_d)_{ij} = 1\}$. Note that in Fig. 4.17, $x = \{x_i \cup \{x_j\}_{j\in N_i}\}$. As in the previous control scheme in Fig. 4.3, the near-optimal control policy u_i is applied to the software implementation of (4.6) whose integral provides $\bar{e}_i$ and $\bar{r}_i$. Then, by translating $\bar{e}_i$ into

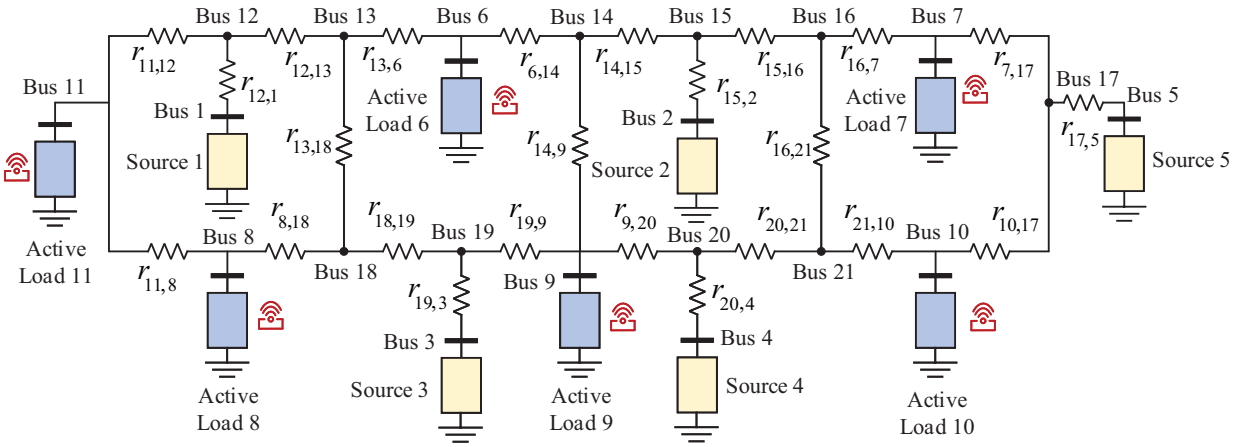

Figure 4.16: Considered DC microgrid layout.

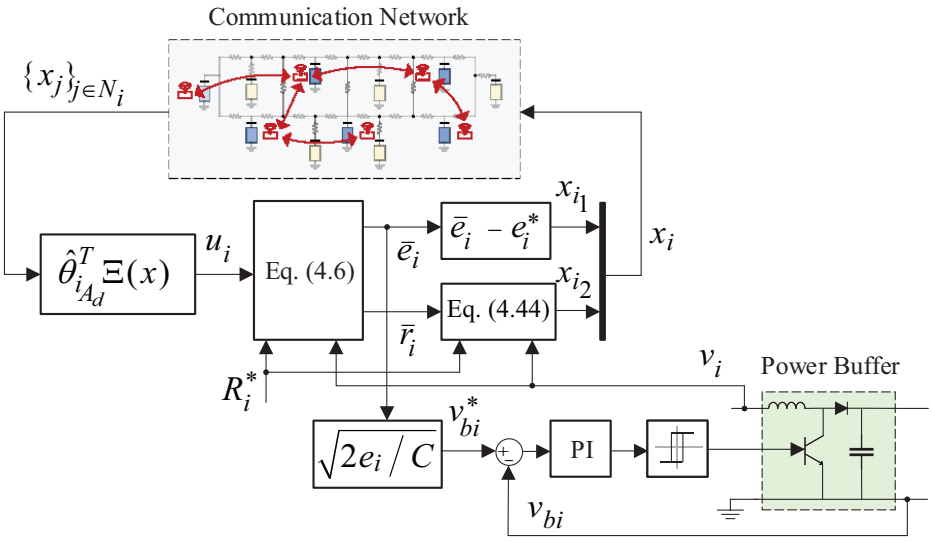

Figure 4.17: Control scheme of the ith power buffer.

the reference of the voltage tracker, using (4.2), the two control objectives mentioned above are attained.

A control policy designed around the half-load operating condition is used and validated for other operating points, as done in [130] and [124]. The resulting feedback controller requires the knowledge of local states, x_{i_1} and x_{i_2}, which represent the deviations with respect to the target operating point. The target stored energy is fixed at e_i^*, thus, the local state x_{i_1} is easily obtained as $x_{i_1} = \bar{e}_i - e_i^*$. Instead, to obtain x_{i_2}, the unknown value of r_i^*, that depends on the overall operating point, is required. In [124] a low-frequency filter extrapolates the quiescent part of the input resistance, i.e., r_i^* to determines the corresponding actual deviation. However, this filter could introduce delays, distortions, and computational demand. Alternatively, in the proposed approach the following approximation is adopted

$$x_{i_2} \approx \bar{r}_i - \frac{C}{2}\frac{R_i^* v_i^2}{e_i^*}, \tag{4.44}$$

wherev_i is the measured input voltage and $\bar{e}_i$ represents the energy profile to be tracked by

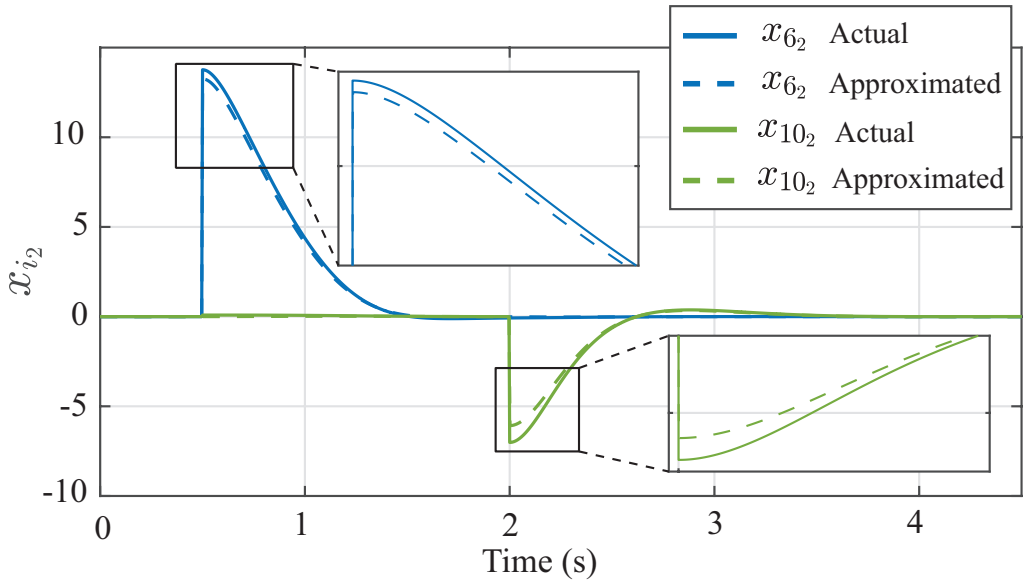

Figure 4.18: Comparison of actual states and approximated ones using (4.44).

the power buffer. The knowledge of the target load, R_i^*, is needed in (4.44). Assuming an ideal buck converter, R_i^* is easily related to the desired load R_{L_i} as $R_i^* = (v_{bi}^*/v_{oi}^*)^2 R_{L_i}$, where v_{oi}^* is the fixed output voltage of the buck converter. The comparison between the actual and approximated states x_{i_2} in Fig. 4.18 shows the effectiveness of (4.44). The depicted scenario uses the local feedback policy $u_i = 2x_{i_1}$, $i = 6, ..., 11$, and varies the final resistive loads of buffers 6 and 10 from 20Ω to 10Ω in $t = 0.5s$ and from 10Ω to 16Ω in $t = 2s$, respectively.

b. Optimizing the Communication Topology

Algorithm 4.6 is employed to solve problem (4.42) for different values of β. Starting from a fully-connected controller, the TS procedure modifies the current communication topology by applying a set of moves and defining the solutions to visit. For each visited solution, characterized by a specific communication topology, the ADP with off policy procedure in Algorithm 4.3 finds the corresponding optimal controller. A set of learning data, i.e., $\Delta\Gamma(t_n)$, $\Psi_i(t_n)$, $\Phi_i(t_n)$, $Q_I(t_n)$, with $i \in \mathcal{L}$ and $n = 1, ..., N_L$, is previously collected and used in every run of Algorithm 4.3. Such data collecting phase is conducted in the Simulink environment on the interconnection of the N subsystems (4.29) with half-loads values, i.e. $R_i^* = 50\Omega$, $i = 6, ..., 11$. Second-order polynomial terms in the 12 states are considered as approximating functions in $\Gamma(x)$, while $\Xi(x) = x$. The learning time intervals are $N_L = 5000$ of $0.01s$ length. The initial controller is $u_i^{(0)} = 2x_{i_1}$, $i = 6, ..., 11$, the stopping threshold is $\delta = 10^{-4}$, and filtered white noises are used as exploration signals.

The stability and performance, i.e., the average value function, of each visited solution are evaluated using Algorithm 4.4. In particular, each visited solution could be: 1) Unstable if Algorithm 4.3 does not converge or if the ratio of the failed stability checks, η_F, is higher than the δ_F threshold, herein set to 0.01; 2) Stable with a DOA smaller than the training space, i.e., $\eta_V < 1$; 3) Stable on the full training space, i.e., $\eta_V = 1$. The DoA is estimated using $N_{Rs} = 6000$ randomly-sampled data during the learning stage. Finally, the utility function is

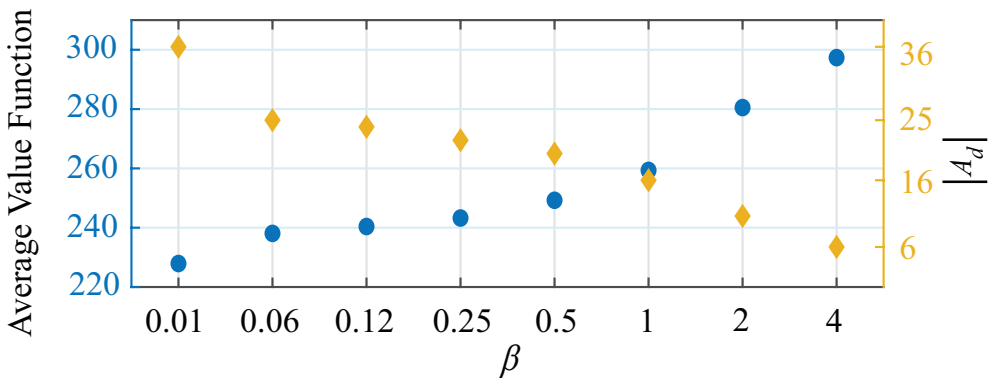

Figure 4.19: Optimal average value function and cardinality of A_d for several β.

defined with $\rho_i(x) = 2$, $i = 6, ..., 11$, and $Q(x) = x^\intercal Q_U x$, where

$$Q_U = \begin{bmatrix} Q_d & Q_2 & Q_6 & Q_6 & Q_2 & Q_4 \\ Q_2 & Q_d & Q_2 & Q_3 & Q_6 & Q_2 \\ Q_6 & Q_2 & Q_d & Q_4 & Q_2 & Q_6 \\ Q_6 & Q_3 & Q_4 & Q_d & Q_4 & Q_2 \\ Q_2 & Q_6 & Q_2 & Q_4 & Q_d & Q_2 \\ Q_4 & Q_2 & Q_6 & Q_2 & Q_2 & Q_d \end{bmatrix}, \tag{4.45}$$

with $Q_d = \mathbf{diag}(30, 15)$ and $Q_k = \mathbf{diag}(-k, 0)$. All entries of matrix A_c are 1. The decision variables are the extra-diagonal elements of A_d, i.e., non-symmetric communication links are allowed while self-loops are present. Each trial of Algorithm 4.6 uses a tabu length of 15 and a maximum number of iterations of 100.

The average-value function, i.e., $\bar{V}$ in Algorithm 4.4, and the resulting cardinality, $|A_d|$, of the optimal solutions obtained by eight different trials of Algorithm 4.6 for increasing values of the weight β in (4.42), are reported in Fig. 4.19. Greater values of β promote sparsity with a decreasing number of active communication links. For $\beta = 0.01$, a fully-connected pattern is obtained, i.e., $|A_d| = 36$. For $\beta = 4$, only local controllers are obtained, i.e., $|A_d| = 6$. As expected, increasing sparsity leads to a lower performance evaluated within the randomly-sampled data during the learning phase.

Figure 4.20 elaborates the results for $\beta = 0.5$ and $\beta = 2$. Figures 4.20(a), 4.20(b), and 4.20(c) show the visited solutions during the optimization procedure. Figure 4.20(a) presents the visited unstable solutions. The objective function has an infinite value for both $\beta = 0.5$ and $\beta = 2$ despite the gap depicted for presentation purposes only. Figure 4.20(b) shows the visited stable solutions with $\eta_V < 1$, which implies higher values of the objective function, especially when the corresponding DOA is significantly smaller than the training space. The optimal and visited solutions when $\eta_V = 1$ are depicted in Fig. 4.20(c). Due to its greater value, $\beta = 2$ has higher values of both optimal and visited solutions compared with those of $\beta = 0.5$ in Fig. 4.20(c). In both cases, proper operation of TS is exhibited through intensified, i.e., more dense, searches around optima. Figure 4.20(d) shows the trend of best solutions during TS iterations. For $\beta = 0.5$ and $\beta = 2$, the optimum is reached in 38 and 34 iterations, respectively. Finally, Fig. 4.20(e) shows the optimal communication topologies. Cardinalities of $|A_d|$ for $\beta = 0.5$ and $\beta = 2$ are 20 and 10, respectively.

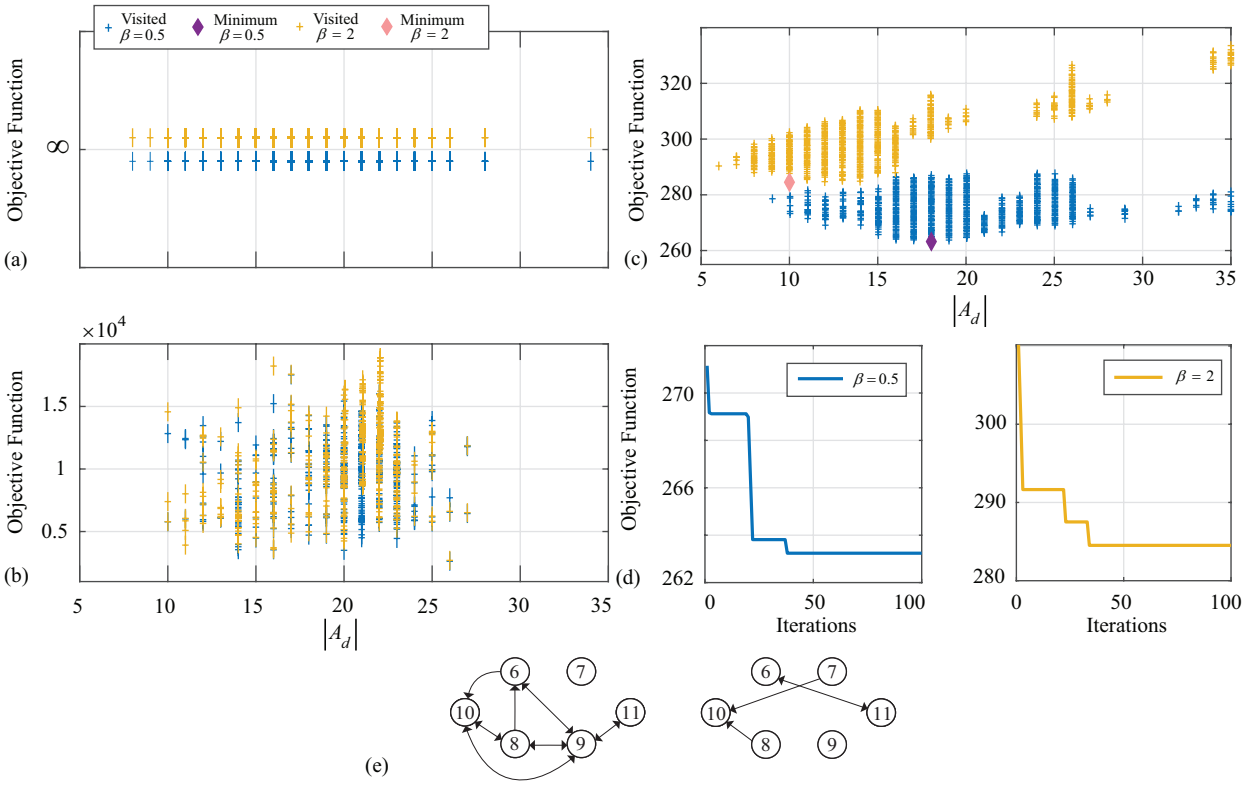

Figure 4.20: Results of the optimization stage: (a) Visited unstable solutions for $\beta = 0.5$ and $\beta = 2$; (b) Visited stable solutions with $\eta_V < 1$ for $\beta = 0.5$ and $\beta = 2$; (c) Visited and optimal solutions with $\eta_V = 1$ for $\beta = 0.5$ and $\beta = 2$; (d) Best solution for each tabu-search iteration; (e) Optimal communication topologies when $\beta = 0.5$ (left) and $\beta = 2$ (right).

c. CHIL Studies

CHIL studies for two optimal control policies, with $\beta = 0.5$ and $\beta = 2$, are reported in Fig. 4.21 and Fig. 4.22, respectively. Final load resistances are $R_{L_6} = 24\Omega$, $R_{L_7} = 18\Omega$, $R_{L_8} = 35\Omega$, $R_{L_9} = 7\Omega$, $R_{L_{10}} = 9\Omega$, and $R_{L_{11}} = 28\Omega$. As seen in Fig. 4.21(c) and Fig. 4.22(c), both scenarios consider the step change in load at $t = 1s$ when load 6 doubles its power demand, i.e., $R_{L_6} = 12\Omega$, at $t = 5s$ when load 7 halves its power demand, i.e., $R_{L_7} = 36\Omega$, at $t = 9s$ when load 8 doubles its power demand, i.e., $R_{L_8} = 17.5\Omega$, at $t = 12s$ when load 10 halves its power demand, i.e., $R_{L_{10}} = 18\Omega$, and at $t = 15s$ when load 11 doubles its power demand, i.e., $R_{L_{11}} = 14\Omega$.

Different communication topologies, as in Fig. 4.20(e), imply different control behaviors during transients, as highlighted in Fig. 4.21(a) and Fig. 4.22(a). The buffer voltages, v_{bi}, $i = 6, ..., 11$, reflect changes in the stored energy according to the distributed control policies actuated by the scheme in Fig. 4.17.

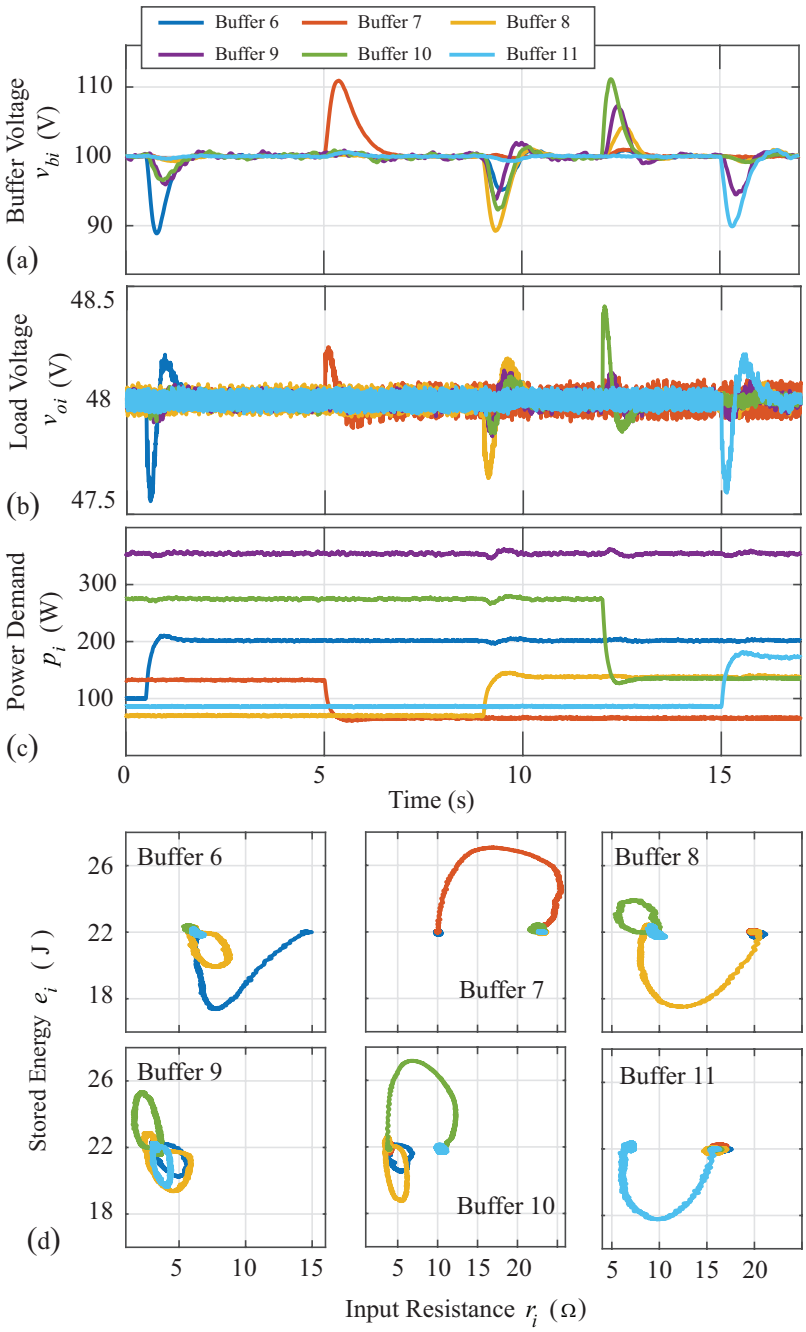

Figure 4.21: CHIL validation when $\beta = 0.5$: (a) Output voltage of power buffer; (b) Output voltage at terminal load resistances; (c) Output power of power buffers; (d) Energy-impedance trajectories.

During the first load change, the topology obtained with $\beta = 0.5$ allows power buffers 9 and 10 to change their stored energies and actively assist load 6. With a more sparse communication topology, i.e., $\beta = 2$, the power buffer 6 is assisted only by the power buffer 11. This results in the increased usage for buffer 6 if compared with the previous topology, see Fig. 4.21(a) and Fig. 4.22(a) during the first transient. Similar considerations are made for the second load change, where assistance is provided for the case of $\beta = 2$ where buffer 10 assists buffer 7.

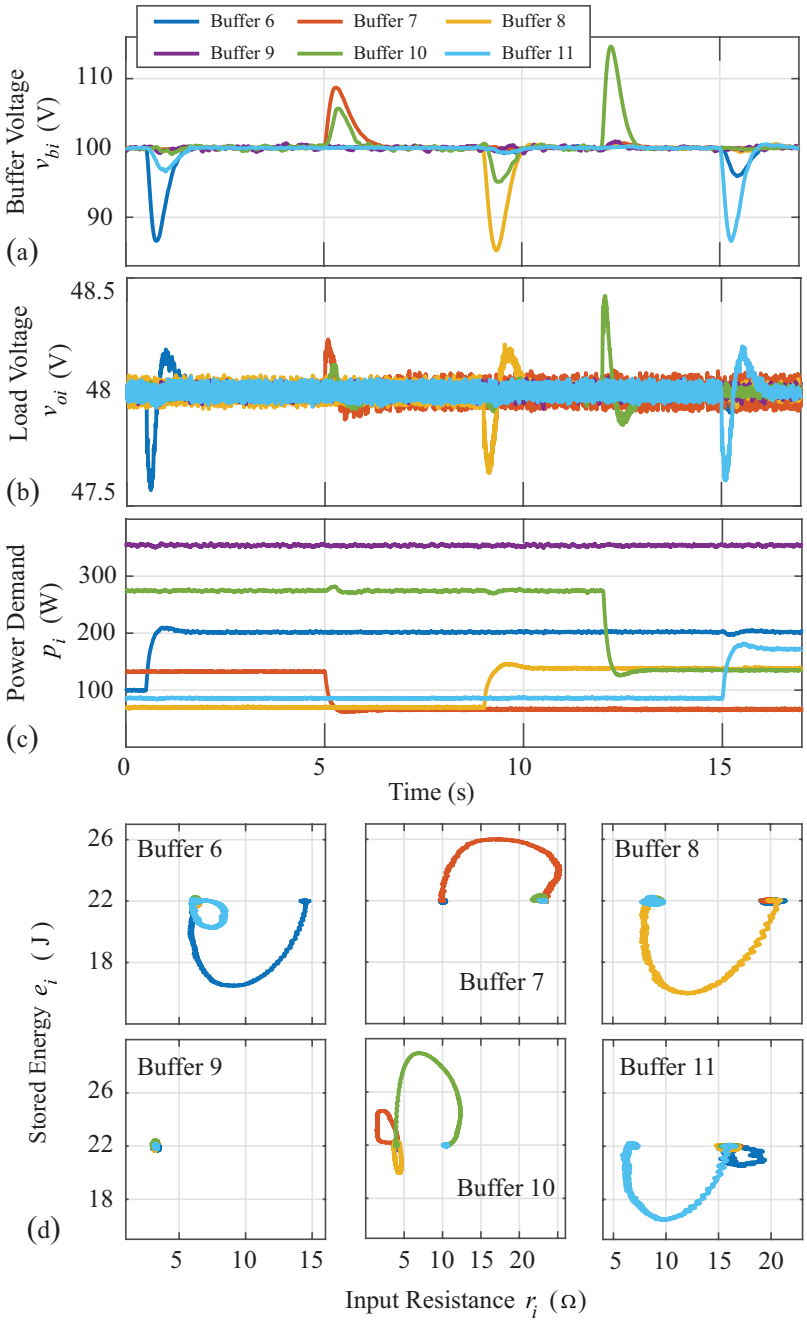

Figure 4.22: CHIL validation when $\beta = 2$: (a) Output voltage of power buffer; (b) Output voltage at terminal load resistances; (c) Output power of power buffers; (d) Energy-impedance trajectories.

For $\beta = 0.5$, buffer 7 does not communicate its states, with no changes in other buffer energies. For $\beta = 0.5$, power buffers 6, 10, and 9, reduce the energy usage of buffer 8 during the third transient, when compared with the case of $\beta = 2$, where buffer 8 is assisted only by buffer 10. Better performances are obtained with $\beta = 0.5$ during the fourth and fifth load changes. Figures 4.21(b) and 4.22(b) show how the load voltages do not substantially change during the transients due to the buffering capabilities of the power buffers.

Energy-impedance trajectories are reported in Fig. 4.21(d) and Fig. 4.22(d), for the two

topologies, respectively. Trajectories during the first, second, third, fourth, and fifth load changes are depicted in blue, orange, red, green, and light blue, respectively. Input impedances and stored energies are modified to provide assistance according to the optimized communication topology. For instance, power buffer 9 reacts to changes in buffers 6, 8, 10, and 11, for $\beta = 0.5$. For $\beta = 2$, where only a local controller is active, the stored energy remains constant at its rated value. Less sparse topologies imply more assistance in terms of faster transient responses with lower energy usage. Note that utility functions, i.e., $Q(x)$ and $\rho_i(x)$, can enhance the performances of specified buffers, e.g., by increasing the corresponding diagonal weighting terms in (4.45).

The effectiveness of the proposed method is demonstrated through a comparison with two other approaches. The first comparison is made with the linear sparsity-promoting algorithm in [145] and used in [58] and [149] for AC microgrid applications. This algorithm is applied to the first-order linearization of (4.3). The algorithm is tuned such that the resulting optimal topology has the same number of active links as the one obtained by the proposed method. The second comparison is made with an optimal LQR obtained on the first-order linearization of (4.3) and truncated such that the communication topology coincides with that of the proposed approach. Comparisons are made for the scenario in Fig. 4.21 and Fig. 4.22, and for various β parameters, i.e., $\beta = 0.5$, $\beta = 1$, $\beta = 2$, and $\beta = 4$ (fully-decentralized controller). While the computational requirements of the proposed method are higher when compared with [145], the proposed approach could handle nonlinear systems. In both cases, the optimization procedure is conducted offline. Since the basis function set $\Xi(x) = x$ provides a linear feedback controller, the implementation of the proposed real-time controller requires the same computational resources as other controllers obtained via [145] and the truncated LQR.

Figure 4.23 compares the buffer voltages obtained with the proposed approach (continuous line), [145] (dotted line), and truncated LQR (dashed line), when $\beta = 0.5$ (Fig. 4.23(a)), $\beta = 1$ (Fig. 4.23(b)), $\beta = 2$ (Fig. 4.23(c)), and $\beta = 4$ (Fig. 4.23(d)). Note that the communication topology obtained with [145] differs from the one obtained by the proposed approach. Thus, when comparing the same load changes, the set of assistive power buffers is different. Compared with both the truncated LQR approach and [145], the proposed method provides faster recovering times for each buffer subject to the load change, i.e., the time needed to restore its initial energy level corresponding to $v_{bi} = 100V$, with a lower maximum energy utilized. The proposed approach shows higher energy drawn from the assisting buffers, to help with the faster restoration of the buffer subject to the load change, e.g., during the first load change in Fig. 4.23(a), during the last load change in Fig. 4.23(b), and during the third load change in Fig. 4.23(c), see corresponding zoomed parts. The proposed method always shows better performance compared with the truncated LQR method. On the other hand, due to different optimized communication topologies, [145] could sometimes show better behaviors, e.g., the second load change in Fig. 4.23(a), and the third load change in Fig. 4.23(b), where the corresponding optimized topologies obtained with the proposed method do not provide assistance for buffers 7 and 8, respectively. However, the proposed approach shows better overall performances on the majority of the loading events, with a smaller overall utility function, as shown next. It also provides better responses with fully decentralized controllers, as in Fig. 4.23(d).

Finally, Table 4.2 compares the proposed method against the two other approaches in terms of the resulting utility functions. The base value used to evaluate the percentage variation in the fifth column is the one obtained by applying the fully-connected optimal controller with

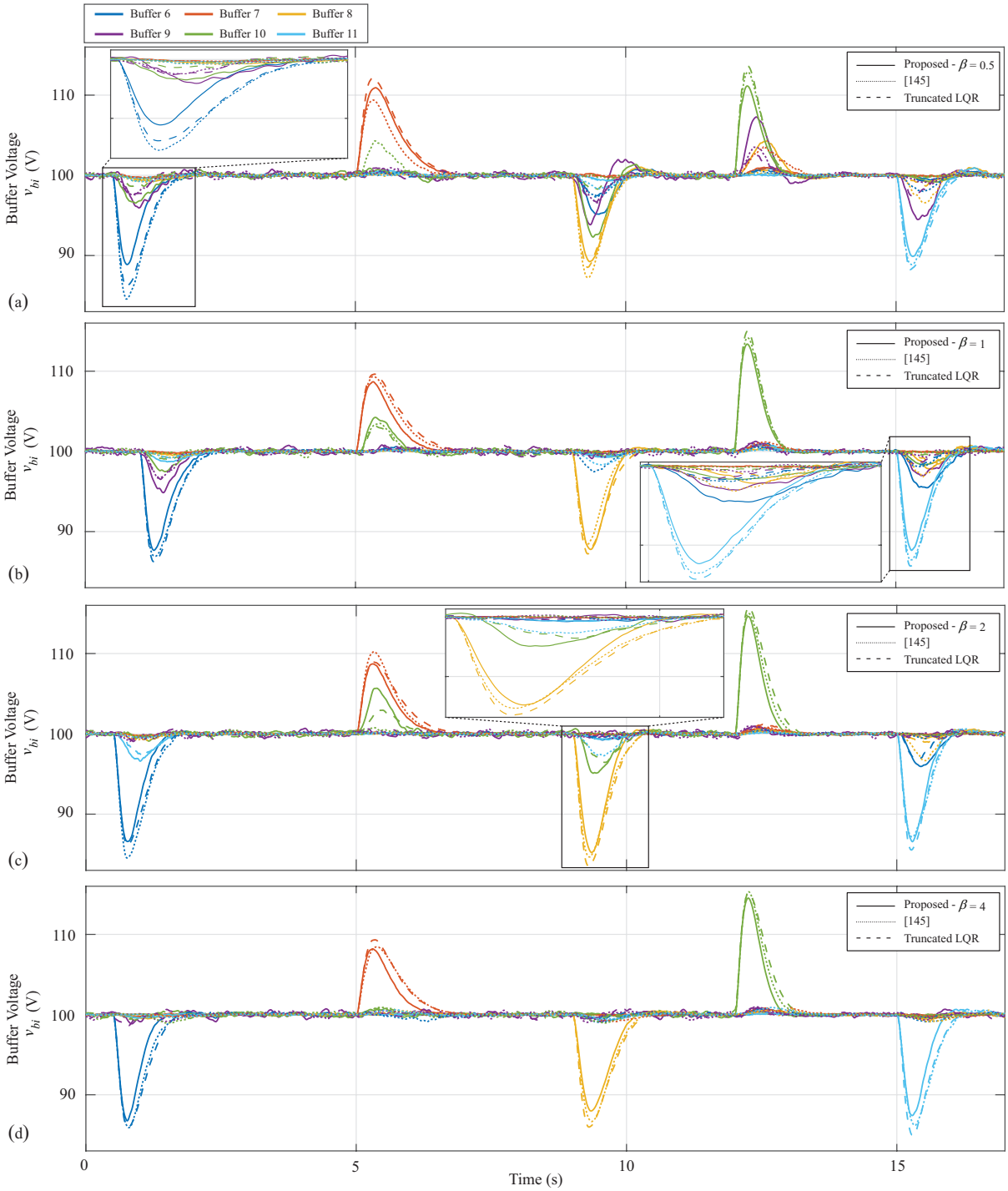

Figure 4.23: Comparison of the buffer voltages using the proposed approach, [145], and the truncated LQR: (a) $\beta = 0.5$; (b) $\beta = 1$; (c) $\beta = 2$; (d) $\beta = 4$.

$\beta = 0.01$. As shown in Fig. 4.19, greater β implies more sparsity with higher performance values. The proposed approach finds a better compromise between the resulting performance and the number of active communication links, i.e., by comparing topologies for $\beta = 0.5$ and $\beta = 2$, some communication links are activated, and other deactivated, to minimize the impact on the performance index. The proposed method outperforms other approaches, even with more

sparse communication topologies (e.g., compare row 6 with rows 4 and 5 in Table 4.2).

Table 4.2: Closed-loop performance comparison between proposed approach, [145], and truncated LQR

	β	$\lvert A_d \rvert$	Utility	Variation %
Proposed	0.01	36	7702.6	0
LQR		36	7910.3	1.4
Proposed	0.5	18	7883.5	2.3
[145]		18	8150.8	5.8
Truncated LQR		18	8187.4	6.3
Proposed	1	16	7964.7	3.4
[145]		16	8158.8	5.9
Truncated LQR		16	8128.8	5.5
Proposed	2	10	8096.5	5.1
[145]		10	8194.8	6.4
Truncated LQR		10	8227.5	6.8
Proposed	4	6	8137.6	5.7
[145]		6	8193.3	6.4
Truncated LQR		6	8292.5	7.7

4.5 Conclusions

In this chapter another application of the ADP approach for the optimal control of complex nonlinear systems has been presented. The DC microgrid with power buffers for load decoupling has been considered as case study. In particular, ADP provided a set of optimal distributed control policies regarding two different approaches.

The first approach considered the fully nonlinear dynamics of the DC microgrid with power buffers and a predefined communication topology where the distributed controllers operate. ADP solves a set of optimal control problems, for each subgroup of power buffers and for a mesh of loading scenarios, providing a set of optimal assistive control policies. With such policies, each power buffer helps the neighboring buffers to smooth the transients during abrupt load changes, improving the performances and stability properties of the DC microgrid. In contrast with the existing literature, the proposed design is based on optimal cooperative strategies, and provides direct feedback controllers by solving the HJB equation via ADP. Moreover, it does not need a turn-based approach, and does not consider a small-signal approximation. Finally, the effectiveness of the proposed approach has been validated through experimental CHIL studies.

The second approach developed the sparsity-promoting optimal control of power buffers. Existing distributed solutions for power buffers in DC microgrids do not consider the effects

of the communication network topologies on the controller performances. Optimal sparse controllers find the best trade-off between the number of the activated communication links and the minimization of a defined closed-loop performance index. While sparsity promoting optimal controllers have been developed for linear systems, a solution for nonlinear system has not been presented yet. Based on ADP and TS methods, the first attempt in developing sparsity promoting optimal control approaches for general nonlinear systems has been proposed in this chapter. In particular, TS seeks the best solution by applying some moves on the decision variables matrix, i.e., the communication topology. Controller performance and stability, corresponding to each topology, are evaluated using the ADP with off-policy learning algorithm and a DoA estimation algorithm. While appearing intuitive, showing that less sparse communication topologies provide better performances is not trivial for nonlinear systems such as DC microgrids. Through CHIL studies, performance improvement following the use of a less sparse communication topology is reflected in a better mutual assistance among the buffers, i.e., faster transient responses with less stored energy utilized. Finally, quantitative comparisons showed that the proposed approach outperforms existing methods.

4.6 Publications

The results presented in this chapter have been published by the author in [162] and [163].

Chapter 5

Conclusions and Future Work

In this doctoral thesis, the application of Reinforcement-Learning (RL) techniques for the optimal control of complex systems has been discussed. The main goal has been to evaluate the potential of such methodologies when dealing with different applications, namely the optimal control aimed at minimizing the actuation energy of a Dielectric Elastomer Actuator (DEA), and the distributed optimal control of power buffers in DC microgrids.

First, RL-based algorithms that solve optimal control problems for nonlinear systems have been over-viewed. Based on a framework typically encountered in the RL theory, i.e., the actor-critic structure, the Adaptive Dynamic Programming (ADP) approach provides approximated solutions of the Hamilton-Jacobi-Bellman (HJB) equation using neural networks. Different learning strategies overcome the mathematical intractability of the HJB equation. In particular, two main solutions have been discussed and compared in Chapter 2: the Policy Iteration (PI) algorithm with on-policy learning, and the PI algorithm with off-policy learning. Pros and cons of those two methods have been highlighted. An on-policy method tunes the approximated optimal policy during normal system's operation, overcoming parameters variations. However, its real-time implementation is computationally intensive, closed-loop stability during the learning stage is hard to design, and a full or partial knowledge of the system dynamics is required. On the other hand, the off-policy approach is conducted off-line, i.e., the optimal policy is learned once a defined batch of information have been previously collected using a stable exploring policy. Full or partial dynamics knowledge is not required, and the real-time computation of the resulting policy is reduced. Clearly, an off-policy approach does not take into account parameter variations, i.e., if some of the system parameters change over time, the off-line learning procedure has to be re-executed. The author suggests that an off-policy method is better suited for real-world applications, where computational burdens and optimization efficiency are crucial elements for a successful feedback control system. Real-world applications in Chapter 3 and Chapter 4 prove such argument. Examples of on-policy and off-policy approaches have been also presented in Chapter 2. In particular, the last example dealt with the optimal structured control of symmetrically-coupled systems with partially unknown dynamics. Such example demonstrated how the off-policy method offers a powerful and versatile mean to tackle optimal control problems where the system dynamics is unknown, even with complex constraints such as the structural one.

The objective of the doctoral research has been to develop optimal controllers, using ADP, able to overcome limits and performances of existing approaches when dealing with two real-

world applications. Procedures and obtained results have been presented and discussed in the main body of this thesis, i.e., Chapter 3 and Chapter 4.

The closed loop optimal control of DEAs has been considered in Chapter 3. The objective has been the minimization of the electric energy required to actuate the device in a position control scheme. Such goal has never been addressed by the existing literature. The first step consisted in an accurate physical modeling procedure aimed at predicting the highly nonlinear behavior of the actuator. In particular, based on thermodynamic considerations, a free-energy model has been developed to describe the system dynamics. The passivity of such model enabled to define and quantify the energy dissipated during the actuation. An experimental identification procedure validated the effectiveness of both system and energy loss models. Afterwards, the energy minimization task has been formulated according to the optimal control theory, considering the energy loss expression as utility function. Due to the involved nonliearities, the resulting optimal control problem has been solved via ADP. The off-policy approach has been employed as a tool to solve off-line several optimal control problems defined according to different target displacement values. Experimental validations assessed the performances of the obtained optimal policies as well as other traditional approaches, i.e., Proportional-Integral and feed-forward controllers. The proposed approach showed significant improvements in terms of energy savings during both charging and discharging tasks. Furthermore, the experiments highlighted how the trade-off between speed of response and energy saving is easier to tune and predict using the proposed approach instead of Proportional-Integral controllers. Finally, a robustness analysis determined the main parameters affecting the controller performances.

Future developments include the integration of the energy-efficient approach with self-sensing control schemes, and the application of the proposed method to complex soft robots structures. Additional studies could also arise when comparing the proposed approach with discrete-time finite-horizon ADP methods, accounting for the true losses occurring during a specific finite-time positioning task.

Chapter 4 designed distributed optimal control policies for power buffers in DC microgrids, using ADP. The objective was to develop distributed assistive control policies, so that each buffer can benefit from the energy stored in the neighboring buffers when abrupt load changes occur during normal operations. Power buffers represent a meaningful way to overcome the stability issues of DC microgrids with no damping elements. Therefore, proper control of such devices is crucial for the performances of the network. Although distributed and decentralized control strategies have been presented in the recent literature, in this thesis cooperative control policies that consider the nonlinear dynamics of the entire microgrid have been developed. Clearly, the communication network enabling distributed controllers plays an important role. Regarding to this, two different control approaches, both based on ADP, have been presented.

The first approach considered a fixed communication graph dictated by the physical vicinity of the buffers. A fully nonlinear model describing the dynamics of each power buffer as well as their coupling relationships has been developed. The cooperative assistive control task has been formulated according to the optimal control theory, with a single utility function shared among all the buffers. The utility function has been designed so that its minimization implies a better reciprocal assistance. Due to the system's nonlinearity, ADP is employed to solve a set of optimal control problems for a mesh of loading scenarios, providing a set of optimal assistive control policies. Controller/Hardware-In-the-Loop (CHIL) studies showed how such policies effectively provide assistance to the neighboring buffers, helping to smooth the transients during

abrupt load changes, and finally improving both performances and stability properties.

The second approach investigated the sparsity-promoting optimal control of power buffers. Generally speaking, when dealing with distributed control systems, it is desired to make the adjacency matrix of the communication graph as sparse as possible without jeopardizing control performances. Sparsity-promoting optimal control problems have been successfully solved for linear systems using traditional optimization methods. However, the sparsity-promoting optimal control of nonlinear systems has never been investigated by the existing literature. The purpose of the presented work has been twofold: 1) To provide an effective tool to solve the sparsity-promoting optimal control for general nonlinear systems; 2) To develop sparsity-promoting optimal controllers for power buffers. Note that the existing distributed solutions for power buffers do not consider the effects of the communication network topology on the controller performances. The proposed sparsity-promoting algorithm makes use of ADP with off-policy learning, Tabu Search (TS) optimization, and Domain of Attraction (DoA) estimation methods. In particular, the TS algorithm defines the different communication topologies to asses, ADP finds optimal controllers according to each of those topologies, while the DoA estimation algorithm evaluates their stability. The proposed approach has been applied to the DC microgrid with power buffers, showing how less-sparse topologies encourage reciprocal assistance. CHIL studies proved the effectiveness of the proposed approach, which outperforms both in a quantitative and qualitative way the existing linear methods. Finally, this work highlighted once again the versatile and powerful capabilities of the off-policy method. In fact, the versatility of such strategy allowed an easy integration with TS optimization and DoA estimation techniques. Moreover, since the off-policy method makes a repeated use of the same collected data, the computational complexity has been sensibly reduced: a single batch of information is used to find optimal controllers according to different incrementally-sparse communication topologies.

Future studies can enhance the proposed strategies by including the development of self-learning algorithms with plug-and-play features for power buffers. Also, cooperative optimal control strategies can be developed for DC sources as well. Finally, sparsity-promoting objectives can also be included in finite-time ADP techniques, providing new tools to solve sparsity-promoting finite-time and discrete-time optimal control problems for nonlinear systems.

Bibliography

[1] A. Daron and R. Pascual, "Automation and new tasks: How technology displaces and reinstates labor," *Journal of Economic Perspectives*, vol. 33, no. 2, pp. 3–30, May 2019.

[2] "Executive summary world robotics 2019 industrial robots," International Federation of Robotics, Tech. Rep., Sep. 2019.

[3] P. Faraboschi, E. Frachtenberg, P. Laplante, K. Mansfield, and D. Milojicic, "Technology predictions: Art, science, and fashion," *Computer*, vol. 52, no. 12, pp. 34–38, Dec. 2019.

[4] R. Sargent, "Optimal control," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 361–371, Dec. 2000.

[5] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004.

[6] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[7] R. Bellman, *Dynamic Programming (Dover Books on Computer Science)*. Dover Publications, 2003.

[8] D. Liberzon, *Calculus of Variations and Optimal Control Theory*. Princeton University Press, Dec. 2011.

[9] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[10] J. Mendel and R. McLaren, "Reinforcement-learning control and pattern recognition systems," in *Mathematics in science and engineering*. Elsevier, 1970, vol. 66, pp. 287–318.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[12] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 178–192, May 2009.

[13] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[14] P. J. Werbos, *The Elements of Intelligence*. Cybernetica, 1968.

[15] ——, "Advanced forecasting methods for global crisis warning and models of intelligence," *General Systems Yearbook*, vol. 22, pp. 25–38, 1977.

[16] ——, "Approximate dynamic programming for real-time control and neural modeling," *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*, pp. 493—-525, 1992.

[17] A. G. Barto, "Reinforcement learning and adaptive critic methods," *Handbook of intelligent control: Neural, fuzzy, and adaptive approaches*, pp. 469—-491, 1992.

[18] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, Jan. 1995.

[19] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 5, pp. 834–846, 1983.

[20] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavior sciences," Ph.D. dissertation, Appl. Math. Harvard Univ., 1974.

[21] P. Werbos, "Neural networks for control and system identification," in *Proceedings of the 28th IEEE Conference on Decision and Control*. IEEE, 1989.

[22] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[23] R. J. Leake and R.-W. Liu, "Construction of suboptimal control sequences," *SIAM Journal on Control*, vol. 5, no. 1, pp. 54–63, Feb. 1967.

[24] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Transactions on Automatic Control*, vol. 16, no. 4, pp. 382–384, Aug. 1971.

[25] D. Kleinman, "On an iterative technique for riccati equation computations," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 114–115, Feb. 1968.

[26] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 997–1007, 1997.

[27] H. Zhang, D. Liu, Y. Luo, and D. Wang, *Adaptive dynamic programming for control: algorithms and stability*. Springer Science & Business Media, 2012.

[28] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[29] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[30] X. Yang, D. Liu, and Y. Huang, "Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints," *IET Control Theory & Applications*, vol. 7, no. 17, pp. 2037–2047, 2013.

[31] R. Song, F. L. Lewis, Q. Wei, and H. Zhang, "Off-policy actor-critic structure for optimal control of unknown systems with disturbances," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1041–1050, 2015.

[32] H. Modares, F. L. Lewis, and Z.-P. Jiang, "h_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2550–2562, 2015.

[33] T. Dierks, B. T. Thumati, and S. Jagannathan, "Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence," *Neural Networks*, vol. 22, no. 5-6, pp. 851–860, 2009.

[34] Y. Huang and D. Liu, "Neural-network-based optimal tracking control scheme for a class of unknown discrete-time nonlinear systems using iterative adp algorithm," *Neurocomputing*, vol. 125, pp. 46–56, 2014.

[35] H. Li and D. Liu, "Optimal control for discrete-time affine non-linear systems using general value iteration," *IET Control Theory & Applications*, vol. 6, no. 18, pp. 2725–2736, 2012.

[36] D. Liu, D. Wang, and X. Yang, "An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs," *Information Sciences*, vol. 220, pp. 331–342, 2013.

[37] R. A. Howard, "Dynamic programming and markov processes." 1960.

[38] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 32, no. 2, pp. 140–153, 2002.

[39] B. Luo, D. Liu, H.-N. Wu, D. Wang, and F. L. Lewis, "Policy gradient adaptive dynamic programming for data-based optimal control," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3341–3354, 2016.

[40] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive dynamic programming with applications in optimal control*. Springer, 2017.

[41] S. B. Thrun and K. Möller, "Active exploration in dynamic environments," in *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Morgan-Kaufmann, 1992, pp. 531–538.

[42] M. Fu and B. Barmish, "Adaptive stabilization of linear systems via switching control," *IEEE Transactions on Automatic Control*, vol. 31, no. 12, pp. 1097–1103, Dec. 1986.

[43] P. A. Ioannou and J. Sun, *Robust adaptive control*. Courier Corporation, 2012.

[44] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 916–932, May 2015.

[45] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 631–647, May 2007.

[46] X. Xu, C. Lian, L. Zuo, and H. He, "Kernel-based approximate dynamic programming for real-time online learning control: An experimental study," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 146–156, Jan. 2014.

[47] A. G. Khiabani and A. Heydari, "Optimal torque control of permanent magnet synchronous motors using adaptive dynamic programming," *IET Power Electronics*, vol. 13, no. 12, pp. 2442–2449, Sep. 2020.

[48] A. M. Dissanayake and N. C. Ekneligoda, "Droop free optimal feedback control of distributed generators in islanded DC microgrids," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, pp. 1–1, 2019.

[49] J. Na, Y. Lv, K. Zhang, and J. Zhao, "Adaptive identifier-critic-based optimal tracking control for nonlinear systems with experimental validation," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 1–14, 2020.

[50] G. Rizzello, D. Naso, B. Turchiano, and S. Seelecke, "Robust position control of dielectric elastomer actuators based on LMI optimization," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 1909–1921, Nov. 2016.

[51] F. Carpi, D. De Rossi, R. Kornbluh, R. E. Pelrine, and P. Sommer-Larsen, *Dielectric elastomers as electromechanical transducers: Fundamentals, materials, devices, models and applications of an emerging electroactive polymer technology*. Elsevier, 2011.

[52] T. Hoffstadt and J. Maas, "Adaptive sliding-mode position control for dielectric elastomer actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2241–2251, Oct. 2017.

[53] L.-L. Fan, V. Nasirian, H. Modares, F. L. Lewis, Y.-D. Song, and A. Davoudi, "Game-theoretic control of active loads in DC microgrids," *IEEE Transactions on Energy Conversion*, vol. 31, pp. 882–895, Sep. 2016.

[54] V. Nasirian, A. P. Yadav, F. L. Lewis, and A. Davoudi, "Distributed assistive control of power buffers in DC microgrids," *IEEE Transactions on Energy Conversion*, vol. 32, pp. 1396–1406, Dec. 2017.

[55] W. W. Weaver and P. T. Krein, "Mitigation of power system collapse through active dynamic buffers," in *Proceeding of the 35th IEEE Annual Power Electronics Specialists Conference*, 2004.

[56] X. Wang, D. Vilathgamuwa, and S. Choi, "Decoupling load and power system dynamics to improve system stability," in *Proceding of the 2005 International Conference on Power Electronics and Drives Systems*, 2005.

[57] B. Banerjee and W. W. Weaver, "Generalized geometric control manifolds of power converters in a DC microgrid," *IEEE Transactions on Energy Conversion*, vol. 29, pp. 904–912, Dec. 2014.

[58] F. Dorfler, M. R. Jovanovic, M. Chertkov, and F. Bullo, "Sparsity-promoting optimal wide-area control of power networks," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2281–2291, Sep. 2014.

[59] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[60] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, Apr. 2009.

[61] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized hamilton-jacobi-bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, Dec. 1997.

[62] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous-time adaptive critics," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 631–647, May 2007.

[63] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[64] H. Leiva and S. Siegmund, "A necessary algebraic condition for controllability and observability of linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2229–2232, Dec. 2003.

[65] Y. Jiang and Z.-P. Jiang, *Robust adaptive dynamic programming*. Hoboken: John Wiley & Sons, Inc., 2017.

[66] E. Najafi, R. Babuška, and G. A. D. Lopes, "A fast sampling method for estimating the domain of attraction," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 823–834, Jul. 2016.

[67] G. Yuan and Y. Li, "Estimation of the regions of attraction for autonomous nonlinear systems," *Transactions of the Institute of Measurement and Control*, vol. 41, no. 1, pp. 97–106, Mar. 2018.

[68] U. Topcu, A. Packard, P. Seiler, and G. Balas, "Robust region-of-attraction estimation," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 137–142, Jan. 2010.

[69] W. Levine and M. Athans, "On the determination of the optimal constant output feedback gains for linear multivariable systems," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 44–48, Feb. 1970.

[70] C. Wenk and C. Knapp, "Parameter optimization in linear systems with arbitrarily constrained controller structure," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 496–500, Jun. 1980.

[71] M. Fardad, F. Lin, and M. R. Jovanovic, "On the optimal design of structured feedback gains for interconnected systems," in *Proceedings of the 48h IEEE Conference on Decision and Control*. IEEE, 2009.

[72] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012.

[73] T. Bian and Z.-P. Jiang, "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design," *Automatica*, vol. 71, pp. 348–360, Sep. 2016.

[74] D. J. Murray-Smith, *Continuous system simulation*. Springer Science & Business Media, 2012.

[75] P. R. Massenio, G. Rizzello, D. Naso, F. L. Lewis, and A. Davoudi, "Data-driven optimal structured control for unknown symmetric systems," in *Proceedings of the 16th IEEE International Conference on Automation Science and Engineering*, 2020.

[76] A. O'Halloran, F. O'Malley, and P. McHugh, "A review on dielectric elastomer actuators, technology, applications, and challenges," *Journal of Applied Physics*, vol. 104, no. 7, 2008.

[77] G. Rizzello, P. Serafino, D. Naso, and S. Seelecke, "Towards sensorless soft robotics: Self-sensing stiffness control of dielectric elastomer actuators," *IEEE Transactions on Robotics*, pp. 1–15, 2019.

[78] S. Shian, K. Bertoldi, and D. R. Clarke, "Dielectric elastomer based "grippers" for soft robotics," *Advanced Materials*, vol. 27, no. 43, pp. 6814–6819, Sep. 2015.

[79] J. Shintake, S. Rosset, B. Schubert, D. Floreano, and H. Shea, "Versatile soft grippers with intrinsic electroadhesion based on multifunctional polymer actuators," *Advanced Materials*, vol. 28, no. 2, pp. 231–238, Nov. 2015.

[80] G. Moretti, M. S. Herran, D. Forehand, M. Alves, H. Jeffrey, R. Vertechy, and M. Fontana, "Advances in the development of dielectric elastomer generators for wave energy conversion," *Renewable and Sustainable Energy Reviews*, vol. 117, p. 109430, Jan. 2020.

[81] P. Lotz, M. Matysek, and H. F. Schlaak, "Fabrication and application of miniaturized dielectric elastomer stack actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 1, pp. 58–66, Feb. 2011.

[82] F. Carpi, C. Menon, and D. D. Rossi, "Electroactive elastomeric actuator for all-polymer linear peristaltic pumps," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 3, pp. 460–470, Jun. 2010.

[83] M. Giousouf and G. Kovacs, "Dielectric elastomer actuators used for pneumatic valve technology," *Smart Materials and Structures*, vol. 22, no. 10, p. 104010, Sep. 2013.

[84] E. Biddiss and T. Chau, "Dielectric elastomers as actuators for upper limb prosthetics: Challenges and opportunities," *Medical Engineering & Physics*, vol. 30, no. 4, pp. 403–418, May 2008.

[85] G. Jordan, D. N. McCarthy, N. N. Schlepple, J. Krissler, H. Schröder, and G. Kofod, "Actuated micro-optical submount using a dielectric elastomer actuator," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 1, pp. 98–102, Feb. 2011.

[86] G.-Y. Gu, J. Zhu, L.-M. Zhu, and X. Zhu, "A survey on dielectric elastomer actuators for soft robots," *Bioinspiration & Biomimetics*, vol. 12, no. 1, p. 011003, Jan. 2017.

[87] C. T. Nguyen, H. Phung, T. D. Nguyen, H. Jung, and H. R. Choi, "Multiple-degrees-of-freedom dielectric elastomer actuators for soft printable hexapod robot," *Sensors and Actuators A: Physical*, vol. 267, pp. 505–516, Nov. 2017.

[88] G.-Y. Gu, U. Gupta, J. Zhu, L.-M. Zhu, and X. Zhu, "Modeling of viscoelastic electromechanical behavior in a soft dielectric elastomer actuator," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1263–1271, Oct. 2017.

[89] A. York, J. Dunn, and S. Seelecke, "Experimental characterization of the hysteretic and rate-dependent electromechanical behavior of dielectric electro-active polymer actuators," *Smart Materials and Structures*, vol. 19, no. 9, p. 094014, aug 2010.

[90] F. Chen, K. Liu, Y. Wang, J. Zou, G. Gu, and X. Zhu, "Automatic design of soft dielectric elastomer actuators with optimal spatial electric fields," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1150–1165, Oct. 2019.

[91] W. Kaal and S. Herold, "Electroactive polymer actuators in dynamic applications," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 1, pp. 24–32, Feb. 2011.

[92] G.-Y. Gu, U. Gupta, J. Zhu, L.-M. Zhu, and X.-Y. Zhu, "Feedforward deformation control of a dielectric elastomer actuator based on a nonlinear dynamic model," *Applied Physics Letters*, vol. 107, no. 4, p. 042907, Jul. 2015.

[93] R. Sarban and R. W. Jones, "Physical model-based active vibration control using a dielectric elastomer actuator," *Journal of Intelligent Material Systems and Structures*, vol. 23, no. 4, pp. 473–483, Feb. 2012.

[94] G. Rizzello, D. Naso, A. York, and S. Seelecke, "Modeling, identification, and control of a dielectric electro-active polymer positioning system," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 632–643, Mar. 2015.

[95] E. D. Wilson, T. Assaf, M. J. Pearson, J. M. Rossiter, S. R. Anderson, J. Porrill, and P. Dean, "Cerebellar-inspired algorithm for adaptive control of nonlinear dielectric elastomer-based artificial muscle," *Journal of The Royal Society Interface*, vol. 13, no. 122, p. 20160547, Sep. 2016.

[96] W. Liang, J. Cao, Q. Ren, and J.-X. Xu, "Control of dielectric elastomer soft actuators using antagonistic pairs," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2862–2872, Dec. 2019.

[97] J. Zou and G. Gu, "High-precision tracking control of a soft dielectric elastomer actuator with inverse viscoelastic hysteresis compensation," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 36–44, Feb. 2019.

[98] J. Zou and G. Gu, "Feedforward control of the rate-dependent viscoelastic hysteresis nonlinearity in dielectric elastomer actuators," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2340–2347, Jul. 2019.

[99] B. N. M. Truong and K. K. Ahn, "Inverse modeling and control of a dielectric electro-active polymer smart actuator," *Sensors and Actuators A: Physical*, vol. 229, pp. 118–127, Jun. 2015.

[100] G. Rizzello, D. Naso, and S. Seelecke, "A thermodynamically consistent port-hamiltonian model for dielectric elastomer membrane actuators and generators," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4855–4862, Jul. 2017.

[101] J. A. Stratton, *Electromagnetic Theory*. London, U.K.: McGraw-Hill, 1941.

[102] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 3.

[103] F. Renaud, J.-L. Dion, G. Chevallier, I. Tawfiq, and R. Lemaire, "A new identification method of viscoelastic behavior: Application to the generalized maxwell model," *Mechanical Systems and Signal Processing*, vol. 25, no. 3, pp. 991–1010, 2011.

[104] P. R. Massenio, G. Rizzello, and D. Naso, "Fuzzy adaptive dynamic programming minimum energy control of dielectric elastomer actuators," in *Proceeding of the 2019 IEEE International Conference on Fuzzy Systems*. IEEE, 2019.

[105] P. R. Massenio, D. Naso, and G. Rizzello, "Energy optimal control of dielectric elastomer actuators via adaptive dynamic programming," in *Proocedings of the 15th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*. ASME, 2019.

[106] P. R. Massenio, G. Rizzello, G. Comitangelo, D. Naso, and S. Seelecke, "Reinforcement learning-based minimum energy position control of dielectric elastomer actuators," *IEEE Transactions on Control Systems Technology*, pp. 1–15, 2020.

[107] A. T. Elsayed, A. A. Mohamed, and O. A. Mohammed, "DC microgrids and distribution systems: An overview," *Electric Power Systems Research*, vol. 119, pp. 407–417, Feb. 2015.

[108] Z. Wang, F. Liu, Y. Chen, S. H. Low, and S. Mei, "Unified distributed control of stand-alone DC microgrids," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 1013–1024, Jan. 2019.

[109] T. Dragicevic, X. Lu, J. Vasquez, and J. Guerrero, "DC microgrids–part i: A review of control strategies and stabilization techniques," *IEEE Transactions on Power Electronics*, pp. 1–1, 2015.

[110] T. Dragicevic, X. Lu, J. C. Vasquez, and J. M. Guerrero, "DC microgrids—part II: A review of power architectures, applications, and standardization issues," *IEEE Transactions on Power Electronics*, vol. 31, no. 5, pp. 3528–3549, May 2016.

[111] E. Planas, J. Andreu, J. I. Gárate, I. M. de Alegría, and E. Ibarra, "AC and DC technology in microgrids: A review," *Renewable and Sustainable Energy Reviews*, vol. 43, pp. 726–749, Mar. 2015.

[112] J. J. Justo, F. Mwasilu, J. Lee, and J.-W. Jung, "AC-microgrids versus DC-microgrids with distributed energy resources: A review," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 387–405, Aug. 2013.

[113] V. Nasirian, A. Davoudi, F. L. Lewis, and J. M. Guerrero, "Distributed adaptive droop control for DC distribution systems," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 944–956, Dec. 2014.

[114] M. Hamzeh, M. Ghafouri, H. Karimi, K. Sheshyekani, and J. M. Guerrero, "Power oscillations damping in DC microgrids," *IEEE Transactions on Energy Conversion*, vol. 31, pp. 970–980, Sep. 2016.

[115] S. Sanchez and M. Molinas, "Degree of influence of system states transition on the stability of a DC microgrid," *IEEE Transactions on Smart Grid*, vol. 5, pp. 2535–2542, Sep. 2014.

[116] M. Patterson, N. F. Macia, and A. M. Kannan, "Hybrid microgrid model based on solar photovoltaic battery fuel cell system for intermittent load applications," *IEEE Transactions on Energy Conversion*, vol. 30, no. 1, pp. 359–366, Mar. 2015.

[117] N. R. Tummuru, M. K. Mishra, and S. Srinivas, "Dynamic energy management of hybrid energy storage system with high-gain PV converter," *IEEE Transactions on Energy Conversion*, vol. 30, no. 1, pp. 150–160, Mar. 2015.

[118] M. Kim and A. Kwasinski, "Decentralized hierarchical control of active power distribution nodes," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 934–943, Dec. 2014.

[119] S. Kazemlou and S. Mehraeen, "Novel decentralized control of power systems with penetration of renewable energy sources in small-scale power systems," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 851–861, Dec. 2014.

[120] D. Logue and P. Krein, "The power buffer concept for utility load decoupling," in *Proccedings of the 31st IEEE Annual Power Electronics Specialists Conference*. IEEE, 2000.

[121] ——, "Preventing instability in DC distribution systems by using power buffering," in *Proccedings of the 32nd IEEE Annual Power Electronics Specialists Conference*. IEEE, 2001.

[122] M. Vasiladiotis and A. Rufer, "A modular multiport power electronic transformer with integrated split battery energy storage for versatile ultrafast EV charging stations," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 3213–3222, May 2015.

[123] W. Weaver and P. Krein, "Game-theoretic control of small-scale power systems," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1560–1567, Jul. 2009.

[124] L.-L. Fan, V. Nasirian, H. Modares, F. L. Lewis, Y.-D. Song, and A. Davoudi, "Game-theoretic control of active loads in DC microgrids," *IEEE Transactions on Energy Conversion*, vol. 31, no. 3, pp. 882–895, Sep. 2016.

[125] W. W. Weaver, "Dynamic energy resource control of power electronics in local area power networks," *IEEE Transactions on Power Electronics*, vol. 26, no. 3, pp. 852–859, Mar. 2011.

[126] N. C. Ekneligoda and W. W. Weaver, "Game-theoretic cold-start transient optimization in DC microgrids," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 12, pp. 6681–6690, Dec. 2014.

[127] B. Banerjee and W. W. Weaver, "Generalized geometric control manifolds of power converters in a DC microgrid," *IEEE Transactions on Energy Conversion*, vol. 29, no. 4, pp. 904–912, Dec. 2014.

[128] A. M. Dissanayake and N. C. Ekneligoda, "Online game theoretic feedback control of DC microgrids," in *Proccedings of the 2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference*. IEEE, 2018.

[129] N. C. Ekneligoda and W. W. Weaver, "Game-theoretic communication structures in microgrids," *IEEE Transactions on Power Delivery*, vol. 27, no. 4, pp. 2334–2341, Oct. 2012.

[130] V. Nasirian, A. P. Yadav, F. L. Lewis, and A. Davoudi, "Distributed assistive control of power buffers in DC microgrids," *IEEE Transactions on Energy Conversion*, vol. 32, no. 4, pp. 1396–1406, Dec. 2017.

[131] Y. Kuriki and T. Namerikawa, "Experimental validation of cooperative formation control with collision avoidance for a multi-UAV system," in *Proceedings of the 6th IEEE Conference on Automation, Robotics and Applications*. IEEE, 2015.

[132] J. Hu and Z. Xu, "Distributed cooperative control for deployment and task allocation of unmanned aerial vehicle networks," *IET Control Theory & Applications*, vol. 7, no. 11, pp. 1574–1582, 2013.

[133] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao, "Cooperative motion generation in a distributed network of redundant robot manipulators with noises," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 48, no. 10, pp. 1715–1724, 2018.

[134] S. Li, H. Du, and P. Shi, "Distributed attitude control for multiple spacecraft with communication delays," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 1765–1773, 2014.

[135] V. Nasirian, S. Moayedi, A. Davoudi, and F. L. Lewis, "Distributed cooperative control of DC microgrids," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 2288–2303, 2015.

[136] Q. Wei, F. L. Lewis, G. Shi, and R. Song, "Error-tolerant iterative adaptive dynamic programming for optimal renewable home energy scheduling and battery management," *IEEE Transctions on Industrial Electronics*, vol. 64, no. 12, pp. 9527–9537, Dec. 2017.

[137] G. K. Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic energy management system for a smart microgrid," *IEEE Transactions on Neural Networks and Learning Sysems*, vol. 27, no. 8, pp. 1643–1656, Aug. 2016.

[138] M. Boaro, D. Fuselli, F. D. Angelis, D. Liu, Q. Wei, and F. Piazza, "Adaptive dynamic programming algorithm for renewable energy scheduling and battery management," *Cognitive Computing*, vol. 5, no. 2, pp. 264–277, Sep. 2012.

[139] D. Fuselli, F. D. Angelis, M. Boaro, S. Squartini, Q. Wei, D. Liu, and F. Piazza, "Action dependent heuristic dynamic programming for home energy resource scheduling," *International Journal of Electrical Power & Energy Systems*, vol. 48, pp. 148–160, Jun. 2013.

[140] S. Xie, W. Zhong, K. Xie, R. Yu, and Y. Zhang, "Fair energy scheduling for vehicle-to-grid networks using adaptive dynamic programming," *IEEE Transactions on Neural Networks and Learning Sysems*, vol. 27, no. 8, pp. 1697–1707, Aug. 2016.

[141] Q. Wei, G. Shi, R. Song, and Y. Liu, "Adaptive dynamic programming-based optimal control scheme for energy storage systems with solar renewable energy," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5468–5478, Jul. 2017.

[142] Y. Tang, H. He, J. Wen, and J. Liu, "Power system stability control for a wind farm based on adaptive dynamic programming," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 166–177, Jan. 2015.

[143] T. Bian, Y. Jiang, and Z.-P. Jiang, "Decentralized adaptive optimal control of large-scale systems with application to power systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2439–2447, Apr. 2015.

[144] A. M. Dissanayake and N. C. Ekneligoda, "Droop free optimal feedback control of distributed generators in islanded DC microgrids," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 2019.

[145] F. Lin, M. Fardad, and M. R. Jovanovic, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, Sep. 2013.

[146] S. Schuler, P. Li, J. Lam, and F. Allgöwer, "Design of structured dynamic output-feedback controllers for interconnected systems," *Internation Journal of Control*, vol. 84, no. 12, pp. 2081–2091, Dec. 2011.

[147] S. Schuler, U. Münz, and F. Allgöwer, "Decentralized state feedback control for interconnected systems with application to power systems," *Journal of Process Control*, vol. 24, no. 2, pp. 379–388, Feb. 2014.

[148] Y. Tian and J. A. Taylor, "Sparsity-promoting controller design for VSC-based microgrids," in *Proceedings of the 2016 IEEE Global Conference on Signal and Information Processing*. IEEE, 2016.

[149] X. Wu, F. Dorfler, and M. R. Jovanovic, "Input-output analysis and decentralized optimal control of inter-area oscillations in power systems," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2434–2444, May 2016.

[150] A. Al-Digs, S. V. Dhople, and Y. C. Chen, "Measurement-based sparsity-promoting optimal control of line flows," *IEEE Transction on Power Systems*, vol. 33, no. 5, pp. 5628–5638, Sep. 2018.

[151] A. Jain, A. Chakrabortty, and E. Biyik, "An online structurally constrained LQR design for damping oscillations in power system networks," in *Proceedings of the 2017 IEEE American Control Conference*. IEEE, 2017.

[152] N. Gaeini, A. M. Amani, M. Jalili, and X. Yu, "Optimization of communication network topology in distributed control systems subject to prescribed decay rate," *IEEE Transactions on Cybernetics*, pp. 1–9, 2019.

[153] F. Glover, "Tabu search—part i," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[154] ——, "Tabu search—part ii," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

[155] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, pp. 150–163, Jan. 2013.

[156] D. Xu, M. Chiang, and J. Rexford, "Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 1717–1730, Dec. 2011.

[157] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine*, vol. 19, pp. 59–70, Oct. 1999.

[158] M. Collotta and G. Pau, "A novel energy management approach for smart homes using bluetooth low energy," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2988–2996, Dec. 2015.

[159] C. J. Hansen, "Internetworking with bluetooth low energy," *ACM GetMobile: Mobile Computing and Communications*, vol. 19, pp. 34–38, Aug. 2015.

[160] R. Rondón, M. Gidlund, and K. Landernäs, "Evaluating bluetooth low energy suitability for time-critical industrial IoT applications," *International Journal of Wireless Information Networks*, vol. 24, pp. 278–290, May 2017.

[161] T. K. Refaat, R. M. Daoud, H. H. Amer, and E. A. Makled, "WiFi implementation of wireless networked control systems," in *Proceedings of the 7th ASM Inernational Conference on Networked Sensing Systems*. ASM, 2010.

[162] P. R. Massenio, D. Naso, F. L. Lewis, and A. Davoudi, "Assistive power buffer control via adaptive dynamic programming," *IEEE Transactions on Energy Conversion*, vol. 35, no. 3, pp. 1534–1546, Sep. 2020.

[163] P. R. Massenio, D. Naso, F. L. Lewis, and A. Davoudi, "Data-driven sparsity-promoting optimal control of power buffers in dc microgrids," *IEEE Transactions on Energy Conversion*, pp. 1–1, 2020.