

Reinforcement Learning Algorithms with Safe Operation Guarantees for Complex Nonlinear Dynamical Systems with Unknown Models

PHD Thesis – Summary

for obtaining the scientific title of Doctor

Universitatea Politehnica Timișoara

in doctoral field of Systems Engineering

Author: Timotei LALA

Scientific coordinator: Prof. Univ. Dr. Ing. Ioan SILEA

May 2026

The contemporary economy relies on complex dynamic systems (critical infrastructures, industrial processes, robotic systems) governed by closed-loop automatic control mechanisms. Stability and Safe Operation (SO) are fundamental, as failures can have catastrophic consequences.

Recently, controller tuning methods based on **Reinforcement Learning** (RL) or **Adaptive Dynamic Programming** (ADP) have been developed, which learn from historical data without requiring explicit mathematical models. RL combines exploration with exploitation. Advances in deep learning and neural networks have enabled remarkable successes in simulated systems, games (AlphaZero, AlphaGO, Agent57), recommendation systems, and even large language models. However, the transition to real-world applications creates a fundamental tension: learning requires exploration, while safe operation limits it. In real control systems, unlike simulated ones, failures are irreversible and unacceptable. The purpose of the exploration process is to introduce random sequential commands at the control input of the controlled system and to observe their effects on the entire control loop. However, unconstrained exploration can violate SO, while excessive restriction prevents obtaining optimal controllers.

At the same time, beyond maintaining SO during exploration, there is a risk that the learned controllers may be destabilizing due to approximation errors, insufficient exploration, or incorrect hyperparameters. The central challenge is guaranteeing SO both during exploration and during testing of the learned controllers — the starting point for the contributions of this thesis.

To address the limitations presented above, the subfield of Safe Reinforcement Learning (Safe RL) has been developed, which introduces a series of algorithms that ensure system operation in SO throughout the learning process. Unlike classical RL algorithms, Safe RL treats safety as an explicit requirement, formulated as a constraint on the system's states, commands, or trajectories.

The existing literature can be organized into several research directions. A first class of algorithms is based on constrained optimization problem formulations, in which the objective function is augmented with inequality-type expressions that penalize violations of safety conditions. These approaches provide an elegant formal mathematical framework, using classical optimization methods such as the Lagrange multiplier method or primal-dual optimization within constrained Markov decision processes. In [6], a constrained optimization method was developed to learn a controller that guarantees approximate constraint satisfaction from each learning iteration, based on a method derived from TRPO. In [7], a method was developed to incorporate the constraints of constrained Markov decision processes into the reward function, developing a general method for solving these problems. Although these methods provide an elegant formal mathematical framework, they do not guarantee the absence of occasional constraint violations during learning, which is unacceptable for many real-world systems.

The second class of methods is represented by filtering algorithms for potentially destabilizing commands, known as "safety layers." These methods are exogenous to RL optimization, acting at the output of the controller as command filters based on stability certificates. Among these methods, algorithms based on Control Barrier Functions (CBF) stand out. During the execution of the controlled system, it is necessary for the operating regime to remain within the predefined CBF in order to guarantee operation in the Safe Region (SR). In [8], the authors present a hybrid controller that combines model-free RL and the CBF algorithm with the mathematical model learned from the system, using the Trust Region Policy Optimization (TRPO) method for adaptive controller updates and Gaussian processes for dynamic system modeling, ensuring SO through real-time quadratic optimization. Furthermore, the use of learning certificates allows modification of controller commands to guarantee system operation in SR, as presented in [9]-[11]. However, these generate conservative behavior during execution, being dependent on the quality of the identified model or the certificate learned from data.

The third class of methods employs concepts from systems theory such as control Lyapunov functions or invariant sets, to provide formal safety guarantees. In [12], the authors used a Lyapunov stability verification method and statistical models of the dynamics to achieve high-performance control. Based on constrained Markov decision processes, the method described in [13] guarantees SO by constructing local linear constraints based on Lyapunov functions. This class of algorithms, however, requires either precise knowledge of the system's dynamic equations or an initial nominal model of the system that is improved using data.

Although the field has seen significant progress, existing algorithms present multiple limitations. Many methods require knowledge of the system's dynamic equations in order to predict future evolution and select commands that keep the system in the safe operating regime (SR). However, obtaining these equations for complex nonlinear systems is difficult. Some approaches do not guarantee closed-loop stability or SO during exploration, but only at convergence. Other algorithms are limited to certain classes of systems and are not applicable to general continuous nonlinear systems. Most methods do not provide theoretical stability or safety guarantees, being presented only as heuristic methods.

These limitations highlight the need for RL algorithms for general nonlinear systems with unknown models, which provide explicit safety guarantees both during exploration and during evaluation of the learned controllers.

Research Objectives. Based on the analysis of the scientific literature mentioned above, this thesis aims to methodically analyze the problem of maintaining closed-loop system stability with the controller tuned using RL and its SO, with the goal of improving current methods and solving existing problems in the field. Specifically, this doctoral thesis proposes the following research objectives (RO):

RO1. Analysis of the specialized literature in the field of Safe RL (SRL) with the aim of identifying the limitations of existing methods for maintaining controlled systems within SO.

Școala Doctorală de Studii Inginerești

RO2. Development of theoretical convergence guarantees for RL algorithms, in order to demonstrate the ability of the learning process to obtain the optimal controller and cost function.

RO3. Development of a method for maintaining the stability and SO of a system with an unknown model in closed loop with a controller learned using RL algorithms.

RO4. Development of a method for maintaining the stability and SO of the system during the exploration of the state-action domain.

RO5. Practical validation of these algorithms on multiple classes of nonlinear systems, in order to demonstrate the applicability of the method across multiple fields of activity.

The first research objective aims at a comprehensive understanding of the Safe RL (SRL) field, drawing from both the specialized literature in Artificial Intelligence and that in automatic control. Due to the fact that the AI literature addresses stochastic controllers and systems of the Markov Decision Process type, it has developed in parallel with the automatic control literature, which deals with deterministic controllers and systems. Furthermore, since SRL methods were initially developed within the AI literature, the goal pursued at this stage is to understand and analyze the developed methods, both with a view to adapting them to the control domain and to finding new solutions to unresolved problems. The second RO of this thesis pursues the development of theoretical convergence guarantees for RL-type learning algorithms. Convergence guarantees are essential for ensuring predictable and deterministic behavior of the algorithms — a critical aspect in applications with SO constraints — while also providing a solid theoretical foundation for the analysis and comparison of various algorithms. The third objective addresses the problem caused by an erroneous learning process, which may generate controllers that either destabilize the system or drive it out of SO. It is therefore necessary to develop a method for maintaining the stability and SO of a system with an unknown model in closed loop with the controllers obtained during the iterations of the RL learning process. The next objective of this doctoral thesis is the maintenance of stability and SO of the system during the exploration of the state-action domain. This need arises from the unpredictability and complexity that the additive exploratory noise at the command input introduces into the control loop, which may result in destabilizing the system and simultaneously driving it out of SO. The final objective addresses the validation of all previously developed methods for achieving the preceding objectives. Validation must be carried out on multiple systems from disjoint fields of activity, in order to validate the generalization of these methods to a broader class of nonlinear dynamics.

Major Contributions. This doctoral thesis systematically proposes multiple methods for maintaining the stability of the closed-loop system with the controller tuned using RL and its operation within a Safe Operation (SO) regime. By addressing this problem, the thesis aims to contribute to the development of RL methods that are both theoretically grounded and applicable in real-world contexts, where safety cannot be compromised at any moment. The details of these major contributions (MC), also presented in the author's specialized publications, are described below:

MC1. Convergence analysis of the Batch Offline Action-Dependent Heuristic Dynamic Programming (BOADHDP) algorithm for multilayer neural network approximators.

MC2. Development of a method for maintaining SO of a system with an unknown model, controlled by a controller tuned using the Constrained Adaptive Dynamic Programming with Deterministic Policy Gradient (CADPDGP) method.

MC3. Development of a method for maintaining the stability and SO of a system with an unknown model, using an initial stabilizing controller in switching with the ADP-tuned controller, referred to as Safe Switching Model-Free Value Iteration (SSMFVI).

MC4. Development of a method for maintaining SO during the exploration process, resulting in the Iterative Model-Free Safe Exploration (IMFSE) algorithm.

The thesis is structured into 8 chapters that successively address the stages of approach for the major contributions presented above. Chapter 1 presents the introduction, motivation, research objectives, and the main development directions of this doctoral thesis. Chapter 2 introduces fundamental theoretical aspects of RL algorithms, such as the Value Iteration (VI) algorithm and the Policy Iteration (PI) algorithm, their implementation in Q-functions, and the convergence theory of the VI algorithm — both for value functions and Q-functions, with and without function approximators. Chapter 3 analyzes the theoretical stability of the BOADHDP algorithm using Uniform Asymptotic Stability (UAS) convergence analysis with Lyapunov functions. The performance of the BO-ADHDP algorithm when using multilayer neural networks compared to single-layer neural networks is validated on a Twin Rotor Aerodynamic System (TRAS). Chapter 4 introduces the implementation of the CADPDPG algorithm with function approximators, along with its convergence analysis. The practical implementation is carried out with neural network function approximators, and UAS convergence is established using Lyapunov functions. The algorithm is validated on a wind turbine process and an active suspension system, demonstrating the algorithm's ability to learn an optimal controller and to constrain the generated commands within a predefined interval. Chapter 5 applies the CADPDPG method for tuning general linear controllers, with validation performed on a third-order nonlinear system. Chapter 6 develops the SSMFVI algorithm, along with the Lyapunov stability analysis and the early instability detection mechanism. Validation of this algorithm is carried out on a linear system and on the TRAS using the switching mechanism. Chapter 7 describes the MFISE algorithm through the development of the closed-loop stability maintenance mechanism for the controller perturbed with finite-magnitude random noise, and the exploration mechanism with SO maintenance. Validation of this algorithm is performed on the TRAS. Chapter 8 presents the conclusions and future research directions.

The following paragraphs synthesize the main contributions and ideas of the thesis from Chapters 2–6, and present the innovations brought to the field of Safe RL (SRL). Chapter 2 describes the discrete-time nonlinear system, described by the state equation:

$$\begin{cases} \xi_{k+1} = F(\xi_k, \mathbf{u}_k), \\ \mathbf{z}_k = g(\xi_k), \end{cases} \quad (1)$$

where $k \in \mathbb{N}$ represents the time index, $\xi_k = [\xi_{1,k}, \dots, \xi_{n,k}]^T \in \Omega_\xi \subset \mathbb{R}^n$ is the state vector of the system, $\mathbf{u}_k = [u_{1,k}, \dots, u_{m,k}]^T \in \Omega_U \subset \mathbb{R}^m$ is the system input vector, $\mathbf{z}_k = [z_{1,k}, \dots, z_{p,k}]^T \in \Omega_Y \subset \mathbb{R}^p$ is the system output, $F: \Omega_\xi \times \Omega_U \rightarrow \Omega_\xi$ and $g: \Omega_\xi \rightarrow \Omega_Y$ unknown differentiable continuous system functions, where Ω_ξ , Ω_U and Ω_Y represents compact subsets of $\mathbb{R}^n$, $\mathbb{R}^m$ and $\mathbb{R}^p$. The control input is generated by $\mathbf{u}_k = C(\xi_k)$, where $C: \Omega_\xi \rightarrow \Omega_U$ is a continuous invariant vectorial function w.r.t. the state ξ . The remainder of this chapter describes the VI and PI algorithms, as well as their Q-function variants.

The algorithm analyzed in MC1 is derived from the ADP algorithm, using neural network approximators that are trained offline. To prove the convergence of the algorithm, Uniformly Ultimately Bounded (UUB) stability of the learning process is established using a Lyapunov-based approach. This proof extends the convergence analyses previously developed for ADP-type algorithms that use single-layer neural networks as function approximators and where learning is performed using one state transition at a time, generalizing the theory to $\$L_ \{Q|C\}$ hidden layers of the neural networks associated with the Q-function and the controller, and to $\$M\$$ state transitions in the dataset.

Chapter 3 presents the BOADHDP algorithm, which uses neural networks trained offline to approximate the cost function and the controller. For batch learning, all state transitions $(\xi_k, \mathbf{u}_k, \xi_{k+1})$ stored in the database $DTB_M = \{(\xi_k, \mathbf{u}_k, \xi_{k+1})\}$, with $k = 1:M$, collected during the exploration process. Algorithm 3.1 describes the steps of the

BOADHDP algorithm, which uses the gradient search method to update the weights of the multilayer neural networks associated with the state-action cost function (SACF) and the controller. For the convergence proof of this algorithm, Section 3.2 describes the UUB convergence analysis using the Lyapunov method. This proof extends the convergence analyses previously developed for ADP-type algorithms that use single-layer neural networks as function approximators and where learning is performed using one state transition at a time, generalizing the theory to $L_{Q|C}$ hidden layers of the neural networks associated with the Q-function and the controller, and to M state transitions in the dataset. The necessity of this theoretical contribution is based on the increasing complexity of RL algorithms in recent years, which employ various multilayer neural network architectures and more complex learning methodologies, based either on batches or mini-batches. Therefore, the existence of a general convergence theory for the BOADHDP algorithm is imperative for guaranteeing the operational behavior of the algorithm.

Based on 4 Lemmas (Lemma 3.1–3.4) that describe the differences in Lyapunov functions between two iterative learning steps, the main Theorem 3.2 is then formulated, which describes the convergence of the BOADHDP learning process conditioned on the value of the learning rate.

$$\alpha_Q < \frac{2(\bar{\theta}_{Q,max}^2 \pi_{Q,max}^2 + \sum_{l_Q=1}^{L_Q-1} \bar{\theta}_{Q,max} \pi_{Q,max}^2 \bar{\theta}_{Q,max} \prod_{l=1}^{l_Q-1} \bar{\theta}_{Q,max} \pi_{Q,max})}{\bar{\theta}_{Q,max}^2 \pi_{Q,max}^4 + \sum_{l_Q=1}^{L_Q-1} \bar{\theta}_{Q,max}^2 \pi_{Q,max}^4 \prod_{l=1}^{l_Q-1} \bar{\theta}_{Q,max}^2 \pi_{Q,max}^2} = \alpha_{Q,max}, \quad (2)$$

$$\alpha_C < \frac{2(\bar{\theta}_{C,max} \Omega_{max} \pi_{max} + \sum_{l_C=1}^{L_C-1} \bar{\theta}_{C,max} \phi_{max} \Omega_{max} \prod_{l=1}^{l_C-1} \bar{\theta}_{C,max} \pi_{C,max})}{\Omega_{max}^2 \pi_{max}^2 + \sum_{l_C=1}^{L_C-1} \pi_{max}^2 \Omega_{max}^2 \prod_{l=1}^{l_C-1} \bar{\theta}_{C,max}^2 \pi_{C,max}^2} = \alpha_{C,max}. \quad (3)$$

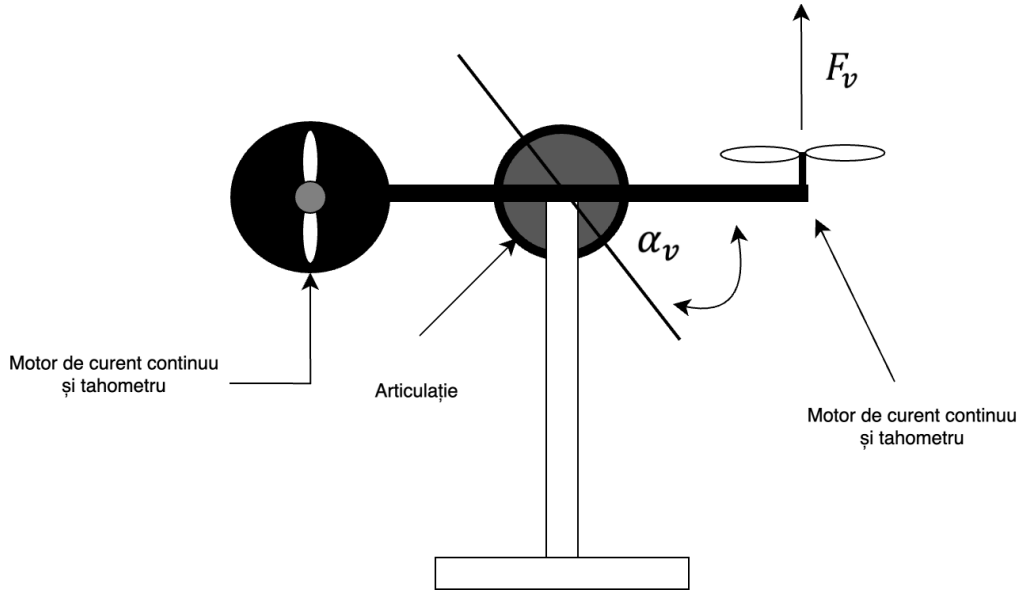


Figure 1. TRAS system.

For the validation of the superiority of multilayer neural networks compared to single-layer neural networks, the BO-ADHDP algorithm was applied to the TRAS process, which represents a nonlinear MIMO (Multiple Input Multiple Output) process with 6 states, two control inputs, and two outputs. This validation demonstrated that, in the case of multilayer neural networks, the convergence of the learning process is faster, and the control performance on the test scenario is likewise superior. This method was published in [14], fulfilling MC1.

The CDPGADP method described in Chapter 4 has as its main objective the constraining of the commands generated by the controller tuned between two learning iterations of the Adaptive Dynamic Programming with

Deterministic Policy Gradient (DPGADP) method. Starting from the stochastic controller analogue of this method [19]-[21], this algorithm extends the methods described in and by adding a constraint that limits the command outputs of the tuned controller between two successive learning iterations to a certain interval defined by $\boldsymbol{\vartheta}$. This method stems from the practical observation that, during the execution of RL-type learning algorithms, the performance of the tuned controller may vary drastically, with the possibility of obtaining a destabilizing controller in a single learning iteration. This is due to several factors, such as the approximation errors inherent to neural networks, inefficient exploration of the state-action domain, noise on the signals collected during exploration that corrupts the database used for tuning the approximators, or even the initialization of the algorithm with incorrect learning hyperparameters. Thus, starting from an initial stabilizing controller and limiting the difference between the commands of the controller tuned in a single learning iteration to an interval $\boldsymbol{\vartheta}$, the impact on the controlled system of a destabilizing controller obtained at any given iteration is restricted. Using this algorithm, the guarantee of SO is only circumstantial to the constrained learning process, as there are no constraints associated with the system states.

The neural network update associated with the controller is performed via projected gradient search, taking the following form:

$$\hat{C}(\xi_k, \hat{\boldsymbol{\theta}}_c^{1,j,i_c+1}, \hat{\boldsymbol{\theta}}_c^{2,j,i_c+1}) = \hat{C}(\xi_k, \hat{\boldsymbol{\theta}}_c^{1,j,i_c}, \hat{\boldsymbol{\theta}}_c^{2,j,i_c}) - \alpha \frac{\partial \hat{Q}(\xi_k, \mathbf{u}, \hat{\boldsymbol{\theta}}_q^{1,j+1}, \hat{\boldsymbol{\theta}}_q^{2,j+1})}{\partial \hat{C}(\xi_k, \hat{\boldsymbol{\theta}}_c^{1,j,i_c}, \hat{\boldsymbol{\theta}}_c^{2,j,i_c})} \Big|_{\mathbf{u}=\hat{C}(\cdot, \hat{\boldsymbol{\theta}}_c^{1,j,i_c}, \hat{\boldsymbol{\theta}}_c^{2,j,i_c})}, \quad (4)$$

$$s.t. |\hat{C}(\xi_k, \hat{\boldsymbol{\theta}}_c^{1,j+1}, \hat{\boldsymbol{\theta}}_c^{2,j+1}) - \hat{C}(\xi_k, \hat{\boldsymbol{\theta}}_c^{1,j}, \hat{\boldsymbol{\theta}}_c^{2,j})| \leq \boldsymbol{\vartheta}.$$

The main difference between DPGADP and CDPGADP lies in the way the gradient signal of the Q-function neural network is backpropagated into the controller neural network. For the DPGADP algorithm, the gradient of the Q-function neural network with respect to the controller's control input, $\partial E_{c,i}^j / \partial \hat{C}(\Xi_p, \hat{\boldsymbol{\theta}}_c^{1,j}, \hat{\boldsymbol{\theta}}_c^{2,j})$, is computed and then multiplied by the gradient of the controller output with respect to its weights, $\partial \hat{C}(\Xi_p, \hat{\boldsymbol{\theta}}_c^{1,j}, \hat{\boldsymbol{\theta}}_c^{2,j}) / \partial \hat{\boldsymbol{\theta}}_c^{1,j}$. For the controller update using CDPGADP, the gradient of the Q-function neural network with respect to the control input, ω_k^{j,i_c} , first updates the controller target, which is then used to update the controller neural network weights via the backpropagation algorithm. Thus, the gradient of the Q-function with respect to the input is propagated indirectly to the controller neural network weights through the targets, differing from the direct gradient information propagation as in the case of DPGADP. Fig. 2 illustrates this fundamental architectural difference between the two methods.

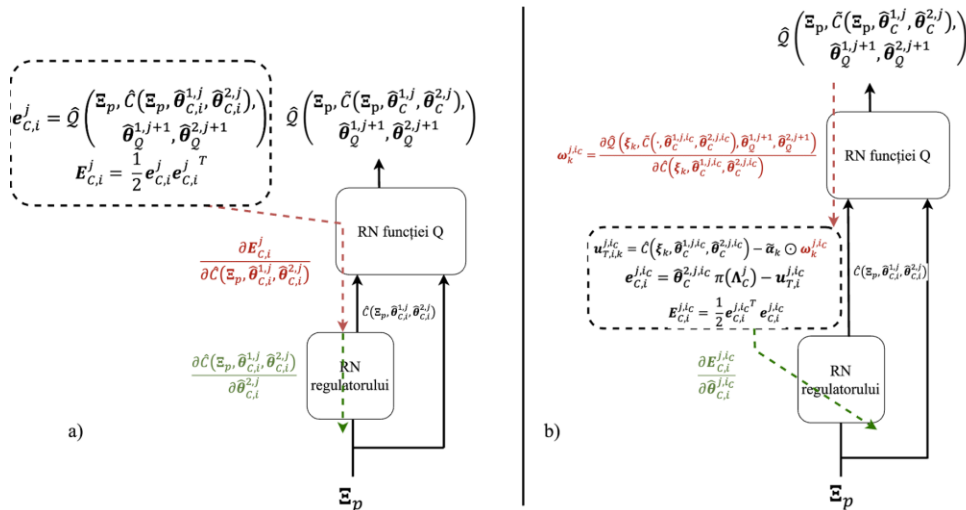


Figure 2. Architectural difference between DPGADP and CDPGADP.

This algorithm was validated on a blade pitch angle control scenario for a wind turbine and on an active suspension system for a quarter-car model. To validate the constraining mechanism, the maximum difference in the controller output was set to $\vartheta = 0.5$, and by applying a single CDPGADP learning step, the squared normalized difference between the outputs of successive controllers was computed, namely $\|\hat{C}(\xi_k^e, \theta_C^{1,7,0}, \theta_C^{2,7,0}) - \hat{C}(\xi_k^e, \theta_C^{1,7,1}, \theta_C^{2,7,1})\|_2^2$. This difference can be visualized in Fig. 3 for all states in the database generated during the exploration process, for the wind turbine system.

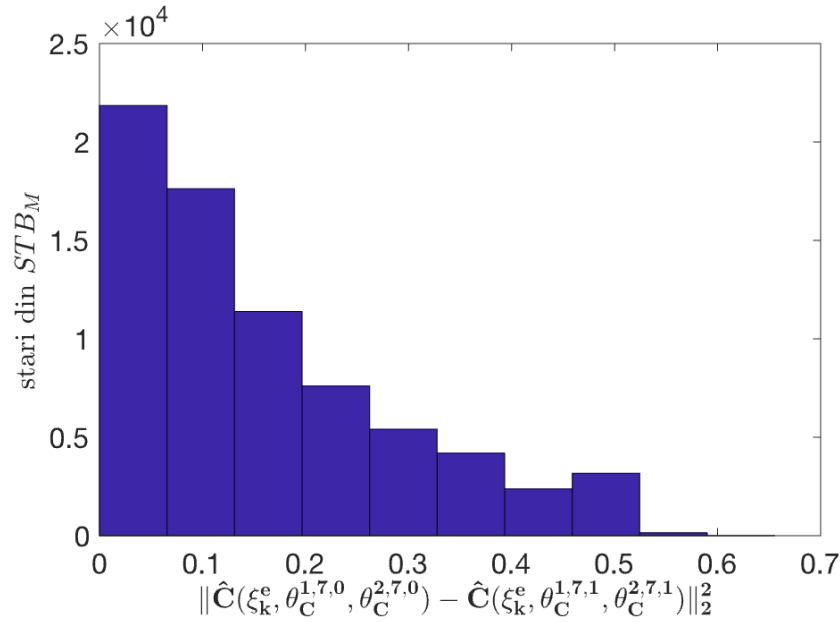


Figure 3. Histogram of normed difference between the commands generated by controller at successive iterations.

For the active suspension system for a quarter car, the CDPGADP algorithm managed to maintain the normed command differences of the controller from successive iterations at the limit $\vartheta = 1e^{-5}$. It can be observed in Fig. 4 that the controller tuned with CDPGADP managed to obtain a normed difference between two successive iterations smaller than the controller tuned with DPGADP.

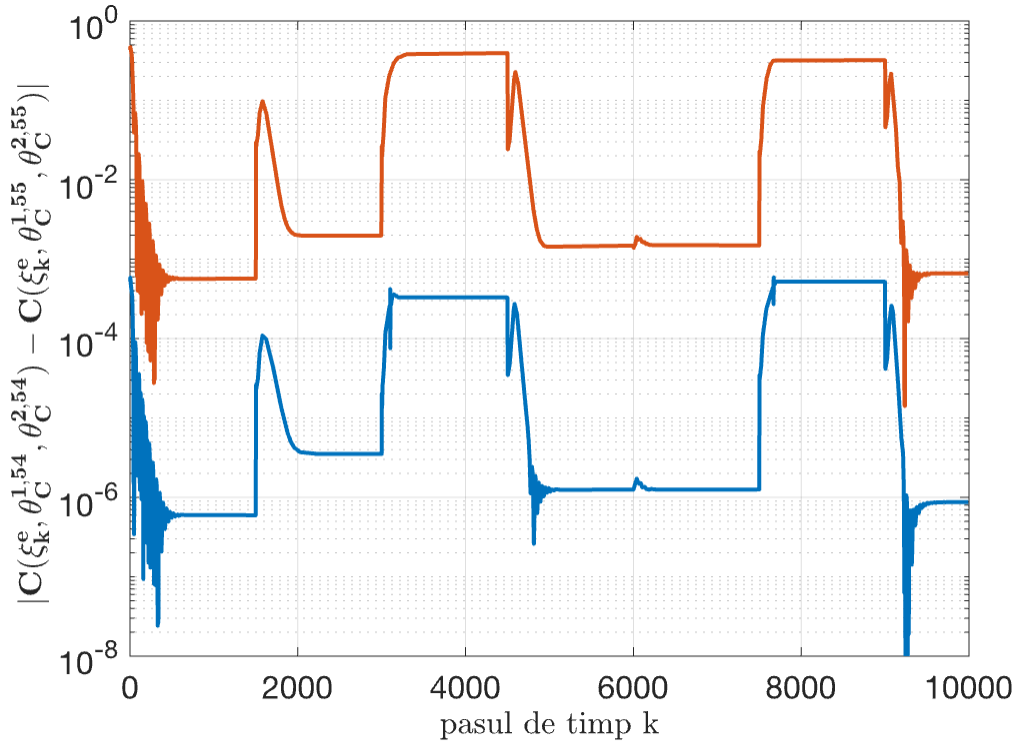


Figure 4. Normed difference between the controller at iteration $j = 54$ and $j = 55$ for algorithm DPGADP (orange) and CDPGADP (blue).

In Chapter 5, the CDPGADP algorithm was applied to update general linear controllers with time-invariant parameters, described by the equation:

$$\mathbf{u}_k = \frac{a_1}{a_0} \mathbf{u}_{k-1} - \dots - \frac{a_{n_u}}{a_0} \mathbf{u}_{k-n_e} + \frac{b_0}{a_0} \boldsymbol{\varepsilon}_k + \frac{b_1}{a_0} \boldsymbol{\varepsilon}_{k-1} + \dots + \frac{b_{n_e}}{a_0} \boldsymbol{\varepsilon}_{k-n_e}, \quad (5)$$

where n_u and n_e represent the polynomial grades associated with $\mathbf{u}(q)$ and $\boldsymbol{\varepsilon}(q)$, with $n_e \leq \chi + 1$ and $n_u \leq \chi$, and $\boldsymbol{\varepsilon}_k = \mathbf{r}_k - \mathbf{z}_k$ represents the tracking error, with $\mathbf{r}_k$ the reference signal. Validation was carried out on a third-order nonlinear system, demonstrating both the ability of this algorithm to converge to a fixed point, as it can be seen in Fig. 5, and to constrain the command outputs of successive controllers. MC2 is thus developed in Chapters 4 and 5 of the thesis and in two of the author's specialized publications, namely and [15], [16].

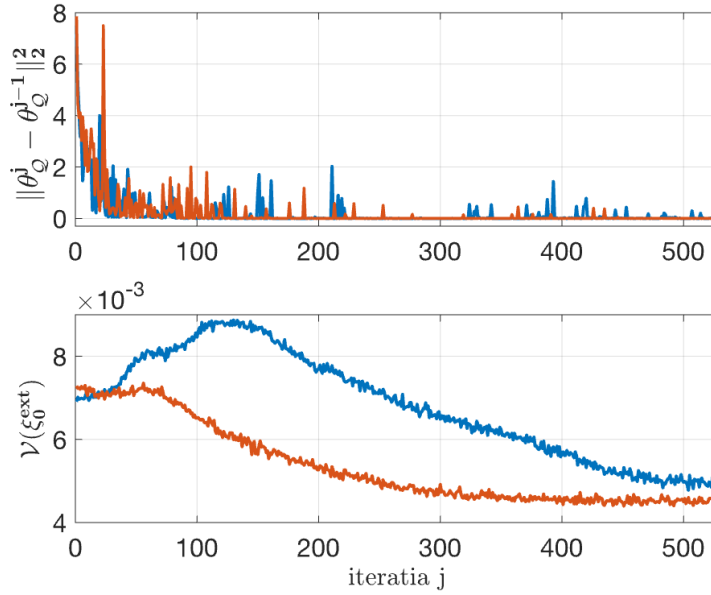


Figure 5. The convergence of CDPGADP (orange) and DPGADP (blue) algorithm.

The MFVI-SS method described in Chapter 6, in contrast to the CDPGADP method described previously, imposes on the system controlled by the model-free Value Iteration (MFVI)-tuned controller the constraint of not transitioning into a subdomain of the state space where there is a risk of physical damage (mechanical, electrical, etc.) to the controlled system, also referred to as the unsafe region. To define SO, geometric methods are used, characterizing the unsafe region as the compact set of states $Y \subset \Omega_{\Xi}$ and the safe region as $S = \Omega_{\Xi} \setminus Y$. The objective of SSMFVI is to avoid state transitions into Y . To this end, an operating region E is defined where the controller corresponding to the current VI learning iteration runs, and a region Π where the controller $C_0(\xi_k^{ext})$ runs, defining a switching controller between the two regions:

$$u_k = \begin{cases} C_j(\xi_k^{ext}), \{\xi_k^{ext} | \xi_k \in E\}, \\ C_0(\xi_k^{ext}), \{\xi_k^{ext} | \xi_k \in \Pi\}. \end{cases} \quad (6)$$

The region E represents the subdomain of the state space Ω_{Ξ} where the system operates during the exploration process. Due to the fact that the controlled system (1) in closed loop with the controller (6) is hybrid, its stability is achieved using the method associated with hybrid systems, namely the Multiple Lyapunov Functions (MLF) method, where a Lyapunov function is identified for each of the two closed-loop configurations: the system in closed loop with the MFVI-tuned controller, and the system in closed loop with the initial stabilizing controller. To this end, a Lyapunov function is defined for the closed-loop system with $C_0(\xi_k^{ext})$, namely:

$$Q_{C_0}(\xi_k^{ext}, C_0(\xi_k^{ext})) = \varsigma(\xi_k^{ext}, C_0(\xi_k^{ext})) + \gamma Q_{C_0}(\xi_{k+1}^{ext}, C_0(\xi_{k+1}^{ext})), \quad (7)$$

and a Lyapunov function associated with the closed-loop system with the controller $C_j(\xi_k^{ext})$, namely the Q-function at the current iteration:

$$Q_j(\xi_k^{ext}, u_k) = \varsigma(\xi_k^{ext}, u_k) + \gamma Q_{j-1}(\xi_{k+1}^{ext}, C_j(\xi_{k+1}^{ext})). \quad (8)$$

Theorem 6.3 demonstrates that if the value of the Lyapunov function of an operating region before switching is greater than or equal to the value of the Lyapunov function after switching back into the same region, then the switching system is stable. This can be visualized in Fig. 6, which illustrates a possible evolution of the trajectory of system (1) in closed loop with controller (6), where stability is maintained. Furthermore, in subsection 6.3.3, an early instability detection method for the system was developed, and in subsection 6.3.4, a method for attenuating the chattering phenomenon inherent to hybrid systems.

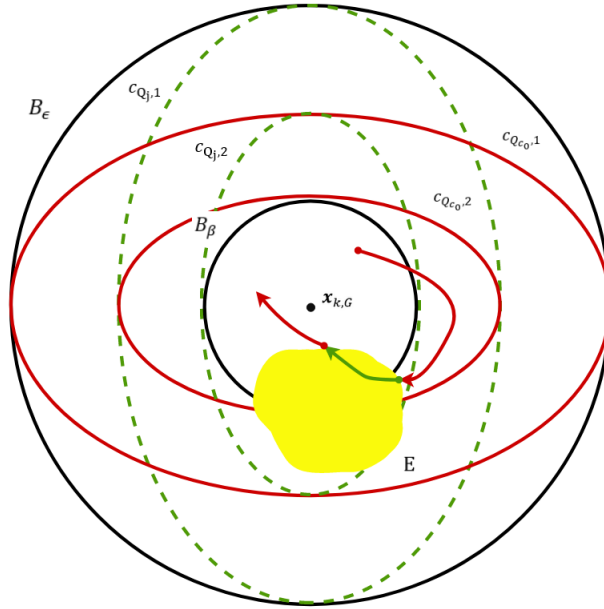


Figure 6. Description of state trajectory of the system in closed loop with control law (6).

For the practical implementation of this algorithm, neural networks were used to approximate functions (7) and (8), with the training of these networks described by Algorithms 6.2 and 6.3. For the implementation of controller (6), the region E was approximated using a method for estimating the maximum one-step state transition δ , described in Algorithm 6.4, using only data collected during exploration, as described in Algorithm 6.5. Based on this parameter, data points with a maximum transition step larger than δ were filtered out. Finally, a one-class Support Vector Machine (SVM) classifier was trained offline to discriminate the region E, with the verification of this classifier performed online during system execution. Convergence proofs were provided for all approximations, to determine the effects of approximation errors. The control architecture diagram of SSMFVI is presented in Fig. 7.

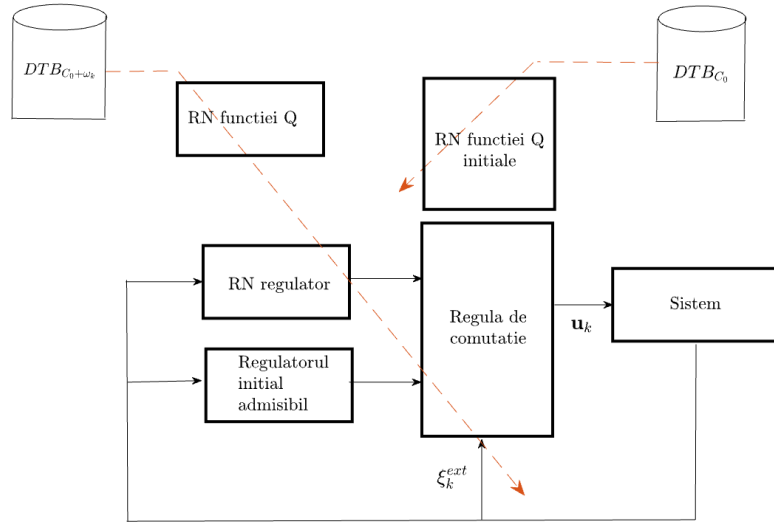


Figure 7. SSMFVI architecture.

The validation of the MFVI-SS method on the linear system

$$\xi_{k+1} = 0.6703\xi_k + 0.3297u_k, \quad (9)$$

and on the EBS, described in Fig. 8, proves the ability of this method to constrain the evolution of the system trajectory within the operating region E, regardless of the control performance of the controller at the current learning iteration, $C_j(\xi_k^{ext})$. For both types of systems, the method guarantees that transitions into the unsafe regions of the state domain are not violated, in the case of having a destabilizing controller $C_{US}(\xi_k^{ext})$ in closed loop with the controlled system.

In Fig. 9 the performance on the first-order linear system in closed loop with controller $C_{US}(\xi_k^{ext})$ and with commutation controller (6) which uses $C_{US}(\xi_k^{ext})$ instead of $C_j(\xi_k^{ext})$. $F_{MVS}(\xi_k^{ext})$ represents the MVS decision frontier signal, σ_k represents the commutation signal (1 – initial stabilizable controller $C_0(\xi_k^{ext})$, 0 – non-stabilizable controller $C_{US}(\xi_k^{ext})$), r_k the step-like reference signal, $Q(\xi_{k+1}^{ext}, C_0(\xi_{k+1}^{ext}), \theta_{Q_{C_0}}^{50})$ Q-function from iteration $j = 50$. The safe region is defined as $S = \{(\xi_k, r_k) | \xi_k \in [-\infty; 3.5], r_k \in [-\infty; 3.5]\}$, and is represented by the purple region.

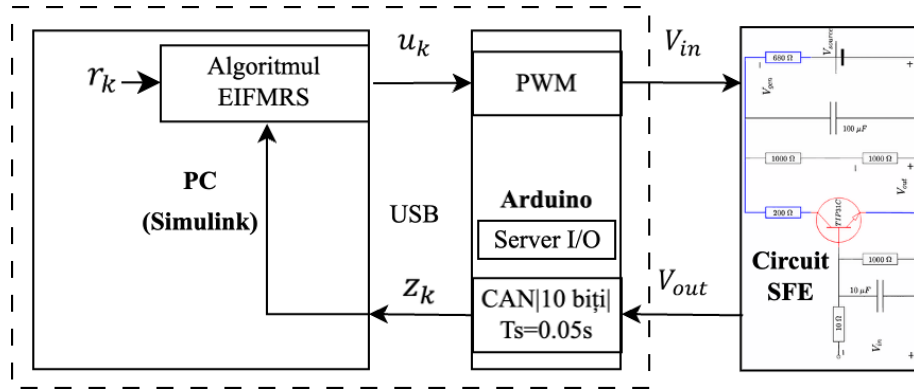


Figure 8. EBS diagram.

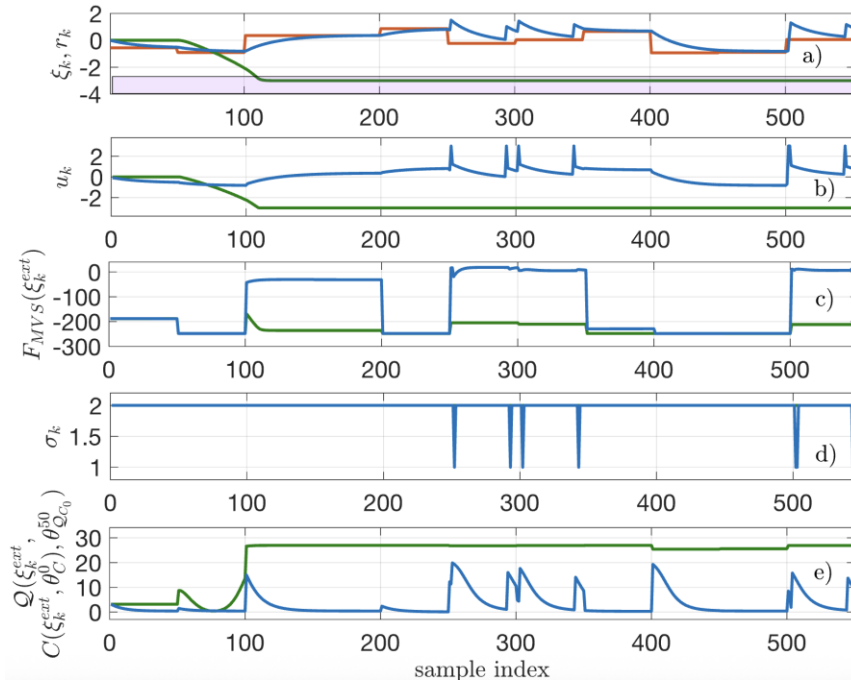


Figure 9. Tracking performance using SSMFVI with non-stabilizable controller $C_{US}(\xi_k^{ext})$: a) ξ_k, r_k (red); b) u_k ; c) $F_{MVS}(\xi_k^{ext})$; d) σ_k ; e) $Q(\xi_k^{ext}, C_0(\xi_{k+1}^{ext}), \theta_{Q_{C_0}}^{50})$. The blue signals from figure a) – e) represent the internal signals of SSMFVI with controller $C_{US}(\xi_k^{ext})$ in closed loop with system (9). The blue signals from graphs a)–e) correspond to the controller $C_{US}(\xi_k^{ext})$ using the control law (6) in closed loop with system (9).

For the EBS system, we obtain during the learning process a non-stabilizable controller at iteration $j = 180$, $C(\xi_k^{ext}, \pi_C^{180})$. The safe region is defined as $S = \{(z_k, r_k) | z_k \in [-\infty; 6.5], r_k \in [-\infty; 6.5]\}$, and is represented by the purple-colored region. Using the Relaxed Exponential Stability (RES) method $\psi(\varepsilon_0)\lambda^k + \tau$, it is proved that the algorithm SSMFVI manages to maintain the system in closed loop with the commutation controller (6) with dwell time in Fig. 10.

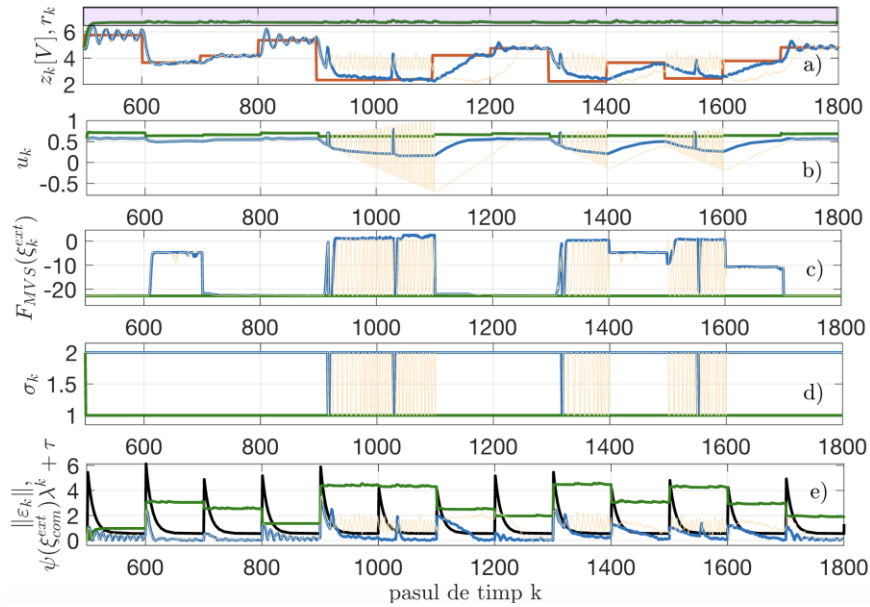


Figure 10. Tracking performance using SSMFVI with non-stabilizable controller $C(\xi_k^{ext}, \pi_C^{180})$: a) z_k, r_k (red); b) u_k ; c) $F_{MVS}(\xi_k^{ext})$; d) σ_k ; e) RES reference signal $\psi(\epsilon_0)\lambda^k + \tau$ (dark). The blue signals from figures a)-e) correspond to the reference tracking of the algorithm SSMFVI using the commutation rule (6) with controller $C(\xi_k^{ext}, \theta_C^{180})$ with dwell time in closed loop with the EBS system. The green signals from figures a)-e) correspond to the controller $C(\xi_k^{ext}, \theta_C^{180})$ in closed loop with EBS. The yellow signals from a)-e) correspond to the reference tracking using the SSMVI algorithm with commutation law (6) with $C(\xi_k^{ext}, \theta_C^{180})$ in closed loop with the EBS system.

The two applications validate the SO maintain mechanism in case of the existence of an non-stabilizable controller. This method is the subject of MC3 and is presented in the scientific paper.

Both the CDPGADP and SSMFVI methods address the problem of maintaining stability and safety only during the testing of the tuned controllers, relying on the assumption that a database exists for implementing the learning process. SO is not guaranteed during the exploration process due to the unpredictability introduced by the additive noise on the command generated by the initial stabilizing controller. Therefore, in the final part of the thesis, the problem of maintaining SO during the exploration process is addressed, resulting in the development of the Iterative Model-Free Safe Exploration (IMFSE) algorithm. MC4 is described in the paper submitted for publication.

The contributions described above have been published in WOS-indexed journals and conferences, summarized in the following list:

- [14] T. Lala, "Constrained Adaptive Dynamic Programming for PID Controllers," in *Proc. IEEE 19th Int. Conf. Control Autom. (ICCA)*, Tallinn, Estonia, Jun. 30-Jul. 3, 2025, pp. 1-6.
- [15] T. Lala, "Stability Analysis of Batch Offline Action-Dependent Heuristic Dynamic Programming Using Deep Neural Networks," *Mathematics*, vol. 13, no. 2, Jan. 2025.
- [16] T. Lala, "Safe Switching Model-Free Value Iteration for General Nonlinear Systems," *IEEE Access*, vol. 13, pp. 132170-132193, May 2025.
- [17] T. Lala, "Constrained deterministic policy gradient adaptive dynamic programming using projected gradient update," *Syst. Sci. Control Eng.*, vol. 13, no. 1, art. 2546823, 2025.

- [18] T. Lala, I. Silea, "Iterative Model Free Safe Exploration using Event-triggered Impulsive Perturbations", **accepted at IFAC congress, Busan, Korea, 23-28 August 2026**, <https://www.ifac2026.org/fairDash.do>.
-

Bibliography

- [1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140-1144, Dec. 2018.
- [2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354-359, 2017.
- [3] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529-533, 2015.
- [4] M. M. Afsar, T. Cremonesi, and P. Turrin, "Reinforcement learning based recommender systems: A survey," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1-38, Dec. 2022.
- [5] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730-27744, 2022.
- [6] J. Achiam, D. Held, A. Tamar, P. Abbeel. Constrained policy optimization," in *Proc. of Int. Conf. on Mach. Learn.*, Sydney, Australia, 2017, pp. 22–31.
- [7] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward Constrained Policy Optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019.
- [8] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. AAAI Conf. on Artif. Int.*, Honolulu, Hawaii, USA, feb. 2019, pp. 3387–3395.
- [9] M. Alshiekh, R. Bloem, R. Udiger Ehlers, B. Konighofer, S. Niekum, U. Topcu, "Safe Reinforcement Learning via Shielding," in *Proc. of the AAAI Conf on Artif. Int.*, Palo Alto, USA, 2018, pp. 1-6.
- [10] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada, "Control Barrier Functions: Theory and Applications," in *Proc. Europ. Contr. Conf.*, Piscataway, NJ, 2019, pp. 3420-3431.
- [11] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947-957, 2005.
- [12] F. Berkenkamp, M. Turchetta, A. P. Schoellig and A. Krause, "Safe Model-based Reinforcement Learning with Stability Guarantees," in *Proc. Neur. Inf. Proc. Syst.*, Long Beach, USA, Dec. 2017, pp 908-919.
- [13] Y. Chow, O. Nachum, E. Duenez-Guzman and M. Ghavamzadeh, "A Lyapunov-based Approach to Safe Reinforcement Learning," in *Proc. Neur. Inf. Proc. Syst.*, Montreal, Canada, Dec. 2018.
- [14] T. Lala, "Constrained Adaptive Dynamic Programming for PID Controllers," in *Proc. IEEE 19th Int. Conf. Control Autom. (ICCA)*, Tallinn, Estonia, Jun. 30-Jul. 3, 2025, pp. 1-6.

- [15] T. Lala, "Stability Analysis of Batch Offline Action-Dependent Heuristic Dynamic Programming Using Deep Neural Networks," *Mathematics*, vol. 13, no. 2, Jan. 2025.
- [16] T. Lala, "Safe Switching Model-Free Value Iteration for General Nonlinear Systems," *IEEE Access*, vol. 13, pp. 132170-132193, May 2025.
- [17] T. Lala, "Constrained deterministic policy gradient adaptive dynamic programming using projected gradient update," *Syst. Sci. Control Eng.*, vol. 13, no. 1, art. 2546823, 2025.
- [18] T. Lala, I. Silea, "Iterative Model Free Safe Exploration using Event-triggered Impulsive Perturbations", **accepted at IFAC congress, Busan, Korea, 23-28 August 2026**, <https://www.ifac2026.org/fairDash.do>.
- [19] J. Schulman, P. Moritz, M. Jordan, P. Abbeel, "Trust Region Policy Optimization," in *Proc. Int. Conf. on Learn. Repr.*, San Diego, USA, 2015, pp. 1-6.
- [20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," ArXiv, abs/1707.06347, 2017.
- [21] J. Achiam, D. Held, A. Tamar, P. Abbeel. Constrained policy optimization," in *Proc. of Int. Conf. on Mach. Learn.*, Sydney, Australia, 2017, pp. 22–31.
- [22] B. Luo, D. Liu, H. Wu, D. Wang, and F. L. Lewis, "Policy gradient adaptive dynamic programming for data-based optimal control," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3341-3354, 2017.
- [23] Y. Zhang, B. Zhao, and D. Liu, "Deterministic policy gradient adaptive dynamic programming for model-free optimal control," *Neurocomputing*, vol. 387, pp. 40-50, 2020.