Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων & Μηχανικων Υπολογιστων
Τομεας Σηματων, Ελεγχου και Ρομποτικης

Reachability Analysis Optimal Control

Διπλωματικη Εργασια

του

ΚΩΝΣΤΑΝΤΙΝΟΥ Δ. ΚΩΣΤΟΠΟΥΛΟΥ

Επιβλέπων: Χαράλαμπος Ψυλλάκης
Καθηγητής ΕΜΠ
Συνεπιβλέπων: Κυριάκος Βαμβουδάκης
Καθηγητής Georgia Institute of Technology

Τόκυο, Σεπτέμβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων & Μηχανικων Υπολογιστων
Τομεας Σηματων, Ελεγχου και Ρομποτικης

Reachability Analysis Optimal Control

Διπλωματικη Εργασια

του

ΚΩΝΣΤΑΝΤΙΝΟΥ Δ. ΚΩΣΤΟΠΟΥΛΟΥ

Επιβλέπων: Χαράλαμπος Ψυλλάκης
Καθηγητής ΕΜΠ

Συνεπιβλέπων: Κυριάκος Βαμβουδάκης
Καθηγητής Georgia Institute of Technology

Εγκρίθηκε από την κάτωθι τριμελή επιτροπή την 12η Σεπτεμβρίου 2023.

Χαράλαμπος Ψυλλάκης	Κυριάκος Βαμβουδάκης	Κωνσταντίνος Τζαφέστας
Καθηγητής ΕΜΠ	Καθηγητής Georgia Institute of Technology	Καθηγητής ΕΜΠ

Τόκυο, Σεπτέμβριος 2023

Κωνσταντίνος Δ. Κωστόπουλος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

Περίληψη

Αυτή η διπλωματική διερευνά το Βέλτιστο Έλεγχο (Optimal Control) μέσω Ανάλυσης Προσπελασιμότητας (Reachability Analysis). Αποσαφηνίζει κατάρχάς τα βασικά στοιχεία του βέλτιστου ελέγχου και την αξία που προσδίδει στον έλεγχο συστημάτων. Συνεχίζει με θεωρία και εφαρμογές στα Προσπελάσιμα Συνόλα (Reachable Sets), καθώς και στη χρήση του δυναμικού προγραμματισμού στην Reach-Avoid ανάλυση, μια μέθοδο επίλυσης προβλημάτων βέλτιστου ελέγχου με στόχο την επίτευξη ενός στόχου με παράκαμψη εμποδίων. Συζητείται επίσης η έννοια της Αποσύνθεσης Προσπελασιμότητας (Reachability Decomposition), η οποία απλοποιεί τα προβλήματα προσπελασιμότητας υψηλών διαστάσεων. Η μελέτη εφαρμόζει αυτές τις θεωρίες σε ένα σύστημα quadrotor 6 διαστάσεων, συγκρίνοντας τα αποτελέσματα της παραδοσιακής ανάλυσης προσπελασιμότητας με την αποσύνθεση προσπελασιμότητας, αναδεικνύοντας την αποτελεσματικότητα της τελευταίας. Στη συνέχεια, διερευνά πώς η Ενισχυτική Μάθηση (RL), μια τεχνική μηχανικής μάθησης λήψης αποφάσεων, μπορεί να ενσωματωθεί στην Ανάλυση Προσπελασιμότητας. Παρουσιάζεται η προσομοίωση ενός συστήματος 6Δ σεληνιακής προσεδάφισης, με τη χρήση RL Reachability Analysis και RL Reachability Decomposition. Τα ευρήματα αναδεικνύουν τα πλεονεκτήματα και τα μειονεκτήματα κάθε μεθόδου.

Στο κύριο μέρος αυτής της διατριβής, χρησιμοποιούμε την ανάλυση προσπελασιμότητας για να επινοήσουμε ένα zero-sum game με model-free, actor-critic Q-learning προσέγγιση για τον υπολογισμό προσπελάσιμων συνόλων σε γραμμικά ή γραμμικοποιήσιμα συστήματα, ακόμη και παρουσία διαταραχών. Αρχικά παρουσιάζουμε το δίκτυο κριτικών (critic network) που χρησιμοποιείται στην προσέγγισή μας, το οποίο αποτελείται από έναν κριτικό και δύο ηθοποιούς: έναν για τον ελεγκτή και έναν για τη διαταραχή. Στη συνέχεια, εισάγουμε μια επαυξημένη κατάσταση που ξαναγράφει τη συνάρτηση πλεονεκτήματος που εξαρτάται από τη δράση των προβλημάτων προσπελασιμότητας σε συμπαγή μορφή. Στη συνέχεια, σχεδιάζουμε έναν αλγόριθμο που προσεγγίζει τη βέλτιστη πολιτική, υπολογίζει το προσβάσιμο σύνολο και είναι βελτιστοποιημένος ως προς τη χρονική πολυπλοκότητα. Τέλος, δείχνουμε την αποτελεσματικότητα του πλαισίου μας μέσω παραδειγμάτων προσομοίωσης.

Λέξεις-κλειδιά: Βέλτιστος Έλεγχος, Δυναμικός Προγραμματισμός, Ενισχυτική Μάθηση, Reachability Decomposition, Hamilton-Jacobi Reachability, Q-Learning, Reachability Analysis, Reach-Avoid Analysis, Actor-Critic

Abstract

This Master's thesis investigates optimal control via Reachability Analysis. It first clarifies the basics of optimal control and the value it brings to system control. The focus then shifts to the theory and applications of reachable sets, and the use of dynamic programming in Reach-Avoid Analysis, a method to solve optimal control problems with the aim of reaching a target while circumventing obstacles. The concept of Reachability Decomposition, which simplifies high-dimensional reachability problems, is also discussed. The study applies these theories to a 6D quadrotor system, comparing the results of traditional reachability analysis with reachability decomposition, highlighting the effectiveness of the latter. It then explores how Reinforcement Learning (RL), a decision-making machine learning technique, can be incorporated into reachability analysis. A 6D lunar lander system simulation is presented, using RL Reachability Analysis and RL Reachability Decomposition. The findings provide insights into the pros and cons of each method in complex control environments.

In the main part of this thesis, we use reachability analysis to devise a zero-sum game with a model-free, actor-critic Q-learning approach for computing reachable sets in linear or linearizable systems, even in the presence of disturbances. We first introduce the critic network that is utilized in our approach, which consists of a critic and two actors: one for the controller and one for the disturbance. Next, we introduce an augmented state that rewrites the action-dependent advantage function of reachability problems in a compact form. We then design an algorithm that approximates the optimal policy, calculates the reachable set and is optimized in terms of time complexity. We finally show the efficacy of our framework through simulation examples.

Keywords: Optimal Control, Dynamic Programming, Reinforcement Learning, Reachability Decomposition, Hamilton-Jacobi Reachability, Q-Learning, Reachability Analysis, Reach-Avoid Analysis, Actor-Critic

Ευχαριστίες

Επί τη ευκαιρία της παρούσης διπλωματικής, θα ήθελα καταρχάς να ευχαριστήσω θερμά τον καθηγητή κ. Χ. Ψυλλάκη για την επίβλεψή της και για την ευκαιρία που μου έδωσε να συνεργαστώ με τον κ. Κ. Γ. Βαμβουδάκη της σχολής Αεροδιαστημικής Μηχανικής του Τεχνολογικού Ινστιτούτου της Τζώρτζια, ΗΠΑ, ο οποίος με δίδαξε πολλά στον τομέα του βέλτιστου ελέγχου. Αξίζει να αναφέρω ότι χάρη στην καθοδήγηση του κ. Ψυλλάκη, τόσο σχετικά με τον Έλεγχο όσο και όχι, ήρθα σε επαφή με τον κ. Βαμβουδάκη. Επίσης, ευχαριστώ ιδιαίτερα τον τελευταίο και τον κ. Νίκο-Μάριο Κοκολάκη, καθώς στάθηκαν αρωγοί μου σε όλα τα επίπεδα, καθ'όλη τη διάρκεια της έρευνας όσο και της συγγραφής. Δεν θα μπορούσα να μην δείξω την ευγνωμοσύνη μου σε όλους τους συνεργάτες-συναδέλφους με τους οποίους εκπονήσαμε λίγες ή και πολλές εργασίες εντός των πανεπιστημιακών εξαμήνων. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την υποστήριξη και την ηθική, αλλά και χρηματική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια, και ειδικά κατά την εκπόνηση της πτυχιακής αυτής στην Ατλάντα.

Κωνσταντίνος Κωστόπουλος
Σεπτέμβριος 2023

Περιεχόμενα

Κατάλογος Σχημάτων

Ακρωνύμια

BRS Backwards Reachable Set.

BRT Backwards Reachable Tube.

FRS Forward Reachable Set.

FRT Forward Reachable Tube.

HJB Hamilton-Jacobi Bellman.

HJRA Hamilton-Jacobi Reachability Analysis.

HJR Hamilton-Jacobi Reachability.

LQR Linear Quadratic Regulator.

LTI Linear Time-Invariant.

PDE Partial Differential Equation.

RL Reinforcement Learning.

ΓΧΑ Γραμμικά Χρονικώς Ανεξάρτητα.

ΕΜ Ενισχυτική Μάθηση.

ΜΔΕ Μερική Διαφορική Εξίσωση.

Κεφάλαιο 1

Βέλτιστος Έλεγχος

Ο βέλτιστος έλεγχος είναι ο τομέας των εφαρμοσμένων μαθηματικών που ασχολείται με την εύρεση του καλύτερου τρόπου ελέγχου ενός συστήματος για την επίτευξη ενός επιθυμητού αποτελέσματος. Ο στόχος του βέλτιστου ελέγχου είναι να βρεθεί μια στρατηγική ελέγχου που μεγιστοποιεί ή ελαχιστοποιεί μια συγκεκριμένη αντικειμενική συνάρτηση, ενώ παράλληλα σέβεται τους φυσικούς περιορισμούς του συστήματος [1].

Ειδικότερα,

$$J(u(t)) = \int_{t_0}^{t_f} L(x(t), u(t), t)dt + \Phi(x(t_f), t_f)$$

όπου $x(t)$ είναι το σήμα εισόδου, $u(t)$ ο έλεγχος, t_0 ο αρχικός χρόνος με $t_0 \geq 0$ και t_f ο τελικός χρόνος με $t_f \in [t_0, \infty)$, $L(x(t), u(t), t)$ είναι το στιγμιαίο κόστος, $\Phi(x(t_f), t_f)$ είναι το τελικό κόστος, ενώ το σύστημα περιγράφεται από τη χρονική εξίσωση

$$\dot{x}(t) = f(x(t), u(t), t) \tag{1.1}$$

όπου $x \in X \subseteq \mathbb{R}^n$ και $u \in U \subseteq \mathbb{R}^n$.

Οι λύσεις προβλημάτων βελτίστου ελέγχου μπορούν να ληφθούν αξιοποιώντας την αρχή ελαχίστου του Pontryagin [2], η οποία αποτελεί αναγκαία συνθήκη ή λύνοντας την HJB εξίσωση, που αποτελεί ικανή συνθήκη [3]. Η Χαμιλτονιανή H δίνεται από τον τύπο

$$H(x(t), u(t), p(t), t) = L(x(t), u(t), t) + p^T f(x(t), u(t), t)$$

όπου $p(t)$ είναι πολλαπλασιαστές Lagrange [4].

Οι αναγκαίες συνθήκες optimality [5] είναι οι εξής:

$$\dot{x} = \frac{\partial H}{\partial p} \text{ (state equation)}$$

$$\dot{p} = -\frac{\partial H}{\partial x} \text{ (costate equation)}$$

$$\frac{\partial H}{\partial u} = 0 \text{ (stationary condition)}$$

$$H(x^*(t), u^*(t), p^*(t), t) \leq H(x^*(t), u(t), p^*(t), t), \forall u \in U$$

όπου U το σύνολο των εισόδων ελέγχου. Από την τελευταία προκύπτει ότι η βέλτιστη στρατηγική ελέγχου που ελαχιστοποιεί το κόστος μετάβασης (cost-to-go/running cost) δίνεται από τον νόμο ελέγχου ανατροφοδότησης της Χαμιλτονιανής [6]:

$$u^*(x(t)) = \arg\min_{u \in U} H(x(t), u(t))$$

ο οποίος δίνει τη βέλτιστη είσοδο ελέγχου για μια δεδομένη κατάσταση $x(t)$.

Η value function [7] αντιπροσωπεύει το ελάχιστο κόστος μετάβασης από μια δεδομένη κατάσταση στην κατάσταση-στόχο και χρησιμοποιείται για τον προσδιορισμό της βέλτιστης στρατηγικής ελέγχου που ελαχιστοποιεί το κόστος. Δηλαδή,

$$V(x) = \min_{u \in U} J(u(t))$$

Η προσεγγισιμότητα (Reachability) είναι μια σημαντική έννοια στον βέλτιστο έλεγχο, καθώς παρέχει έναν τρόπο να προσδιοριστεί αν μια συγκεκριμένη κατάσταση του συστήματος μπορεί να επιτευχθεί από μια δεδομένη αρχική κατάσταση, χρησιμοποιώντας μια συγκεκριμένη στρατηγική ελέγχου. Η ανάλυση της δυνατότητας προσέγγισης είναι ιδιαίτερα χρήσιμη σε καταστάσεις όπου το σύστημα είναι πολύπλοκο και οι εισροές ελέγχου είναι περιορισμένες, όπως στην αυτόνομη πλοήγηση οχημάτων, όπου το όχημα πρέπει να αποφύγει εμπόδια ενώ οδεύει προς τον προορισμό του, σε εφαρμογές της ρομποτικής σε path planning, κ.ά., βλ. [8], [9].

Πρόκειται για μια τεχνική που χρησιμοποιείται στον βέλτιστο έλεγχο για τον προσδιορισμό της προσπελασιμότητας ενός συστήματος από μια δεδομένη αρχική κατάσταση. Παρέχει έναν τρόπο υπολογισμού του συνόλου προσπελασιμότητας του συστήματος (Reachable Set) [10], το οποίο είναι το σύνολο όλων των καταστάσεων που μπορούν να επιτευχθούν από την αρχική κατάσταση χρησιμοποιώντας μια συγκεκριμένη στρατηγική ελέγχου. Υπολογίζεται ως

$$R = \{x \mid \exists u(t) \text{ με } x(t) = x \text{ για κάποιο } t\}$$

Το Reachable Set μπορεί να χρησιμοποιηθεί για το σχεδιασμό μιας στρατηγικής ελέγχου που μεγιστοποιεί την προσπελασιμότητα του συστήματος, ενώ παράλληλα δεν παραβιάζει τους φυσικούς περιορισμούς του.

1.0.1 Hamilton-Jacobi Reachability Analysis

Ένα αξιοσημείωτο παρακλάδι του Reachability είναι το Hamilton-Jacobi Reachability Analysis (HJRA) [11]. Πρόκειται για μια τεχνική που χρησιμοποιείται στον

βέλτιστο έλεγχο, στην οποία, όταν δοθεί ένα δυναμικό σύστημα και ένας στόχος, παράγει το σύνολο των αρχικών καταστάσεων από τις οποίες το σύστημα εγγυημένα θα επιτύχει τον στόχο αυτό. Επιπλέον, η μέθοδος παρέχει τον βέλτιστο έλεγχο για την επίτευξη του στόχου. Μπορεί επίσης να εφαρμοστεί για εμπόδια και μη ασφαλείς καταστάσεις: Η προσπελασιμότητα HJ θα επιστρέψει ένα σύνολο αρχικών καταστάσεων από τις οποίες το σύστημα πρέπει να μείνει μακριά για να παραμείνει ασφαλές, καθώς και τον έλεγχο για να το επιτύχει αυτό. Παρέχεται επίσης η δυνατότητα να συνδυαστούν σενάρια τόσο με στόχους όσο και με εμπόδια, και δύναται ακόμη και να λάβουμε υπόψη τις χειρότερες δυνατές διαταραχές (π.χ. άνεμος). Εκτός αυτού, η Hamilton-Jacobi ανάλυση προσεγγισιμότητας αποτελεί ισχυρό εργαλείο για την επίλυση προβλημάτων βέλτιστου ελέγχου, καθώς μπορεί να χειριστεί μη γραμμικά συστήματα με περιορισμούς και αβεβαιότητες.

Βασίζεται στη μερική διαφορική εξίσωση Hamilton-Jacobi, η οποία περιγράφει την εξέλιξη της συνάρτησης αξίας του προβλήματος βέλτιστου ελέγχου. Η μερική διαφορική εξίσωση Hamilton-Jacobi δίνεται από:

$$\frac{\partial V}{\partial t} + H(x, \frac{\partial V}{\partial x}) = 0$$

όπου $V(x,t)$ είναι η συνάρτηση αξίας.

Πρόκειται στην πράξη για ένα πρόβλημα δυναμικού προγραμματισμού, κατά το οποίο ένα πρόβλημα βέλτιστου ελέγχου αναλύεται σε μια ακολουθία απλούστερων υποπροβλημάτων μέσω της επίλυσης της μερικής διαφορικής εξίσωσης Hamilton-Jacobi προς τα πίσω στο χρόνο, ξεκινώντας από την κατάσταση-στόχο και πηγαίνοντας προς τα πίσω στην αρχική κατάσταση. Η λύση του HJRA παρέχει τη συνάρτηση αξίας, η οποία στη συνέχεια χρησιμοποιείται για τον υπολογισμό της βέλτιστης στρατηγικής ελέγχου.

Reachable Sets

Δεδομένου ότι μιλάμε για Reachability, η εξίσωση 1.1 εξειδικεύεται ως εξής:

$$\dot{x}(t) = f(x(t), u(t), t), x(0) = x_0, t \geq 0 \tag{1.2}$$

Αν οριστεί ως $\xi_x^u(\cdot)$ η τροχιά που ξεκινάει από την κατάσταση $x(\cdot)$ ελέγχω $u(\cdot)$, όπου $\xi : \mathbb{R}_+ \to X$, τότε για ένα *target set* $\mathcal{T} \subset \mathbb{R}^n$ - το σύνολο των καταστάσεων που αναπαριστούν το στόχο των βελτίστων τροχιών - και ένα *constraint set* $\mathcal{C} \subset \mathbb{R}^n$ - το σύνολο όλων των επιτρεπτών states κατά την εξέλιξη μιας τροχιάς - ορίζονται το *ασφαλές σύνολο* και το *αποτυχές σύνολο*.

Το ασφαλές σύνολο, εφεξής safe set, συναρτήσει του $\mathcal{C}$ ορίζεται ως το σύνολο των αρχικών καταστάσεων από τις οποίες ο ελεγκτής μπορεί να διατηρήσει επ' αόριστον το σύστημα εντός του συνόλου περιορισμών:

$$\Omega(\mathcal{C}) := \{x \in X | \exists u \in \mathbb{U}, \forall \tau \geq 0, \xi_x^u(\tau) \in \mathcal{C}\} \tag{1.3}$$

Αντίστοιχα, συμβολίζοντας το συμπλήρωμα του συνόλου περιορισμών ως $\mathcal{F} = \mathcal{C}^c$, λαμβάνουμε το *failure set*, στην οποία περίπτωση το Ω γίνεται:

$$\Omega(\mathcal{F}) := \{x \in X | \exists u \in \mathbb{U}, \forall \tau \geq 0, \xi_x^u(\tau) \notin \mathcal{F}\} \tag{1.4}$$

Έτσι, ορίζεται το *Backwards Reachable Set (BRS)* του στόχου $\mathcal{T}$ ως:

$$\mathcal{R}(\mathcal{T}) := \{x \in X | \exists u \in \mathbb{U}, \exists \tau \geq 0, \xi_x^u(\tau) \in \mathcal{T}\} \tag{1.5}$$

Στην ουσία, το σύνολο αρχικών καταστάσεων από τις οποίες το σύστημα πρέπει να ξεκινήσει ώστε με βεβαιότητα και με βέλτιστο τρόπο να φτάσει στο στόχο του σε χρόνο t_f, αποτελεί το Backwards Reachable Set. Στο πλαίσιο αυτό, ορίζεται και το Backwards Reachable Tube (BRT), το οποίο αποτελεί το σύνολο αρχικών καταστάσεων από τις οποίες το σύστημα πρέπει να ξεκινήσει ώστε με βεβαιότητα και με βέλτιστο τρόπο να φτάσει στο στόχο του σε χρόνο *το πολύ* t_f. Παρόμοιοι ορισμοί υπάρχουν και για Forward Reachable Set (FRS) και Forward Reachable Tube (FRT) [12].

Από την 1.4 είναι εμφανές ότι το $\Omega(\mathcal{F})$ είναι το *avoid backwards reachable set*. Από εδώ και πέρα θα αναφέρεται ως $\mathcal{A}(\mathcal{F})$ για λόγους συμβολισμού. Συνδυάζοντας, λοιπόν, τα backwards reachable set και avoid backwards reachable set ορίζεται το *Reach-Avoid Set*, δηλαδή το σύνολο των καταστάσεων για τις οποίες ο έλεγχος οδηγεί το σύστημα στο $\mathcal{T}$ ενώ ταυτοχρόνως αποφεύγει το $\mathcal{F}$. Συμβολίζοντας το σύνολο αυτό ως $\mathcal{RA}$ προκύπτει:

$$\mathcal{RA}(\mathcal{T}; \mathcal{F}) := \{x \in X | \exists u \in \mathbb{U}, \exists \tau \geq 0, \xi_x^u(\tau) \in \mathcal{T} \wedge \forall \kappa \in [0, \tau] \xi_x^u(\kappa) \notin \mathcal{F}\} \tag{1.6}$$

το οποίο δηλώνει το σύνολο των καταστάσεων για τις οποίες υπάρχει κάποιο σήμα ελέγχου που μπορεί να οδηγήσει το σύστημα στο $\mathcal{T}$ αποφεύγοντας το $\mathcal{F}$ όλες τις προγενέστερες χρονικές στιγμές.

Δυναμικός Προγραμματισμός για Reach-Avoid Ανάλυση

Ο υπολογισμός του συνόλου Reach-Avoid απαιτεί τον ορισμό δύο επιφανειακών συναρτήσεων που θα διαχωρίζουν τις καταστάσεις που ανήκουν στον χώρο $\mathcal{T}$ από εκείνες που βρίσκονται στον χώρο $\mathcal{F}$. Οι συναρτήσεις αυτές θα είναι *signed* συναρτήσεις $l(\cdot) : \mathcal{X} \to \mathbb{R}$ και $g(\cdot) : \mathcal{X} \to \mathbb{R}$, οι οποίες είναι συνεχείς κατά Lipschitz εκ κατασκευής και ορίζονται ως εξής:

$$l(x(t)) = \begin{cases} \leq 0, & x(t) \in \mathcal{T} \\ > 0, & x(t) \in \mathcal{T}^c \end{cases}$$

και

$$g(x(t)) = \begin{cases} \leq 0, & x(t) \in \mathcal{C} \\ > 0, & x(t) \in \mathcal{F} \end{cases}$$

αντίστοιχα.

Το l θα αναφέρεται ως η συνάρτηση *payoff* και το g ως η συνάρτηση *discriminator*, όπως περιγράφεται στο [13].

Οι επιφανειακές συναρτήσεις αυτές καθιστούν δυνατό τον ορισμό του continuous-time functional

$$\mathcal{V}^u(x(t)) = \min_{\tau\in[t,T]} \max\left\{ l(\xi^u_{x(t)}(\tau),\tau),\ \max_{\kappa\in[t,T]} g(\xi^u_{x(t)}(\kappa),\kappa)\right\}. \tag{1.7}$$

Πρέπει να σημειωθεί εδώ ότι το $\mathcal{V}^u(x(t))$ θα είναι αρνητικό εάν και μόνο εάν η τροχιά φτάσει στο $\mathcal{T}$, χωρίς ποτέ να παραβιάζει το σύνολο περιορισμών $\mathcal{C}$.

Η ελαχιστοποίηση του functional (1.7) ορίζει τη *συνάρτηση αξίας* ως:

$$\mathcal{V}(x(t)) = \inf_{u\in U} \mathcal{V}^u(x(t)) \tag{1.8}$$

Ισοδύναμα,

$$V(x(t)) = \begin{cases} \leq 0, & x(t)\in \mathcal{RA}(\mathcal{T};\mathcal{F}) \\ > 0, & x(t)\notin \mathcal{RA}(\mathcal{T};\mathcal{F}) \end{cases}$$

Τέλος, για να συγκλίνει η value function στη λύση, πρέπει να ικανοποιεί την ακόλουθη *RABE* (Reach-Avoid Bellman Equation):

$$\mathcal{V}(x(t)) = \max\left\{ g(x(t)), \min\{l(x(t)),\ \inf_{u\in U}\mathcal{V}(x^u_+)\}\right\}.$$

όπου $x^u_+ = \xi^u_x(t+\delta-t)$, με $0<\delta\leq T-t$ και, προφανώς, $0\leq t<T$.

1.0.2 Reachability Decomposition

Η προσπελασιμότητα HJ μπορεί να μην είναι πρακτική για συστήματα υψηλών διαστάσεων λόγω της υπολογιστικής πολυπλοκότητας της μεθόδου που αυξάνεται εκθετικά με τη διάσταση του συστήματος. Ως αποτέλεσμα, προτάθηκε το System Decomposition [14]. Με αυτό το τέχνασμα, το BRS μπορεί να υπολογιστεί με ακρίβεια για αποσυνδεδεμένα δυναμικά (decoupled dynamics) και δυναμικά που είναι συνδεδεμένα με έναν συγκεκριμένο τρόπο που οδηγεί στα λεγόμενα *αυτοτελή υποσυστήματα* (self-contained subsystems). Για dynamics που δεν υπάγονται στις παραπάνω κατηγορίες, προσεγγιστικές λύσεις υπολογίζονται.

Αν ένα σύστημα

$$x = (y_1, y_2, y_c),$$

μπορεί να τεθεί στην επονομαζόμενη self-contained form

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_2} \\ \dot{y_c} \end{bmatrix} = \begin{bmatrix} f_1(y_1, y_c, u) \\ f_2(y_2, y_c, u) \\ f_c(y_c, u) \end{bmatrix}$$

τότε είναι διασπάσιμο ως

$$\begin{bmatrix} \dot{y_1} \\ \dot{y_c} \end{bmatrix} = \begin{bmatrix} f_1(y_1, y_c, u) \\ f_c(y_c, u) \end{bmatrix} and \begin{bmatrix} \dot{y_2} \\ \dot{y_c} \end{bmatrix} = \begin{bmatrix} f_2(y_2, y_c, u) \\ f_c(y_c, u) \end{bmatrix}$$

κατά το [15].

Αν δεν υπάρχει coupling στον έλεγχο, η value V του αρχικού συστήματος υπολογίζεται ως:

$$V(y_1, y_2, y_c) = \max\{V(y_1, y_c), V(y_2, y_c)\}$$

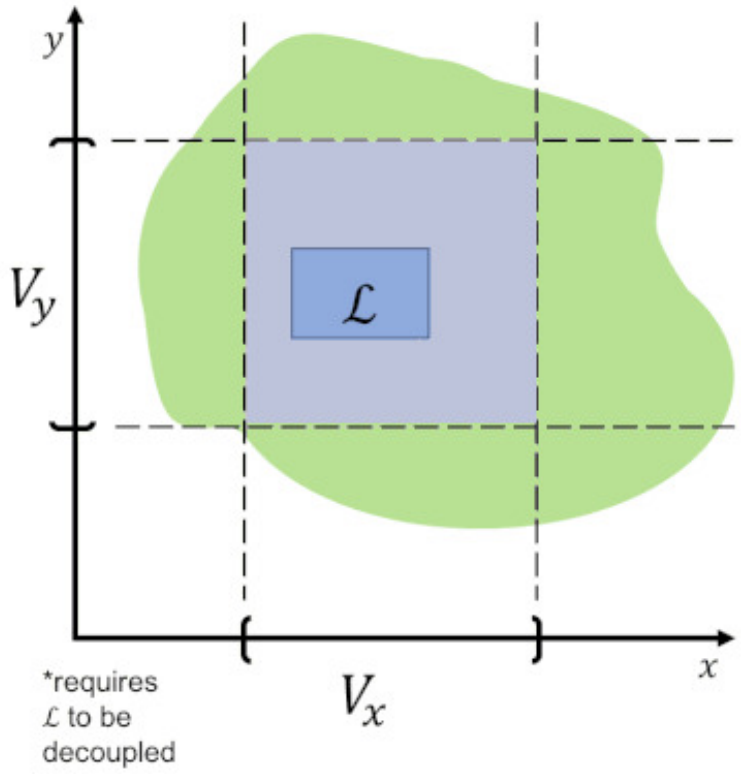
και το BRS υπολογίζεται με ακρίβεια. Στην ουσία, η αξία V σε ένα οποιοδήποτε σημείο είναι η μέγιστη, δηλαδή η χειρότερη, value μεταξύ των δυο δισδιάστατων συστημάτων.

Εάν ο έλεγχος είναι coupled, η αποσύνθεση είναι ακριβής μόνο για τις περιπτώσεις στις οποίες πρέπει να χρησιμοποιηθεί ο βέλτιστος έλεγχος *είτε* για το υποσύστημα 1 *είτε* για το υποσύστημα 2. Η δήλωση αυτή εικονοποιείται παρακάτω ως εξής:

Reconstruction \ Coupled subsystem controls	Avoid target	Reach target
Intersection of back projection	✓	✗
Union of back projection	✗	✓

Σχήμα 1.1: Πίνακας υποσυστημάτων με coupled control. Η πλήρης κι επιτυχής ανακατασκευή του backwards reachable set ισχύει για τις περιπτώσεις με το τικ [16].

Σε περιπτώσεις όπου το σύστημα δεν μπορεί να τεθεί σε αυτοτελή μορφή ή όπου η ανακατασκευή δεν μπορεί να γίνει, εξετάζουμε την προσεγγιστική αποσύνθεση (*approximate decomposition*). Η έλλειψη ζωτικών καταστάσεων σε κάθε υποσύστημα μπορεί να αντισταθμιστεί θεωρώντας ότι πρόκειται για άγνωστες ¨ψεύτικες διαταραχές¨, οι οποίες περιέχουν πάντα τις χειρότερες δυνατές τιμές. Συνεπώς, τα συστήματα είναι πλέον decoupled και μπορεί να εφαρμοστεί reachability decomposition για να υπολογιστούν τα backwards reachable sets, αλλά με σημαντικά χαμηλότερη ακρίβεια. Πρώτα προβάλλεται το target set στα δύο υποσυστήματα και στη συνέχεια γίνεται progagate κάθε υποσύστημα προς τα πίσω για να ληφθεί το BRS κάθε υποσυστήματος. Στη συνέχεια λαμβάνεται η τομή (χειρότερη περίπτωση) μεταξύ αυτών των συνόλων για να προσεγγιστεί το BRS του συνολικού συστήματος.



Σχήμα 1.2: Υποπροσέγγιση. Το υπολογισμένο reachable set (μπλε) είναι σημαντικά μικρότερο από το πραγματικό του αρχικού συστήματος (πράσινο) [17].

Στην περίπτωση αυτή, συνήθως υπολογίζεται το 40-55% του Backwards Reachable Set του αρχικού συστήματος.

1.0.3 Hamilton-Jacobi Reachability Analysis σε Quadrotor 6 Διαστάσεων

Οι προσομοιώσεις μέχρι το τέλος του κεφαλαίου αυτού βρίσκονται στο GitHub[1] μου, fork του framework ονόματι helperOC του HJ Reachability Group. Αναδιοργάνωσα το repository τους ώστε να απαρτίζεται από περισσότερα παραδείγματα και έφτιαξα και ένα bug που βρήκα στο dynSys του Quadrotor6D. Οι αλλαγές υπάρχουν σε ξεχωριστό branch στο δικό μου repo, ενώ έχω ήδη ενημερώσει έναν συντηρητή του δικού τους για την αλλαγή και σκοπεύω να ανοίξω ένα pull request ώστε να συγχωνευθεί. Χρησιμοποιούνται Level Set Methods [18], δηλαδή δυναμικός προγραμματισμός, στο εν λόγω framework.

Η δυναμική του συστήματος του εξαδιάστατου quadrotor είναι:

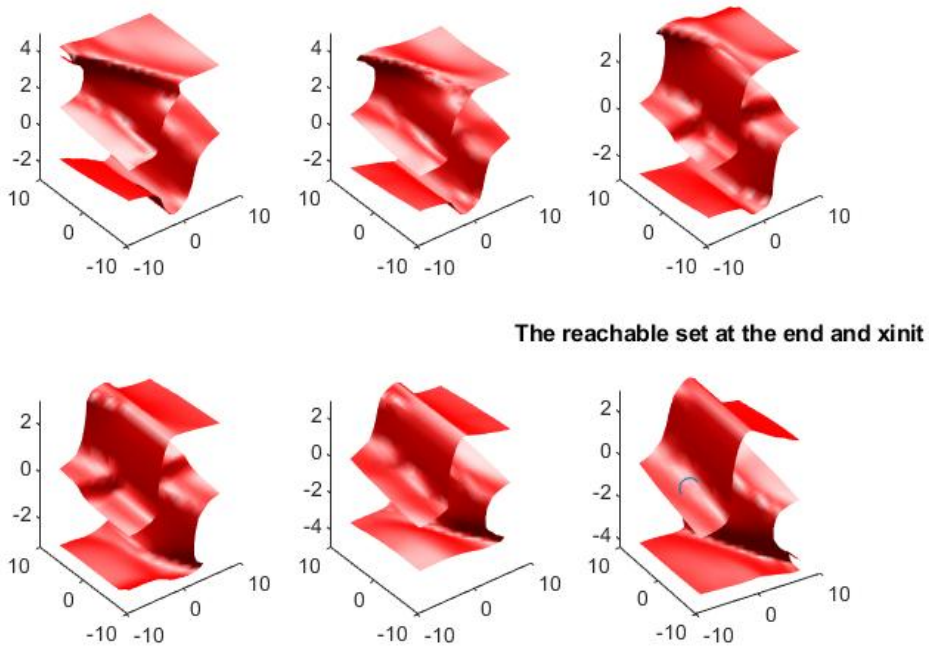
$$z = (x, \dot{x}, y, \dot{y}, \theta, \dot{\theta}) \tag{1.9}$$

[1] https://github.com/Costopoulos/helperOC/tree/master

$$\dot{z} = \begin{bmatrix} \dot{x} \\ -\frac{T\dot{x}+sin\theta(u_1+u_2)}{m} \\ \dot{y} \\ \frac{cos\theta(u_1+u_2)-mg-T\dot{y}}{m} \\ \dot{\theta} \\ \frac{l(u_2-u_1)-R\dot{\theta}}{I_{yy}} \end{bmatrix}$$

όπου l είναι το μήκος από το κέντρο μάζας έως το άκρο του τηλεκατευθυνόμενου οχήματος, T είναι η μεταφορική αντίσταση translational drag, R είναι η περιστροφική αντίσταση rotational drag, m είναι η μάζα του τετράτροχου και I_{yy} είναι η ροπή αδράνειάς του.

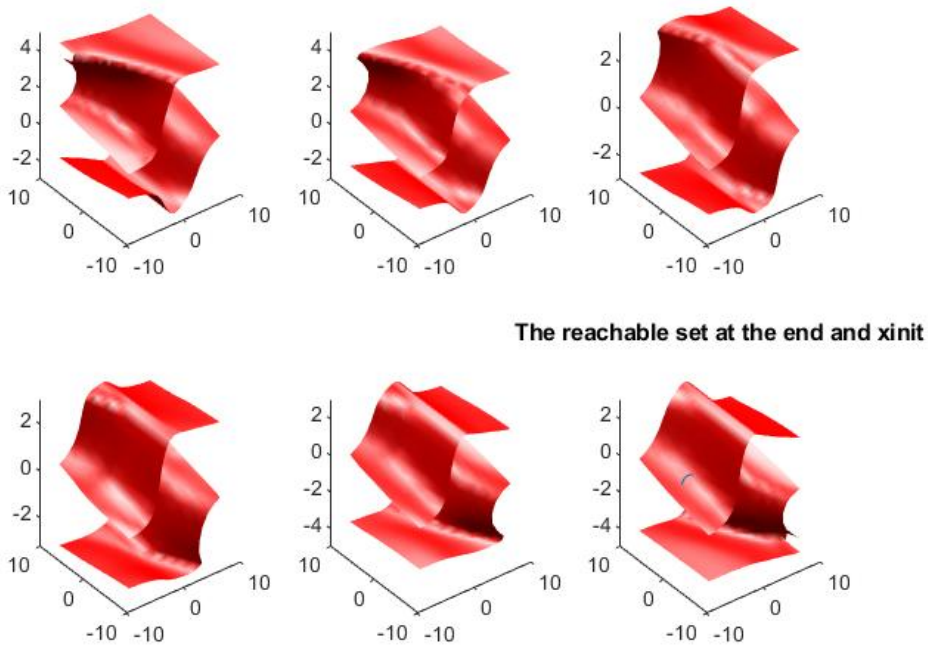
Στην εικόνα παρακάτω φαίνεται ο υπολογισμός του BRS για εργασία αποφυγής κυλινδρικού εμποδίου ακτίνας 3 που βρίσκεται στην αρχή των αξόνων. Σε επεξεργαστή Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz με μνήμη RAM 16GB ο υπολογισμός διήρκησε 12 λεπτά. xinit είναι η αρχική θέση του drone.



Σχήμα 1.3: Υπολογισμός backwards reachable set. Το υπολογισμένο reachable set βρίσκεται στην κάτω-δεξιά εικόνα.

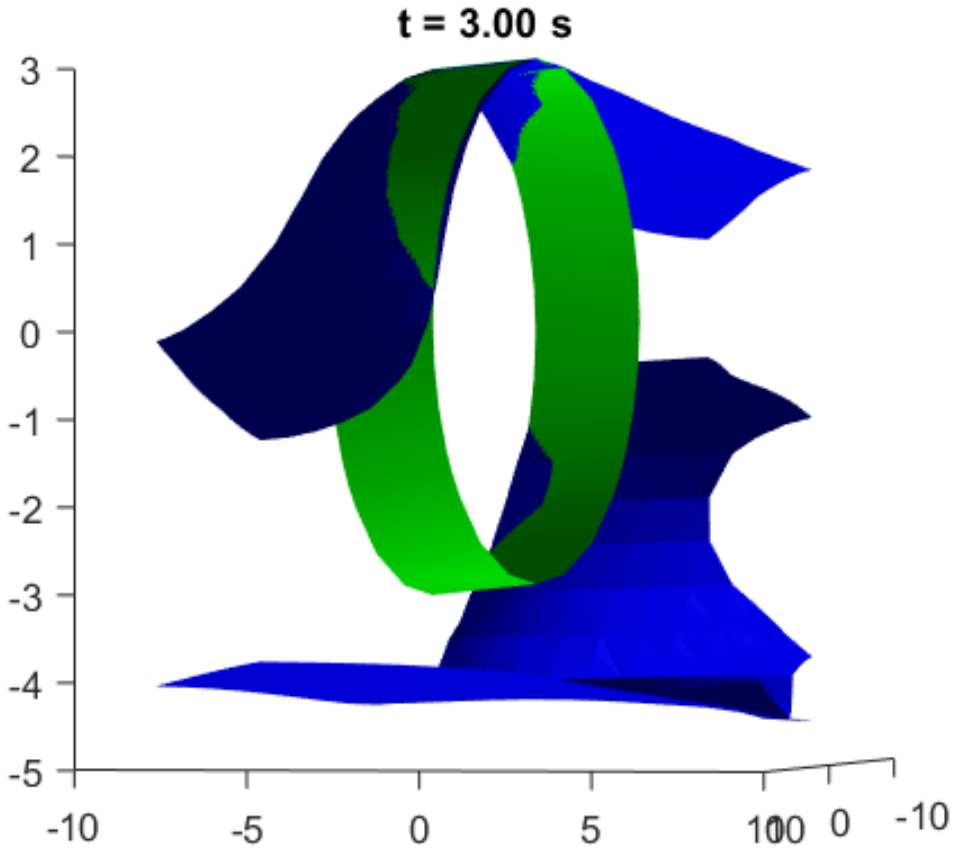
Προσθέτοντας Γκαουσιανό θόρυβο standard deviation στον x-άξονα με τιμή 0.2,

ελήφθησαν τα παρακάτω.



Σχήμα 1.4: Υπολογισμός backwards reachable set με λευκό θόρυβο στον οριζόντιο άξονα. Το υπολογισμένο reachable set βρίσκεται στην κάτω-δεξιά εικόνα.

Είναι εμφανές ότι ο θόρυβος αναστάτωσε πλήρως τον υπολογισμό του BRS, με αποτέλεσμα το quadrotor να κινδυνεύει να έρθει σε επαφή με το εμπόδιο.



Σχήμα 1.5: Υπολογισμός backwards reachable set με λευκό θόρυβο στον οριζόντιο άξονα. Το υπολογισμένο reachable set βρίσκεται στην κάτω-δεξιά εικόνα.

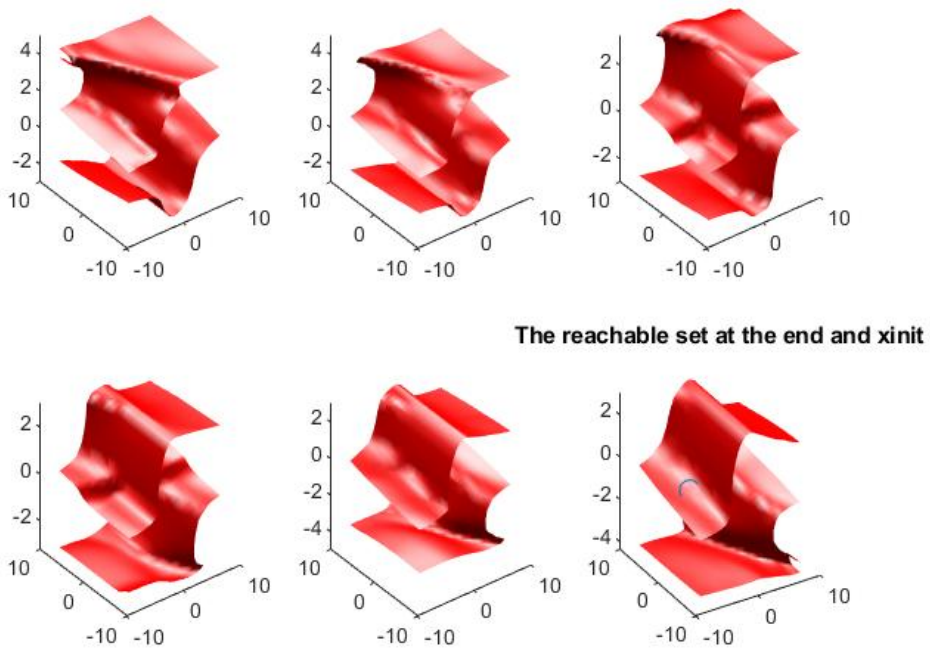
Decomposed Quadrotor

Το σύστημα 1.9 διασπάται στα δυο παρακάτω συστήματα:

$$p_1 = (x, \dot{x}, \theta, \dot{\theta})$$

$$p_2 = (y, \dot{y}, \theta, \dot{\theta})$$

Δεδομένου ότι μελετάται σενάριο αποφυγής, το Reachability Decomposition είναι εφικτό και το BRS μπορεί να κατασκευαστεί με πλήρη ακρίβεια. Στον ίδιο επεξεργαστή, η ανακατασκευή του παρακάτω Backwards Reachable Set ολοκληρώθηκε σε 4 λεπτά, στο 1/3 του αρχικού χρόνου.



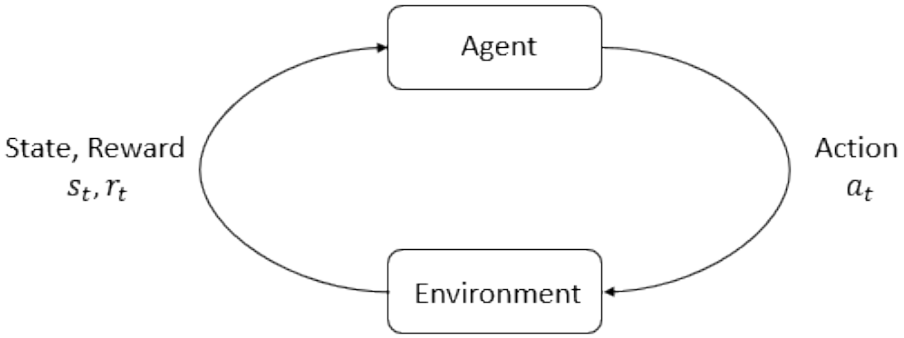
Σχήμα 1.6: Υπολογισμός backwards reachable set για το διασπασμένο σύστημα. Το υπολογισμένο reachable set βρίσκεται στην κάτω-δεξιά εικόνα και είναι ίδιο με του αρχικού συστήματος.

Το υπολογισμένο Reachable Set είναι τελείως ίδιο με αυτό του συνολικού συστήματος. Το ποσοστό κερδισμένου χρόνου με τη μέθοδο αυτή είναι 66.7%.

Κεφάλαιο 2

Ενισχυτική Μάθηση

Η ενισχυτική μάθηση (Reinforcement Learning) είναι μια προσέγγιση μηχανικής μάθησης όπου ένας πράκτορας (agent) μαθαίνει πώς να επιτύχει έναν στόχο μέσω αλληλεπιδράσεων δοκιμής και λάθους με ένα δυναμικό περιβάλλον. Τυποποιεί την ιδέα ότι η ανταμοιβή (reward) ή η τιμωρία (punishment) ενός agent για τη συμπεριφορά του τον κάνει πιο πιθανό να επαναλάβει ή να αποφύγει αυτή τη συμπεριφορά στο μέλλον.



Σχήμα 2.1: Βρόχος αλληλεπίδρασης πράκτορα-περιβάλλοντος [19].

Ο πράκτορας και το περιβάλλον είναι οι βασικοί χαρακτήρες του RL. Το περιβάλλον είναι το περιβάλλον στο οποίο ο πράκτορας λειτουργεί και αλληλεπιδρά όπως φαίνεται στο Σχήμα 2.1. Ο πράκτορας αποφασίζει ποια ενέργεια θα λάβει αφού ολοκληρώσει μια (ενδεχομένως ελλιπή) παρατήρηση της κατάστασης του κόσμου σε κάθε στάδιο της αλληλεπίδρασης. Το περιβάλλον μπορεί να αλλάξει από μόνο του ή ως αποτέλεσμα των ενεργειών του πράκτορα. Ο πράκτορας λαμβάνει επίσης ένα σήμα ανταμοιβής (reward) από το περιβάλλον του, το οποίο είναι ένας αριθμητικός δείκτης του πόσο καλή ή άσχημη είναι η κατάσταση του κόσμου αυτή τη στιγμή. Ο στόχος του πράκτορα είναι να μεγιστοποιήσει τη συνολική ανταμοιβή του, την απόδοση (return). Οι μέθοδοι ενισχυτικής μάθησης επιτρέπουν στον πράκτορα να επιλέγει συμπεριφορές που θα τον βοηθήσουν να επιτύχει τον στόχο του.

Συνεχίζοντας με τις ορολογίες, μια κατάσταση (state) s είναι μια πλήρης περιγραφή της κατάστασης του κόσμου. Δεν υπάρχει καμία πληροφορία για τον κόσμο που

να είναι κρυμμένη από την κατάσταση. Μια παρατήρηση (observation) ο είναι μια μερική περιγραφή μιας κατάστασης, η οποία μπορεί να παραλείπει πληροφορίες. Για παράδειγμα, η κατάσταση ενός drone μπορεί να περιγράφεται από τις γωνίες και τις ταχύτητες των αρθρώσεων του, ενώ μια παρατήρηση μπορεί να περιγράφεται από έναν πίνακα με τις (x,y) τιμές αυτού και των εμποδίων γύρω του.

Διαφορετικά περιβάλλοντα επιτρέπουν διαφορετικούς τύπους ενεργειών. Ο όρος ¨χώρος δράσης¨ (action space) αναφέρεται στο σύνολο όλων των πιθανών δραστηριοτήτων σε ένα συγκεκριμένο περιβάλλον. Ορισμένα περιβάλλοντα, όπως αυτό στο οποίο γίνεται προσομοίωση παρακάτω στο κεφάλαιο αυτό, έχουν διακριτούς χώρους δράσης, όπου μόνο ένας πεπερασμένος αριθμός κινήσεων είναι διαθέσιμος στον πράκτορα. Άλλα περιβάλλοντα προσφέρουν συνεχείς χώρους δράσης, όπως εκείνα όπου ο πράκτορας χειρίζεται ένα ρομπότ στον πραγματικό κόσμο. Οι δράσεις είναι διανύσματα πραγματικών τιμών σε χώρους που είναι συνεχείς.

Οι επιπτώσεις αυτής της διάκρισης για τις προσεγγίσεις Βαθιάς Ενισχυτικής Μάθησης (Deep RL) είναι μεγάλες. Ορισμένες οικογένειες αλγορίθμων μπορούν να χρησιμοποιηθούν άμεσα μόνο σε μία κατάσταση και θα απαιτούσαν σημαντική επεξεργασία στην άλλη.

Προχωρώντας στην πολιτική (policy), πρόκειται για έναν κανόνα που χρησιμοποιείται από έναν πράκτορα για να αποφασίσει σε ποιες ενέργειες θα προβεί. Μπορεί να είναι ντετερμινιστική, όποτε συμβολίζεται με μ: $a_t = \mu(s_t)$, ή στοχαστική, στην οποία περίπτωση συμβολίζεται με π : $a_t \sim \pi(\cdot|s_t)$. Στη βαθιά ενισχυτική μάθηση εφαρμόζονται παραμετροποιημένες πολιτικές, δηλαδή κανόνες των οποίων τα αποτελέσματα είναι υπολογίσιμες συναρτήσεις που εξαρτώνται από ένα σύνολο παραμέτρων (για παράδειγμα, τα βάρη και τις προκαταλήψεις (biases) ενός νευρωνικού δικτύου), οι οποίες μπορούν να αλλαχθούν μέσω τεχνικών βελτιστοποίησης προκειμένου να επηρεαστεί η συμπεριφορά.

Οι δράσεις ενός agent πραγματοποιούνται ανάλογα με την πολιτική του. Έτσι, ορίζεται η έννοια της μετάβασης καταστάσεως (state transition): πρόκειται για το τι συμβαίνει στον κόσμο μεταξύ της κατάστασης τη στιγμή (t, s_t) και της κατάστασης τη στιγμή ($t+1$, s_{t+1}) και εξαρτάται μόνο από τους φυσικούς νόμους του περιβάλλοντος και την πιο πρόσφατη δράση a_t. Μπορεί να είναι ντετερμινιστική ή στοχαστική.

Με βάση όλες τις παραπάνω έννοιες, οι έννοιες της ανταμοιβής και της απόδοσης καθίστανται πιο κατανοητές. Η συνάρτηση ανταμοιβής εξαρτάται από την τρέχουσα κατάσταση του κόσμου, την ενέργεια που μόλις έγινε και την επόμενη κατάσταση του κόσμου:

$$r_t = R(s_t, a_t, s_{t+1}).$$

Η απόδοση ορίζεται ως το άθροισμα όλων των ανταμοιβών που έχει λάβει ποτέ ο πράκτορας, αλλά discounted ανάλογα με το πόσο μακριά στο μέλλον θα ληφθούν. Αυτή η διατύπωση της ανταμοιβής περιλαμβάνει έναν συντελεστή έκπτωσης $\gamma \in$

$(0, 1)$:

$$R(\xi) = \sum_{t=0}^{\infty} \gamma^t r_t.$$

Ο συντελεστής έκπτωσης χρειάζεται διότι ένα infinite-horizon άθροισμα ανταμοιβών μπορεί να μη συγκλίνει σε πεπερασμένη τιμή, γεγονός που πρέπει να αποφευχθεί [20].

Επομένως, ο στόχος του RL είναι να διαλέξει την πολιτική που μεγιστοποιεί την αναμενόμενη απόδοση (expected return) όταν ο πράκτορας πράττει με βάση αυτήν. Πιο συγκεκριμένα, η αναμενόμενη απόδοση, που συμβολίζεται με $J(\pi)$, είναι τότε:

$$J(\pi) = E_{\xi \sim \pi}[R(\xi)],$$

με Ε να είναι η μέση τιμή.

Το κεντρικό πρόβλημα βελτιστοποίησης στο RL μπορεί τότε να εκφραστεί ως εξής

$$\pi^* = \arg\max_{\pi} J(\pi),$$

όπου π^* είναι η βέλτιστη πολιτική.

Τέλος, η συνάρτηση αξίας (value function) είναι σπουδαίας σημασίας, καθώς δείχνει την αξία μιας κατάστασης. Με τον όρο αξία, εννοείται η αναμενόμενη απόδοση εάν ξεκινήσει ο πράκτορας στην εκάστοτε κατάσταση ή στο εκάστοτε ζεύγος κατάστασης-δράσης και στη συνέχεια δρα σύμφωνα με μια συγκεκριμένη πολιτική για πάντα. Έτσι, η on-policy value function ή απλώς value function, $V^{\pi}(s)$, η οποία δίνει την αναμενόμενη απόδοση εάν ο πράκτορας ξεκινήσει από κατάσταση s και ενεργεί πάντα σύμφωνα με την πολιτική π ορίζεται ως:

$$V^{\pi}(s) = E_{\xi \sim \pi}[R(\xi) \,|s_0 = s]\,.$$

Η βέλτιστη συνάρτηση αξίας, optimal value function, η οποία δίνει την αναμενόμενη απόδοση εάν ο πράκτορας ξεκινήσει από κατάσταση s και ενεργεί πάντα σύμφωνα με τη βέλτιστη πολιτική π^* ορίζεται ως:

$$V^*(s) = \max_{\pi} E_{\xi \sim \pi}[R(\xi) \,|s_0 = s]\,,$$

ή, ισοδύναμα,

$$V^*(s) = \max_{\pi} J(\xi)$$

Να σημειωθεί εδώ ότι υπάρχουν δύο μεγάλες κατηγορίες αλγορίθμων ενισχυτικής μάθησης: model-free και model-based. Οι μέθοδοι χωρίς μοντέλο δεν προσπαθούν να μάθουν ένα ρητό μοντέλο του περιβάλλοντος. Αντί αυτού, επικεντρώνονται στην άμεση εκμάθηση μιας πολιτικής ή μιας συνάρτησης αξίας για τη λήψη αποφάσεων και τη βελτίωση της απόδοσης. Οι αλγόριθμοι χωρίς μοντέλο μαθαίνουν από τις αλληλεπιδράσεις με το περιβάλλον, συνήθως με δοκιμή και σφάλμα, χωρίς

να δημιουργούν μια αναπαράσταση της υποκείμενης δυναμικής του συστήματος. Παραδείγματα αλγορίθμων ενισχυτικής μάθησης χωρίς μοντέλα περιλαμβάνουν: Q-Learning [21], [22], SARSA (State-Action-Reward-State-Action), DQN (Deep Q-Network) [23], Policy Gradient methods [24], όπως PPO [25], REINFORCE [26], [27].

Οι model-based μέθοδοι, από την άλλη, περιλαμβάνουν την εκμάθηση ενός ρητού μοντέλου του περιβάλλοντος, το οποίο αποτυπώνει τη δυναμική του συστήματος. Το μοντέλο αυτό χρησιμοποιείται για το σχεδιασμό και τη λήψη αποφάσεων. Μόλις το μοντέλο μαθευτεί, μπορούν να εφαρμοστούν διάφοροι αλγόριθμοι σχεδιασμού για την εύρεση μιας βέλτιστης πολιτικής ή συνάρτησης αξίας. Παραδείγματα αλγορίθμων ενισχυτικής μάθησης που βασίζονται σε μοντέλα περιλαμβάνουν: Μεθόδους δυναμικού προγραμματισμού (π.χ. Value Iteration, Policy Iteration), Δενδρική αναζήτηση Μόντε Κάρλο (MCTS), Model-Based παραλλαγές του Q-Learning (π.χ. Dyna-Q [28]).

Οι μέθοδοι που βασίζονται σε μοντέλα μπορεί να είναι επωφελείς όταν το περιβάλλον είναι σχετικά απλό και η δυναμική του είναι γνωστή ή μπορεί να προσεγγιστεί με ακρίβεια. Με την ύπαρξη ενός μοντέλου, οι μέθοδοι αυτές μπορούν να εκτελούν προσομοιώσεις και να σχεδιάζουν εκ των προτέρων για να βρουν τις καλύτερες ενέργειες που πρέπει να ληφθούν. Ωστόσο, μπορεί να είναι περιορισμένη η εφαρμογή τους όταν το περιβάλλον είναι ιδιαίτερα πολύπλοκο και η απόκτηση ενός ακριβούς μοντέλου είναι δύσκολη.

2.0.1 HJRA με Ενισχυτική Μάθηση

Ένα από τα βασικά αντικείμενα μελέτης στο RL είναι κι ο βέλτιστος έλεγχος, ενώ μελετάται εκτενώς και το Hamilton-Jacobi Reachability. Αντί η συνάρτηση αξίας να υπολογίζεται με δυναμικό προγραμματισμό και level set methods, μαθαίνεται από τον agent κατά την αλληλεπίδρασή του με το δυναμικό περιβάλλον.

Για συνεχείς χώρους, η βέλτιστη συνάρτηση αξίας ικανοποιεί τη μερική διαφορική εξίσωση Hamilton-Jacobi (HJ PDE). Η λύση της HJ PDE δίνει το βέλτιστο κόστος μετάβασης για κάθε κατάσταση, συνθέτοντας κατά αυτόν τον τρόπο τη βέλτιστη πολιτική. Ωστόσο, η επίλυση των HJ PDEs αποτελεί πρόκληση για προβλήματα υψηλών διαστάσεων, πρόβλημα γνωστό και ως the curse of dimensionality. Μια εναλλακτική λύση είναι η προσέγγιση της συνάρτησης αξίας με τη χρήση νευρωνικών δικτύων σε ένα πλαίσιο γνωστό ως νευρωνικός δυναμικός προγραμματισμός (neural dynamic programming). Οι παράμετροι του νευρωνικού δικτύου βελτιστοποιούνται με τη χρήση αλγορίθμων RL, όπως ο Proximal Policy Optimization (PPO) και Q-Learning. Βέβαια, επειδή οι policy-gradient μέθοδοι επιστρέφουν κατανομή πιθανοτήτων επί των δράσεων, αντί για τη μέγιστη αναμενόμενη μελλοντική ανταμοιβή (όπως το Deep Q-Learning), προτιμούνται σε προβλήματα με high-dimensional action spaces.

Κατά πλήρη αναλογία με τα λεγόμενα στο παραπάνω υποκεφάλαιο, σε ένα πρόβλημα βέλτιστου ελέγχου ο σκόπος είναι να βρεθεί η καλύτερη πολιτική ελέγχου

ώστε να επιτευχθεί ένας συγκεκριμένος στόχος. Συνήθως, στόχος είναι να ελαχιστοποιηθεί το running cost J, από το οποίο προκύπτει η Χαμιλτονιανή H, η οποία αντιπροσωπεύει τη συνολική ενέργεια του συστήματος. Επομένως, ο καλύτερος ελεγκτής u^* προκύπτει σπαταλώντας τη λιγότερη δυνατή ενέργεια, εξ ου και

$$u^\star = \arg\min_u H(u)$$

Ομοίως, η βέλτιστη συνάρτηση αξίας, που λαμβάνεται όταν ο πράκτορας δρα σύμφωνα με τη βέλτιστη πολιτική u^*, δίνεται από τη σχέση:

$$V^\star(x) := \min_u J(x)$$

Στο πλαίσιο Reachability Analysis που έχει οριστεί ως εδώ, το Lagrange functional που θέλουμε να μεγιστοποιηθεί είναι, κατά αναλογία με το $V^\pi(s)$ που ορίστηκε παραπάνω:

$$\mathcal{V}(x) = \sum_\tau \gamma^\tau r(x_\tau, u_\tau), \tag{2.1}$$

όπου $r(x,u) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ είναι η *συνάρτηση ανταμοιβής (reward function)* και $\gamma \in [0,1)$ είναι η παράμετρος χρονικής έκπτωσης, γνωστή ως *ρυθμός μάθησης (learning rate)*. Για το πρόβλημα του υποκεφαλαίου 2.0.2 επιλέχθηκε παραλλαγή του αλγορίθμου Q-Learning που συγκλίνει επιτυχώς στη λύση.

2.0.2 Εφαρμογή σε Lunar-Lander

Στο υποκεφάλαιο αυτό εφαρμόζονται τα παραπάνω στη σεληνάκατο (lunar lander) του OpenAI Gym [29]. Η κατάσταση δίνεται από το εξαδιάστατο διάνυσμα $z = (x, \dot{x}, y, \dot{y}, \theta, \dot{\theta})$ και ο προσεληνωτής έχει 4 πιθανές δράσεις (δηλαδή τετραδιάστατο action space): να χρησιμοποιήσει την αριστερή, την κύρια ή την αριστερή μηχανή, ή να μην προβεί σε δράση. Το περιβάλλον αυτό δεν έχει τις μεταβλητές bool που αντιπροσωπεύουν αν κάθε πόδι αγγίζει το φεγγάρι, επομένως δεν είναι 8D, όπως το αντίστοιχο του OpenAI Gym. Να σημειωθεί επίσης ότι είναι διακριτό περιβάλλον. Ορίζοντας την ελάχιστη προσημασμένη απόσταση D μεταξύ ενός σημείου $[x,y]^T \in \mathbb{R}^2$ κι ενός συνόλου S ως

$$D(x,y;S) = c\left(\min_{[\hat{x},\hat{y}]\in\partial\mathcal{S}} \sqrt{(x-\hat{x})^2 + (y-\hat{y})^2}\right) \tag{2.2}$$

όπου $c = -1$ αν $[x,y]^T \in S$ και $c = +1$ αλλού, με ∂S να είναι το σύνορο του συνόλου.

Το περιθώριο ασφαλείας ορίζεται από την απόσταση D από το κέντρο του σκάφους έως την επιφάνεια της σελήνης, καθώς κι από την ελάχιστη απόσταση από τα αριστερά, τα δεξιά και τα άνω όρια. Μαζί, αυτές οι τέσσερις συνιστώσες ορίζουν ένα πολύγωνο $P_c \subset \mathbb{R}^2$ το οποίο οριοθετεί το constraint set. Το ασφαλές περιθώριο (safety margin) ορίζεται επομένως ως $g(s) = -D(x,y;P_c)$. Το target set T ορίζε-

ται ως μια ορθογώνια περιοχή εντός του συνόλου περιορισμών (δηλ. $T \subset P_c$) που σκοπός είναι να φταστεί χωρίς ύπαρξη συγκρούσεων. Το περιθώριο στόχου (target margin) ορίζεται ομοίως ως η ελάχιστη προσημασμένη απόσταση D από το T , δηλαδή $l(s) = D(x, y; T)$.

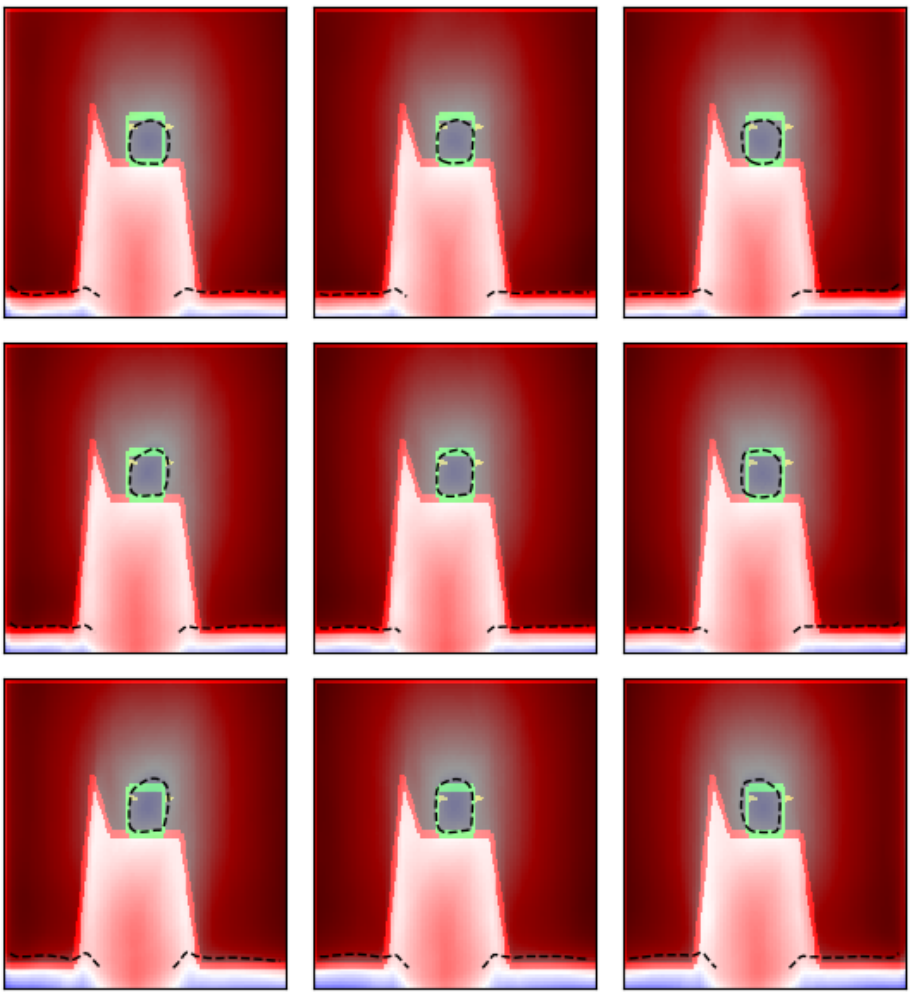
Χρησιμοποιώντας τον αλγόριθμο Reach-Avoid Q-Learning [30] υπολογίζονται τα reach-avoid $\mathcal{RA}$ σύνολα.

Οι προσομοιώσεις μέχρι το τέλος του κεφαλαίου αυτού βρίσκονται στο GitHub repository[1] μου, fork του framework ονόματι safety_rl του Safe Robotics Lab. Έφτιαξα ένα bug που βρήκα στο repository τους σχετικά με τα visualizations του lunar lander, κατά το οποίο η εικόνα σβήνει πιο γρήγορα από ό,τι γράφεται πληροφορία πάνω της, συνεπώς αποτυπώνεται κενή.

Στα παρακάτω πειράματα το περιβάλλον τρέχει για 3.5 εκατομμύρια gradient updates με αρχικό γ 0.9 και τελικό-μέγιστο 0.999999. Το learning rate αυξάνεται ανά 500000 gradient updates με decay 0.1, δηλαδή το γ παίρνει τις τιμές $\{0.9, 0.99, 0.999, 0.9999, 0.99999, 0.999999\}$. Το πράσινο ορθογώνιο παραλληλόγραμμο δηλώνει το στόχο $\mathcal{T}$, το μπλε αναδεικνύει τον επιτρεπόμενο χώρο P_c και το κόκκινο το failure set που σκοπός είναι να αποφευχθεί. Οι 9 εικόνες αποτυπώνουν slices της συνάρτησης αξίας για διαφορετικές τιμές του v_x και v_y, με $\theta = 0$ και $\dot{\theta} = 0$. Επάνω, μεσαία και κάτω σειρά αντιστοιχούν σε $v_y = 1, 0, -1$ αντίστοιχα. Πρώτη, δεύτερη και τελευταία στήλη αντιστοιχούν σε $v_x = -1, 0, 1$ αντίστοιχα. Η διακεκομμένη γραμμή υποδηλώνει το σύνολο μηδενικών επιπέδων και τα βέλη υποδηλώνουν την κατεύθυνση της αρχικής ταχύτητας.

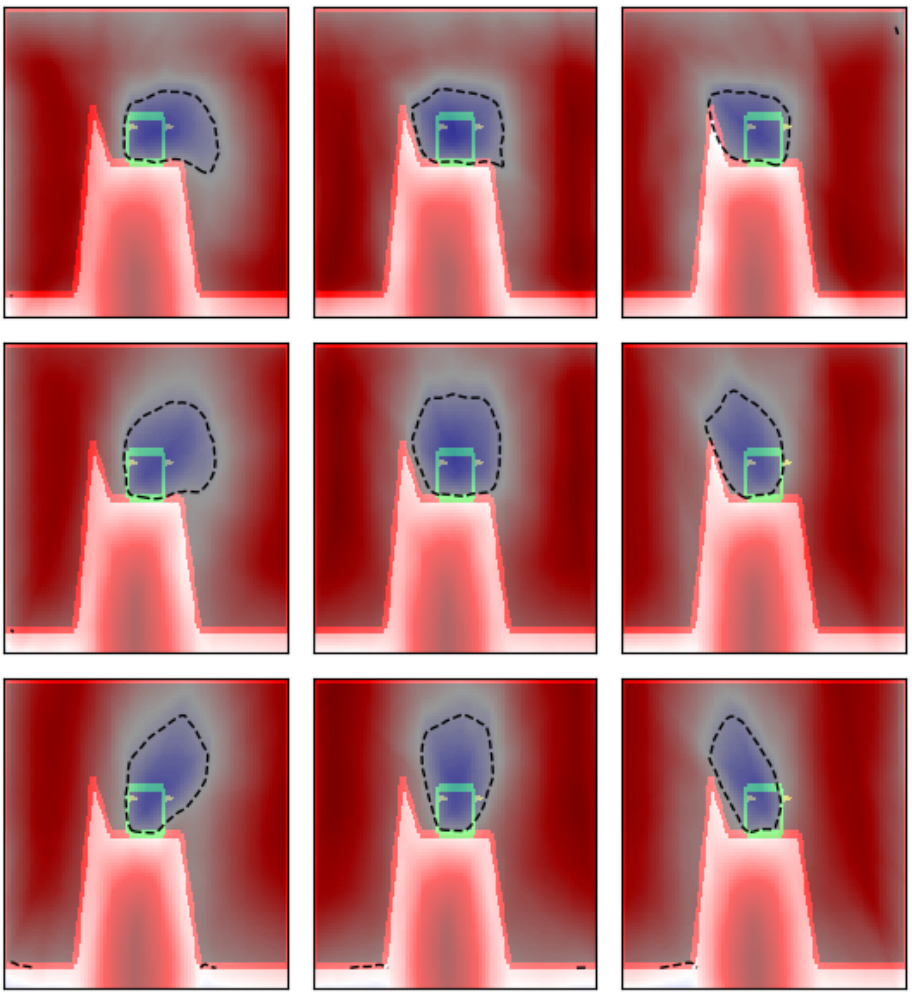
Ο κώδικας έτρεξε για 6 ώρες και 10 λεπτά σε NVIDIA GeForce GTX 1050 και 11 ώρες και 15 λεπτά σε dedicated 8GB RAM, 4-core CPU, hosted στο Linode.

[1] https://github.com/Costopoulos/Safety-RL



Σχήμα 2.2: Υπολογισμός reach-avoid set για το σύστημα 6Δ σεληνακάτου. Ο ρυθμός μάθησης είναι 0.9.

Ο ρυθμός μάθησης 0.9 δεν είναι αρκετός για να υπολογίσει ο αλγόριθμος σωστά το reach-avoid set.

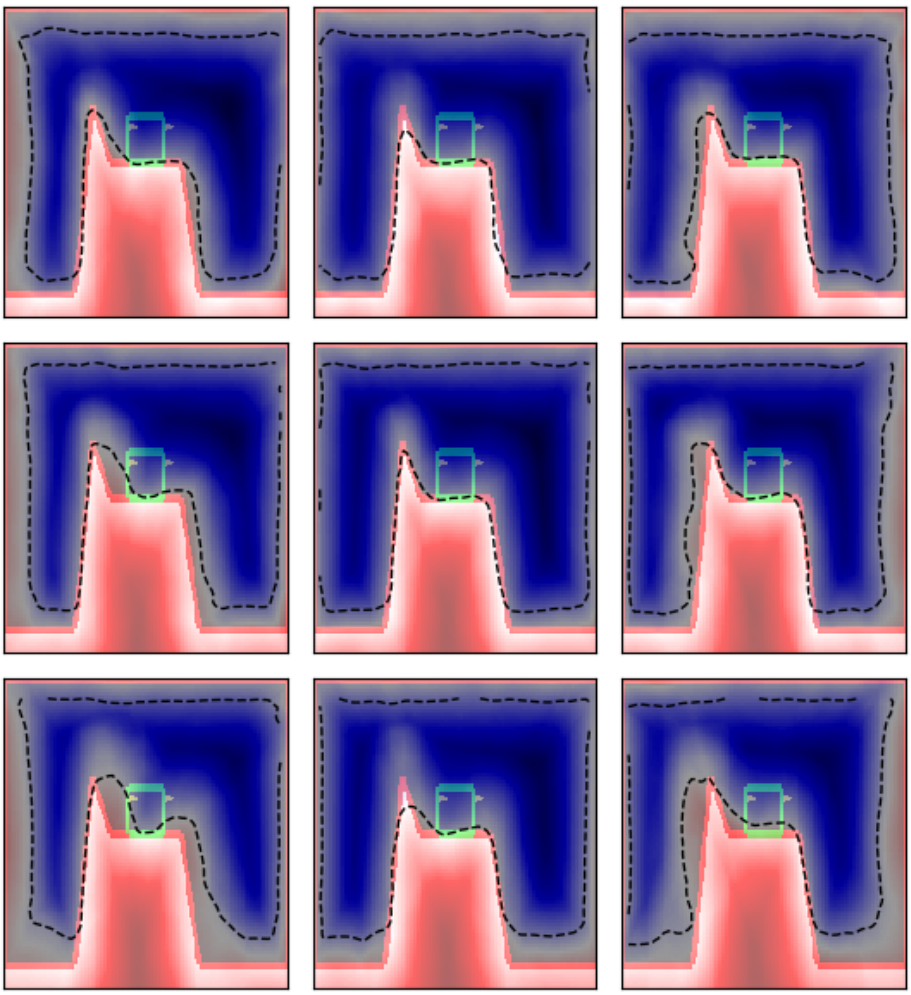


Σχήμα 2.3: Υπολογισμός reach-avoid set για το σύστημα 6Δ σεληνακάτου. Ο ρυθμός μάθησης είναι 0.99.

Ο ρυθμός μάθησης 0.99 δεν είναι αρκετός για να υπολογίσει ο αλγόριθμος σωστά το reach-avoid set, ωστόσο το P_c ξεκινάει να σχηματίζεται.

Για ρυθμό μάθησης από 0.999 και πάνω αλλάζει σημαντικά η απεικονιζόμενη value function. Το reach-avoid set έχει σχηματιστεί πλέον σχεδόν τελείως. Οι αρχικές ταχύτητες προς τα αριστερά και προς τα δεξιά προκαλούν τη συνάρτηση αξίας να επεκτείνεται οριζόντια από το όριο και το εμπόδιο, όπως αναμενόταν. Ομοίως, οι αρχικές κατακόρυφες ταχύτητες προκαλούν την συνάρτηση τιμής να είναι θετική κοντά στο κάτω και το πάνω μέρος του περιβάλλοντος, αναδεικνύοντας έτσι το εμπόδιο.

Για το λόγο αυτό συμπεριλαμβάνεται μόνο η εικόνα με ρυθμό μάθησης 0.999999.



Σχήμα 2.4: Υπολογισμός reach-avoid set για το σύστημα 6Δ σεληνακάτου. Ο ρυθμός μάθησης είναι 0.999999.

Ο ρυθμός μάθησης 0.999999 είναι ο βέλτιστος. Οι διακεκομένες γραμμές που οριοθετούν τον απαγορευμένο από τον επιτρεπόμενο χώρο προσεγγίζουν ικανοποιητικά τα όρια της επιφάνειας της σελήνης, συμπεριλαμβανομένων και των γωνιών, και του χώρου στα αριστερά και δεξιά. Το reach-avoid set έχει σχηματιστεί πλέον.

Decomposed Lunar-Lander

Το σύστημα της εξαδιάστατου σεληνακάτου διασπάται στα δυο παρακάτω συστήματα:

$$z_1 = (x, \dot{x}, \theta, \dot{\theta})$$

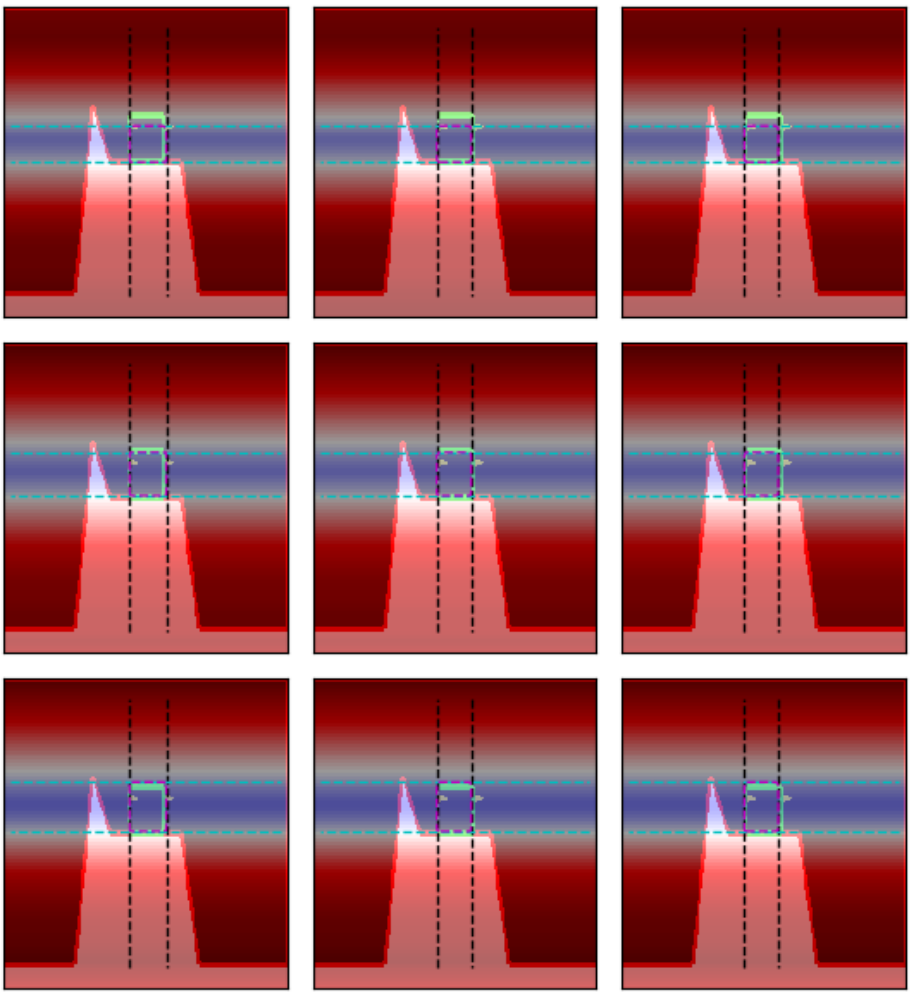
$$z_2 = (y, \dot{y}, \theta, \dot{\theta})$$

Δεδομένου ότι μελετάται σενάριο άφιξης, το Reachability Decomposition δεν είναι εφικτό και η ανακατασκευή του reach-avoid set θα υποεκτιμηθεί κατά τη μέθοδο του approximate decomposition.

Στα παρακάτω πειράματα το περιβάλλον τρέχει για 1.5 εκατομμύρια gradient

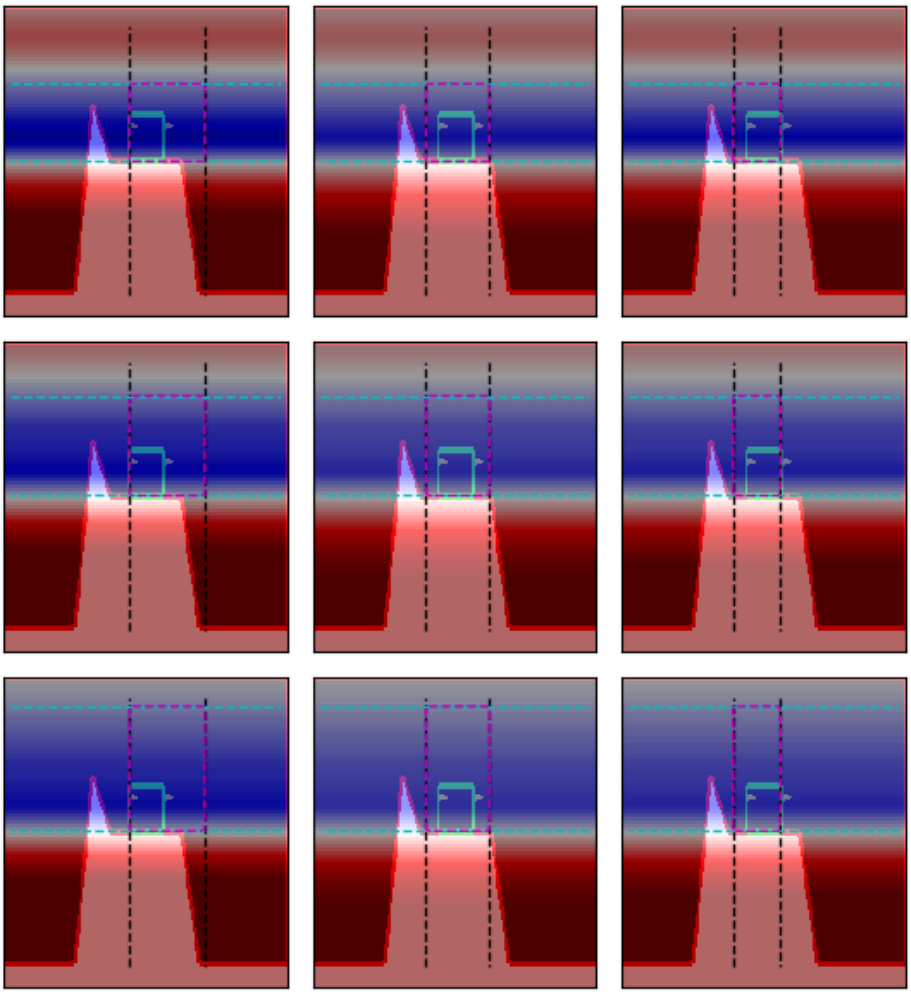
updates με αρχικό γ 0.9 και τελικό-μέγιστο 0.999. Το learning rate αυξάνεται ανά 500000 gradient updates με decay 0.1, δηλαδή το γ παίρνει τις τιμές $\{0.9, 0.99, 0.999\}$. Το πράσινο ορθογώνιο παραλληλόγραμμο δηλώνει το στόχο $\mathcal{T}$, το μπλε αναδεικνύει τον επιτρεπόμενο χώρο P_{cx} που επιτρέπεται να κινηθεί το χ-υποσύστημα, z_1, και το κόκκινο το failure set που σκοπός είναι να αποφευχθεί. Η περιοχή ανάμεσα στις μαύρες διακεκομμένες σηματοδοτεί τον επιτρεπόμενο χώρο P_{cy} που επιτρέπεται να κινηθεί το ψ-υποσύστημα, z_2. Το εσωτερικό του ροζ παραλληλογράμμου σηματοδοτεί τον συνολικό επιτρεπόμενο χώρο P_c. Οι 9 εικόνες αποτυπώνουν slices της συνάρτησης αξίας για διαφορετικές τιμές του v_x και v_y, με $\theta = 0$ και $\dot{\theta} = 0$. Επάνω, μεσαία και κάτω σειρά αντιστοιχούν σε $v_y = 1, 0, -1$ αντίστοιχα. Πρώτη, δεύτερη και τελευταία στήλη αντιστοιχούν σε $v_x = -1, 0, 1$ αντίστοιχα.

Ο κώδικας έτρεξε για 2 ώρες και 25 λεπτά στην ίδια κάρτα γραφικών NVIDIA GeForce GTX 1050 και 4 ώρες και 52 λεπτά στον ίδιο υπολογιστή dedicated 8GB RAM, 4-core CPU, hosted στο Linode.



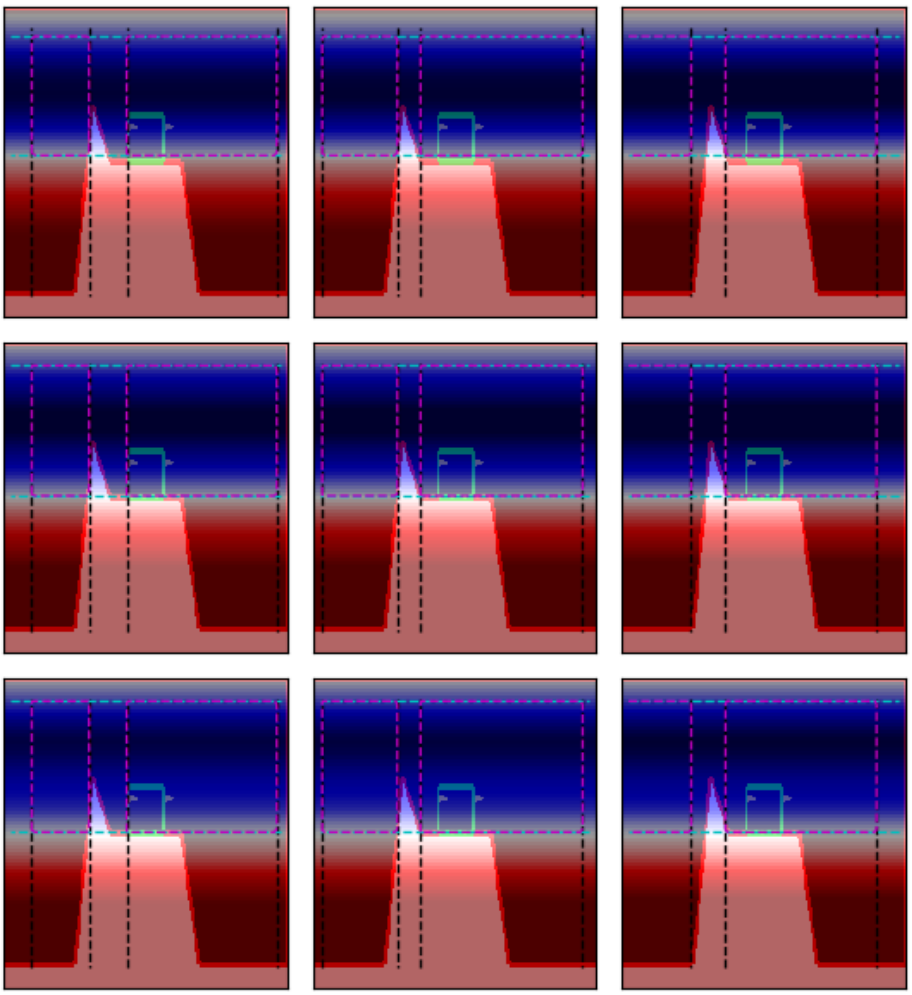
Σχήμα 2.5: Υπολογισμός reach-avoid set για το διασπασμένο σύστημα 6Δ σεληνακάτου. Ο ρυθμός μάθησης είναι 0.9.

Ο ρυθμός μάθησης 0.9 δεν είναι αρκετός για να υπολογιστεί σωστά το reach-avoid set.



Σχήμα 2.6: Υπολογισμός reach-avoid set για το διασπασμένο σύστημα 6Δ σεληνακάτου. Ο ρυθμός μάθησης είναι 0.99.

Ο ρυθμός μάθησης 0.99 δεν είναι αρκετός για να υπολογίσει σωστά το reach-avoid set, ωστόσο το P_c ξεκινάει να σχηματίζεται.



Σχήμα 2.7: Υπολογισμός reach-avoid set για το διασπασμένο σύστημα 6Δ σεληνακάτου. Ο ρυθμός μάθησης είναι 0.999.

Ο ρυθμός μάθησης 0.999 είναι ο βέλτιστος. Το zero level set στον οριζόντιο άξονα έχει πετύχει το καλύτερο δυνατό αποτέλεσμα, όντας πολύ κοντά στο έδαφος στο σημείο που βρίσκεται ο στόχος, αλλά και στο πάνω μέρος. Το αντίστοιχο στον κατακόρυφο άξονα έχει πλήρως απομονώσει την κορυφή στο βουνό αυτό της σελήνης, προσδίδοντας επίσης βέλτιστα αποτελέσματα. Ο επιτρεπόμενος χώρος είναι το σύνολο που απαρτίζεται από τα δύο ροζ πολύγωνα. Το reach-avoid set έχει σχηματιστεί πλέον.

Αν το γ φτάσει τιμές μεγαλύτερες του 0.999, το αποτέλεσμα δεν είναι ιδιαίτερα διαφορετικό για αυτό και δεν συμπεριλαμβάνεται στις απεικονίσεις.

Στην κάρτα γραφικών NVIDIA GeForce GTX 1050, το πλήρες reach-avoid set ανακατασκευάστηκε 61% ταχύτερα, στο 39% του χρόνου της πλήρους μεθόδου! Όσον αφορά στη μνήμη RAM 8GB, CPU 4 πυρήνων, η λύση βρέθηκε στο 46.7% του αρχικού χρόνου, με επιτάχυνση 53,3%.

Προφανώς η ανακατασκευή του reach-avoid set του πλήρους συστήματος υπεκτιμήθηκε, διότι για την άφιξη στην τομή των δύο υποσυστημάτων το decomposition δεν

λειτουργεί (Σχήμα 1.1), οπότε καταφεύγει κανείς στην προσεγγιστική αποσύνθεση. Ωστόσο, σε τόσο σημαντική διαφορά χρόνου, το reach-avoid σύνολο ανακατασκευάστηκε με ακρίβεια περίπου 43%.

Κεφάλαιο 3

Εκτεταμένη Περίληψη

Περιγραφή Προβλήματος

Θεωρούμε ένα δυναμικό σύστημα ενός προς τα εμπρός κινούμενου πράκτορα με μετρήσιμη κατάσταση $x(t) \in X \subseteq \mathbb{R}^n$, είσοδο ελέγχου $u(t) \in U \subseteq \mathbb{R}^m$, και διαταραχή $w(t) \in W \subseteq \mathbb{R}^p$, το οποίο εξελίσσεται σύμφωνα με την ακόλουθη δυναμική εξίσωση

$$\dot{x}(t) = Ax(t) + Bu(t) + Cw(t), x(0) = x_0, t \geq 0, \tag{3.1}$$

όπου $A \in \mathbb{R}^{n \times n}$ είναι ο άγνωστος plant πίνακας, $B \in \mathbb{R}^{m \times m}$ είναι ο άγνωστος πίνακας εισόδου και $C \in \mathbb{R}^{p \times p}$ είναι ο άγνωστος πίνακας διαταραχών.

Σημειώνεται ότι το σύστημα (3.1) μπορεί είτε να εξαρτάται από το χρόνο είτε όχι. Η μέθοδος που προτείνεται στην παρούσα εργασία είναι ανεξάρτητη από το χρόνο, αλλά μπορεί κανείς εύκολα να προσαρμόσει το πλαίσιο αυτό σε συστήματα που μεταβάλλονται στο χρόνο.

Ο στόχος είναι να υπολογιστεί το Reachable Set (Προσπελάσιμο Σύνολο) G στο τέλος του χρονικού ορίζοντα $T \in \mathbb{R}_+$. Το προσιτό σύνολο του (3.1) στον τελικό χρόνο από ένα σύνολο αρχικών καταστάσεων $X_0 \subseteq \mathbb{R}^n$ είναι:

$$\begin{aligned} G(X_0) = \{s \in X \mid & x_0 \in X_0, u(\tau) \in U, w(\tau) \in W, \\ & \text{τ.ω. } x(T) = s \text{ και } x(\tau) \notin X_{obs}, \forall \tau \in [0, T]\} \end{aligned} \tag{3.2}$$

όπου X_{obs} είναι το σύνολο που περιγράφει όλα τα εμπόδια στο περιβάλλον.

Με άλλα λόγια, το τελικό reachable set αποτελείται από όλες τις καταστάσεις που μπορούν να επιτευχθούν τη χρονική στιγμή T, από οποιαδήποτε αρχική κατάσταση στο X_0, για οποιαδήποτε αποδεκτή εξωτερική είσοδο ή διαταραχή, δεδομένου ότι ο πράκτορας δεν θα έρθει σε επαφή με κάποιο εμπόδιο.

Μέθοδος finite-horizon βέλτιστου ελέγχου

Συμβολίζουμε τη διαφορά μεταξύ της τρέχουσας κατάστασης $x(t)$ και της τελικής κατάστασης $x(T)$ ως τη νέα μας κατάσταση $\bar{x}(t) := x(t) - x(T)$. Ορίζοντας την τελική κατάσταση ως $x_f := x(T)$, μπορούμε να γράψουμε $\bar{x}(t) := x(t) - x_f$. Συνεπώς, ο έλεγχος και η διαταραχή γίνονται $\bar{u}(t) := u(t) - u_f$ και $\bar{w}(t) := w(t) - w_f$, αντίστοιχα, θεωρώντας $u_f := u(T)$ και $w_f := w(T)$.

Το ενημερωμένο δυναμικό σύστημα γίνεται

$$\begin{aligned}\dot{\bar{x}}(t) =& \dot{x}(t) - \dot{x_f} \\ =& A\bar{x}(t) + B\bar{u}(t) + C\bar{w}(t), \bar{x}(0) = x_0 - x_f, t \geq 0.\end{aligned} \tag{3.3}$$

Στόχος είναι η επίτευξη της βέλτιστης συνάρτησης αξίας (value function)

$$V^\star(\bar{x}; t_0, T) := \max_{\bar{w}} \min_{\bar{u}} J(\bar{x}; \bar{u}; \bar{w}; t_0, T) \tag{3.4}$$

για το άγνωστο σύστημα. $J(\bar{x}; \bar{u}; \bar{w}; t_0, T)$ είναι η συνάρτηση κόστους που ορίζεται ως εξής

$$\begin{aligned}J(\bar{x}; \bar{u}; \bar{w}; t_0, T) = \Phi(T) \\ + \frac{1}{2} \int_{t_0}^{T} (\bar{x}^{\mathrm{T}} M \bar{x} + \bar{u}^{\mathrm{T}} R \bar{u} - \bar{w}^{\mathrm{T}} F \bar{w}) d\tau\end{aligned} \tag{3.5}$$

όπου $\Phi(T) := (1/2)\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T)$ είναι το τελικό κόστος με $P(T) := P_T \in \mathbb{R}^{n \times n} > 0$ να είναι ο τελευταίος πίνακας Riccati και $M \in \mathbb{R}^{n \times n} \geq 0$, $R \in \mathbb{R}^{m \times m} > 0$ και $F \in \mathbb{R}^{p \times p} > 0$ είναι οι πίνακες κατάστασης, εισόδου ελέγχου και διαταραχής, αντίστοιχα, οι οποίοι ορίζονται από το χρήστη. Θα υποθέσουμε εδώ ότι το σύστημα είναι ανιχνεύσιμο και ελέγξιμο.

Μας ενδιαφέρει ουσιαστικά να βρούμε έναν βέλτιστο έλεγχο $\bar{u}^\star$ και μια βέλτιστη διαταραχή $\bar{w}^\star$ τέτοια ώστε $J(\bar{x}; \bar{u}^\star; t_0, T) \leq J(\bar{x}; \bar{u}, t_0, T), \forall \bar{x}, \bar{u}$ και $J(\bar{x}; \bar{w}^\star; t_0, T) \geq J(\bar{x}; \bar{w}; t_0, T), \forall \bar{x}, \bar{w}$, αντίστοιχα, οι οποίες μπορούν να περιγραφούν ως το διαφορικό παίγνιο που φαίνεται στο (3.4). Το αποτέλεσμα αυτού του παιγνίου θα παράγει το *Reachable Set* στο τέλος του χρονικού ορίζοντα λόγω της οριακής συνθήκης [31]:

$$V^\star(\bar{x}(T); T) = \bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T), \forall \bar{x}(T) \in G \tag{3.6}$$

όπου G θα είναι το προσιτό σύνολο. Ένα τέτοιο πρόβλημα έχει λύση μόνο αν $x_f \approxeq 0$ [32]. Στην παρούσα εργασία θα προταθεί μια προσέγγιση χωρίς μοντέλο για την εύρεση του προσβάσιμου συνόλου G.

Για την επίλυση του προβλήματος βέλτιστου ελέγχου πεπερασμένου ορίζοντα που περιγράφεται από την (3.4), εξάγουμε την ακόλουθη εξίσωση Hamilton-Jacobi-

Bellman (HJB):

$$-\frac{\partial V^\star}{\partial t} = \frac{1}{2}(\bar{x}^{\mathrm{T}} M \bar{x} + \bar{u}^{*\mathrm{T}} R \bar{u}^\star - \bar{w}^{*\mathrm{T}} F \bar{w}^\star) + \lambda^{\mathrm{T}}(A\bar{x} + B\bar{u}^\star + C\bar{w}^\star), \forall \bar{x} \tag{3.7}$$

αφού η Χαμιλτονιανή $\mathcal{H}$ σε σχέση με (3.3),(3.4) και (3.5) ορίζεται ως εξής

$$\mathcal{H}(\bar{x}; \bar{u}; \bar{w}; \lambda) = \frac{1}{2}(\bar{x}^{\mathrm{T}} M \bar{x} + \bar{u}^{\mathrm{T}} R \bar{u} - \bar{w}^{\mathrm{T}} F \bar{w}) + \lambda^{\mathrm{T}}(A\bar{x} + B\bar{u} + C\bar{w}), \forall \bar{x}, \bar{u}, \bar{w}, \lambda \tag{3.8}$$

Κάνοντας χρήση της μεθόδου σάρωσης (sweep method) [33] και $\lambda = \frac{\partial V^\star}{\partial x}$, η εξίσωση (3.7) γίνεται

$$-\frac{\partial V^\star}{\partial t} = \frac{1}{2}(\bar{x}^{\mathrm{T}} M \bar{x} + \bar{u}^{*\mathrm{T}} R \bar{u}^\star - \bar{w}^{*\mathrm{T}} F \bar{w}^\star) + \frac{\partial V^{\star\,\mathrm{T}}}{\partial x}(A\bar{x} + B\bar{u}^\star + C\bar{w}^\star), \forall \bar{x} \tag{3.9}$$

Λαμβάνοντας υπόψη ότι το σύστημα (3.1) είναι γραμμικό και η διαταραχή μεγιστοποιεί τη συνάρτηση κόστους LQR (3.5), η συνάρτηση αξίας θα έχει τετραγωνική μορφή στην κατάσταση $\bar{x}$, όπως φαίνεται παρακάτω.

$$V^\star(\bar{x}; t) = \frac{1}{2}\bar{x}^{\mathrm{T}} P(t) \bar{x}, \forall \bar{x}, t. \tag{3.10}$$

όπου $P(t) \in \mathbb{R}^{n \times n}$ είναι η λύση [34] της εξίσωσης Riccati

$$-\dot{P}(t) = M + P(t)A + A^{\mathrm{T}} P(t) + P(t)(CF^{-1}C^{\mathrm{T}} - BR^{-1}B^{\mathrm{T}})P(t) \tag{3.11}$$

Έτσι, ο βέλτιστος έλεγχος και η διαταραχή είναι αντίστοιχα:

$$\bar{u}^\star(\bar{x}; t) = -R^{-1}B^{\mathrm{T}} P(t)\bar{x}, \forall \bar{x}, t. \tag{3.12}$$

$$\bar{w}^\star(\bar{x}; t) = F^{-1}C^{\mathrm{T}} P(t)\bar{x}, \forall \bar{x}, t. \tag{3.13}$$

Για να χειριστεί κανείς τη μέθοδο επίλυσης του συστήματος (3.1) ως διαφορικό παίγνιο, θα έπρεπε να ξαναγράψει τις παραπάνω εξισώσεις ως εξής

$$\bar{u}^\star(\bar{x}; t) = \arg \min_{u \in U} \max_{w \in W} \mathcal{H}(\bar{x}; \bar{u}; \bar{w}; \lambda) \tag{3.14}$$

$$\bar{w}^\star(\bar{x}; t) = \arg \max_{w \in W} \min_{u \in U} \mathcal{H}(\bar{x}; \bar{u}; \bar{w}; \lambda) \tag{3.15}$$

Model-free Actor-Critic Μέθοδος

Ορίζουμε την ακόλουθη συνάρτηση $\mathcal{Q} : \mathbb{R}^{(n+m+p)} \times \mathbb{R}^{(n+m+p)\times(n+m+p)} \to \mathbb{R}^+$:

$$\begin{aligned}\mathcal{Q}(\bar{x};\bar{u};\bar{w};t) &:= V^{\star}(\bar{x};t) + \mathcal{H}(\bar{x};\bar{u};\bar{w};\frac{\partial V^{\star}}{\partial t},\frac{\partial V^{\star}}{\partial x}) \\ &= V^{\star}(\bar{x};t) + \frac{1}{2}\bar{x}^{\mathrm{T}}M\bar{x} + \frac{1}{2}\bar{u}^{\mathrm{T}}R\bar{u} \\ &- \frac{1}{2}\bar{w}^{\mathrm{T}}F\bar{w} + \bar{x}^{\mathrm{T}}P(t)(A\bar{x}+B\bar{u}+C\bar{w}) \\ &+ \frac{1}{2}\bar{x}^{\mathrm{T}}\dot{P}(t)\bar{x}, \forall \bar{x},\bar{u},\bar{w},t. \end{aligned} \tag{3.16}$$

Στη συνέχεια ορίζουμε την επαυξημένη κατάσταση $U := [\bar{x}^{\mathrm{T}}\bar{u}^{\mathrm{T}}\bar{w}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{(n+m+p)}$ για να ξαναγραφεί σε συμπαγή μορφή η συνάρτηση πλεονεκτήματος που εξαρτάται από τη δράση (action-dependent value function) [35]. Πιο συγκεκριμένα,

$$\mathcal{Q}(\bar{x};\bar{u};\bar{w};t) = \frac{1}{2}U^{\mathrm{T}}\bar{\mathcal{Q}}(t)U \tag{3.17}$$

όπου

$$\bar{\mathcal{Q}}(t) = \begin{bmatrix} Q_{\mathrm{xx}}(t) & Q_{\mathrm{xu}}(t) & Q_{\mathrm{xw}}(t) \\ Q_{\mathrm{ux}}(t) & Q_{\mathrm{uu}}(t) & Q_{\mathrm{uw}}(t) \\ Q_{\mathrm{wx}}(t) & Q_{\mathrm{wu}}(t) & Q_{\mathrm{ww}}(t) \end{bmatrix} \tag{3.18}$$

Κάνοντας χρήση των συμμετρικών ιδιοτήτων του πίνακα Riccati, $\bar{x}^{\mathrm{T}}P(t)A\bar{x} = (1/2)\bar{x}^{\mathrm{T}}(P(t)A + A^{\mathrm{T}}P(t))\bar{x}$, και για τους ελέγχους $\bar{x}^{\mathrm{T}}P(t)B\bar{u} = (1/2)\bar{x}^{\mathrm{T}}(P(t)B + B^{\mathrm{T}}P(t))\bar{u}$ και $\bar{x}^{\mathrm{T}}P(t)C\bar{w} = (1/2)\bar{x}^{\mathrm{T}}(P(t)C + C^{\mathrm{T}}P(t))\bar{w}$. Έτσι, κάθε μεμονωμένο Q_i έχει ως εξής:

$Q_{\mathrm{xx}}(t) = \dot{P}(t) + P(t) + M + P(t)A + A^{\mathrm{T}}P(t) + P(t)B + P(t)C$, $Q_{\mathrm{xu}}(t) = Q_{\mathrm{ux}}(t)^{\mathrm{T}} = P(t)B$, $Q_{\mathrm{xw}}(t) = Q_{\mathrm{wx}}(t)^{\mathrm{T}} = P(t)C$, $Q_{\mathrm{uu}}(t) = R$, $Q_{\mathrm{uw}}(t) = Q_{\mathrm{wu}}(t)^{\mathrm{T}} = 0$, $Q_{\mathrm{uu}}(t) = F$.

Τα $u^{\star}$ και $w^{\star}$ των εξισώσεων (3.12) και (3.13) μπορούν να διατυπωθούν με model-free τρόπο χρησιμοποιώντας τις στατικές συνθήκες $\partial\mathcal{Q}(\bar{x};\bar{u};\bar{w};t)/\partial\bar{u} = 0$ και $\partial\mathcal{Q}(\bar{x};\bar{u};\bar{w};t)/\partial\bar{w} = 0$. Λαμβάνουμε

$$\bar{u}^{\star}(\bar{x};t) = \arg\min_{\bar{u}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) = -Q_{\mathrm{uu}}^{-1}Q_{\mathrm{ux}}(t)\bar{x}. \tag{3.19}$$

$$\bar{w}^{\star}(\bar{x};t) = \arg\max_{\bar{w}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) = Q_{\mathrm{ww}}^{-1}Q_{\mathrm{wx}}(t)\bar{x}. \tag{3.20}$$

Στην παρούσα εργασία, θα προταθεί μια προσέγγιση διαφορικών παιγνίων ως λύση του συστήματος (3.1), επομένως οι παραπάνω εξισώσεις θα γίνουν τώρα

$$\bar{u}^{\star}(\bar{x};t) = \arg\min_{\bar{u}}\max_{\bar{w}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) \tag{3.21}$$

$$\bar{w}^{\star}(\bar{x};t) = \arg\max_{\bar{w}}\min_{\bar{u}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) \tag{3.22}$$

Λήμμα 1: Η αξία του παιγνίου $\mathcal{Q}^{\star}(\bar{x};\bar{u}^{\star};\bar{w}^{\star};t) = \max_{\bar{w}}\min_{\bar{u}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t)$ ταυ-

τίζεται με τη βέλτιστη τιμή $V^\star$ στο (3.10) του διαφορικού παιγνίου (3.4), όπου $P(t)$ είναι ο πίνακας Riccati που βρέθηκε από την επίλυση της εξίσωσης (3.11).

Απόδειξη. : Αρκεί κανείς να αντικαταστήσει τις εξισώσεις (3.14) και (3.15) στη συνάρτηση $\mathcal{Q}$ (3.16) για να λάβει την (3.11). Επομένως, $\mathcal{Q}(\bar{x}; \bar{u}^\star; \bar{w}^\star; t) = V^\star(\bar{x}; t)$ □

3.0.1 Actor-Critic Αρχιτεκτονική

Αξιοποιώντας τις συμμετρικές ιδιότητες του πίνακα $\bar{\mathcal{Q}}$, μπορούμε να υπολογίσουμε επιτυχώς τη συνάρτηση $\mathcal{Q}$ στην (3.16) ως εξής

$$\mathcal{Q}^\star(\bar{x}; \bar{u}^\star; \bar{w}^\star; t) = \frac{1}{2} U^{\mathrm{T}} \bar{\mathcal{Q}}(t) U = \frac{1}{2} vech(\bar{\mathcal{Q}}(t))^{\mathrm{T}} (U \otimes U) \tag{3.23}$$

όπου $vech(\bar{\mathcal{Q}}(t)) \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)}$ είναι η πράξη μισής διανυσματοποίησης, η οποία μειώνει σημαντικά την υπολογιστική πολυπλοκότητα. Ορίζουμε τώρα τον όρο $v(t)^{\mathrm{T}} W_c := \frac{1}{2} vech(\bar{\mathcal{Q}}(t))$, που προσεγγίζει την Q-function ως εξής

$$\mathcal{Q}^\star(\bar{x}; \bar{u}^\star; \bar{w}^\star; t) = W_c^{\mathrm{T}} v(t) (U \otimes U) \tag{3.24}$$

με $W_c \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)}$ να είναι οι εκτιμήσεις βάρους από τον κριτή και $v(t) \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2) \times ((n+m+p)(n+m+p+1)/2)}$ μια φραγμένη συνάρτηση ακτινικής βάσης, αποκλειστικά εξαρτώμενη από το χρόνο.

Στόχος μας είναι να βρούμε τις ιδανικές εκτιμήσεις βαρών, επομένως εφαρμόζουμε μια προσαρμοστική τεχνική εκτίμησης που χρησιμοποιεί τα τρέχοντα βάρη, όπως φαίνεται στην [36]. Κατά συνέπεια, έχουμε

$$\hat{\mathcal{Q}}(\bar{x}; \bar{u}; \bar{w}; t) = \hat{W_c}^{\mathrm{T}} v(t) (U \otimes U) \tag{3.25}$$

όπου $\hat{W_c}^{\mathrm{T}} v(t) := \frac{1}{2} vech(\bar{\hat{Q}}(t))$.

Όσον αφορά στη δομή του actor, δημιουργούμε δύο actor instances, έναν για τον έλεγχο και έναν για τη διαταραχή. Παρόμοια με το critic, ανατίθεται $\mu(t)^{\mathrm{T}} W_{a_1} := -Q_{\mathrm{uu}}^{-1} Q_{\mathrm{ux}}(t) \bar{x}$ και $\mu(t)^{\mathrm{T}} W_{a_2} := Q_{\mathrm{ww}}^{-1} Q_{\mathrm{wx}}(t) \bar{x}$ για να γραφεί

$$\bar{u}^\star(\bar{x}; t) = W_{a_1}^{\mathrm{T}} \mu(t) \bar{x}. \tag{3.26}$$

$$\bar{w}^\star(\bar{x}; t) = W_{a_2}^{\mathrm{T}} \mu(t) \bar{x}. \tag{3.27}$$

όπου $W_{a_1} \in \mathbb{R}^{n \times m}$ είναι οι εκτιμήσεις βαρών του ελεγκτή, $W_{a_2} \in \mathbb{R}^{n \times p}$ οι εκτιμήσεις βάρους του παράγοντα διαταραχής και $\mu(t) \in \mathbb{R}^{n \times n}$ είναι μια άλλη περιορισμένη ακτινική συνάρτηση που εξαρτάται μόνο από το χρόνο. Οι actors τότε γίνονται

$$\hat{\bar{u}}(\bar{x}; t) = \hat{W_{a_1}^{\mathrm{T}}} \mu(t) \bar{x}. \tag{3.28}$$

$$\hat{\bar{w}}(\bar{x}; t) = \hat{W_{a_2}^{\mathrm{T}}} \mu(t) \bar{x}. \tag{3.29}$$

Πρέπει εδώ να σημειωθεί ότι αυτή η δομή αφορά σε ολόκληρο το χώρο και όχι μόνο ένα συμπαγές σύνολο. Αυτό συμβαίνει καθώς οι critic approximators που περιγράφονται στις εξισώσεις (3.25) - (3.29) δεν χαρακτηρίζονται από σφάλματα προσέγγισης, οπότε επιτυγχάνονται βέλτιστες πολιτικές.

Η εξίσωση Bellman αναπαρίσταται χρησιμοποιώντας την ολοκληρωτική προσέγγιση ενισχυτικής μάθησης (integral reinforcement learning approach) από το [37] ως εξής

$$V^\star(\bar{x}(t);t) = V^\star(\bar{x}(t-\Delta t);t-\Delta t) - \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^{\mathrm{T}}M\bar{x} + \hat{\bar{u}}^{\mathrm{T}}R\hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}}F\hat{\bar{w}})d\tau \tag{3.30}$$

$$V^\star(\bar{x}(T);T) = \frac{1}{2}\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T), \tag{3.31}$$

με $\Delta t \in \mathbb{R}^+$ να είναι μια μικροσκοπική σταθερή τιμή.

Χρησιμοποιώντας το παραπάνω Λήμμα, όπου απεδείχθη ότι $\mathcal{Q}^\star(\bar{x};\hat{\bar{u}}^\star;\hat{\bar{w}}^\star;t) = V^\star(\bar{x};t)$, οι παραπάνω εξισώσεις μπορούν να ξαναγραφούν ως εξής

$$Q^\star(\bar{x}(t);\hat{\bar{u}}^\star(t);\hat{\bar{w}}^\star(t);t) = Q^\star(\bar{x}(t-\Delta t);\hat{\bar{u}}^\star(t-\Delta t);\hat{\bar{w}}^\star(t-\Delta t);t-\Delta t) - \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^{\mathrm{T}}M\bar{x} + \hat{\bar{u}}^{\mathrm{T}}R\hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}}F\hat{\bar{w}})d\tau \tag{3.32}$$

$$Q^\star(\bar{x}(T);\hat{\bar{u}}^\star(T);\hat{\bar{w}}^\star(T);T) = \frac{1}{2}\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T) \tag{3.33}$$

Από το (3.6) και από το παραπάνω Λήμμα, είναι προφανές ότι το reachable set G μπορεί να αποκτηθεί άμεσα από το $Q^\star(\bar{x}(T);\hat{\bar{u}}^\star(T);\hat{\bar{w}}^\star(T);T)$.

Στη συνέχεια επιλέγουμε τα σφάλματα e_{c1} και e_{c2}, τα οποία θέλουμε να μηδενίσουμε με κατάλληλη προσαρμογή των κριτικών βαρών του (3.25). Το αρχικό σφάλμα του critic, $e_{c1} \in \mathbb{R}$ περιγράφεται ως εξής

$$\begin{aligned} e_{c1} :=\ & \hat{\mathcal{Q}}(\bar{x};\hat{\bar{u}};\hat{\bar{w}};t) \\ & - \hat{Q}(\bar{x}(t-\Delta t);\hat{\bar{u}}(t-\Delta t);\hat{\bar{w}}(t-\Delta t);t-\Delta t) \\ & + \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^{\mathrm{T}}M\bar{x} + \hat{\bar{u}}^{\mathrm{T}}R\hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}}F\hat{\bar{w}})d\tau \\ & = \hat{W}_c^{\mathrm{T}}v(t)((\hat{U}(t)\otimes\hat{U}(t)) - (\hat{U}(t-\Delta t)\otimes\hat{U}(t-\Delta t)) \\ & + \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^{\mathrm{T}}M\bar{x} + \hat{\bar{u}}^{\mathrm{T}}R\hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}}F\hat{\bar{w}})d\tau \end{aligned} \tag{3.34}$$

όπου $\hat{U} := [\bar{x}^{\mathrm{T}}\hat{\bar{u}}^{\mathrm{T}}\hat{\bar{w}}^{\mathrm{T}}]^{\mathrm{T}}$ είναι η επαυξημένη κατάσταση που αποτελείται από το μετρήσιμο διάνυσμα της πλήρους κατάστασης και τον εκτιμώμενο έλεγχο και τη διαταραχή. Όσον αφορά στο δεύτερο critic error, εφεξής αναφερόμενο ως *τελικό*

σφάλμα κριτικού (final critic error), ορίζεται ως η πραγματική τιμή

$$e_{cf} := \frac{1}{2}\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T) - \hat{W}_c^{\mathrm{T}}v(T)((\hat{U}(T) \otimes \hat{U}(T)) \tag{3.35}$$

Στη συνέχεια, δηλώνουμε τα σφάλματα προσέγγισης για τους δύο actors, τον έλεγχο και τη διαταραχή, ως $e_{a1} \in \mathbb{R}^m, e_{a2} \in \mathbb{R}^p$ αντίστοιχα. Πιο συγκεκριμένα,

$$e_{a1} := \hat{W}_{a_1}^{\mathrm{T}}\mu(t)\bar{x} + \hat{Q}_{\mathrm{uu}}^{-1}\hat{Q}_{\mathrm{ux}}(t)\bar{x} \tag{3.36}$$

$$e_{a2} := \hat{W}_{a_2}^{\mathrm{T}}\mu(t)\bar{x} - \hat{Q}_{\mathrm{ww}}^{-1}\hat{Q}_{\mathrm{wx}}(t)\bar{x} \tag{3.37}$$

με τα $\hat{Q}_{\mathrm{uu}}$, $\hat{Q}_{\mathrm{ux}}$, $\hat{Q}_{\mathrm{ww}}$, $\hat{Q}_{\mathrm{wx}}$ να προκύπτουν από την εκτίμηση του κριτή $\hat{W}_c$.

Η τετραγωνική νόρμα των σφαλμάτων, ακολουθώντας τη μέθοδο adaptive ελέγχου του [36], μπορεί να ληφθεί ως εξής

$$N_1(\hat{W}_c, \hat{W}_c(T)) = \frac{1}{2}||e_{c1}||^2 + \frac{1}{2}||e_{cf}||^2 \tag{3.38}$$

$$N_2(\hat{W}_{a_1}) = \frac{1}{2}||e_{a1}||^2 \tag{3.39}$$

$$N_3(\hat{W}_{a_2}) = \frac{1}{2}||e_{a2}||^2 \tag{3.40}$$

Έτσι, το learning framework μπορεί να υλοποιηθεί με τη χρήση ενός αλγορίθμου gradient descent στο (3.38), με τον ακόλουθο κανονικοποιημένο τρόπο

$$\begin{aligned}\dot{\hat{W}}_c &= -\alpha_c \frac{\partial N_1}{\partial \hat{W}_c} \\ &= -\alpha_c \left(\frac{\sigma e_{c1}}{(1+\sigma^T\sigma)^2} + \frac{\sigma_f e_{c2}}{(1+\sigma_f^T\sigma_f)^2} \right)\end{aligned} \tag{3.41}$$

με $\alpha_c \in \mathbb{R}^+$ να είναι ο ρυθμός σύγκλισης που καθορίζει ο σχεδιαστής για το gradient descent και $\sigma(t) := v(t)(\hat{U}(t) \otimes \hat{U}(t) - (\hat{U}(t-\Delta t) \otimes \hat{U}(t-\Delta t))$, $\sigma(T) := v(T)(\hat{U}(T) \otimes \hat{U}(T)$. Καθώς τα e_{c1} και e_{c2} τείνουν στο 0, η ενημέρωση του κριτικού (3.41) εξασφαλίζει ότι $\hat{W}_c \to W_c$ και $\hat{W}_c(T) \to W_c(T)$.

Ομοίως, εφαρμόζοντας μια κανονικοποιημένη gradient descent προσέγγιση στα (3.39) - (3.40) προκύπτουν τα ακόλουθα αποτελέσματα ρυθμίσεων των ηθοποιών

$$\dot{\hat{W}}_{a_1} = -\alpha_a \frac{\partial N_2}{\partial \hat{W}_{a_1}} = -\alpha_a \bar{x} e_{a1}^T \tag{3.42}$$

$$\dot{\hat{W}}_{a_2} = -\alpha_a \frac{\partial N_3}{\partial \hat{W}_{a_2}} = -\alpha_a \bar{x} e_{a2}^T \tag{3.43}$$

με $\alpha_a \in \mathbb{R}^+$ να είναι ένας άλλος ρυθμός σύγκλισης που καθορίζει ο σχεδιαστής για την κάθοδο κλίσης. Τα tuning των ηθοποιών (3.42) και (3.43) εγγυώνται ότι καθώς $e_{a1}, e_{a2} \to 0$, τότε $\hat{W}_{a_1} \to W_{a_1}$ και $\hat{W}_{a_2} \to W_{a_2}$, αντίστοιχα.

Ορίζουμε τώρα το σφάλμα εκτίμησης του κριτικού ως $\tilde{W}_c \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)}$ ως $\tilde{W}_c := W_c - \hat{W}_c$. Ομοίως, τα σφάλματα εκτίμησης των δύο actors $\tilde{W}_{a_1} \in \mathbb{R}^{n \times m}$, $\tilde{W}_{a_2} \in \mathbb{R}^{n \times p}$ ορίζονται ως $\tilde{W}_{a_1} := W_{a_1} - \hat{W}_{a_1}$ και $\tilde{W}_{a_2} := W_{a_2} - \hat{W}_{a_2}$, αντίστοιχα.

Η δυναμική του σφάλματος εκτίμησης του κριτικού είναι

$$\begin{aligned}
\dot{\tilde{W}}_c &= \left(\frac{\partial W_c}{\partial e_{c1}} \frac{de_{c1}}{\delta t} + \frac{\partial W_c}{\partial e_{cf}} \frac{de_{cf}}{\delta t} \right) - \left(\frac{\partial \hat{W}_c}{\partial e_{c1}} \frac{de_{c1}}{\delta t} + \frac{\partial \hat{W}_c}{\partial e_{cf}} \frac{de_{cf}}{\delta t} \right) \\
&= \frac{\partial W_c}{\partial e_{c1}} \frac{de_{c1}}{\delta t} - \frac{\partial \hat{W}_c}{\partial e_{c1}} \frac{de_{c1}}{\delta t} \\
&= -\alpha_c \frac{\sigma \sigma^T \tilde{W}_c}{(1 + \sigma^T \sigma)^2}
\end{aligned} \tag{3.44}$$

και οι αντίστοιχες δυναμικές των φορέων ελέγχου και διαταραχών είναι

$$\dot{\tilde{W}}_{a_1} = -\alpha_a \bar{x} \bar{x}^T \mu(t) \tilde{W}_{a_1} - \alpha_a \bar{x} \bar{x}^T \frac{\mu(t) \tilde{Q}_{\text{xu}} R^{-1}}{||1 + \mu(t)^T \mu(t)||^2} \tag{3.45}$$

$$\dot{\tilde{W}}_{a_2} = -\alpha_a \bar{x} \bar{x}^T \mu(t) \tilde{W}_{a_2} - \alpha_a \bar{x} \bar{x}^T \frac{\mu(t) \tilde{Q}_{\text{xw}} F^{-1}}{||1 + \mu(t)^T \mu(t)||^2} \tag{3.46}$$

όπου $\tilde{Q}_{\text{xu}} := \text{mat}(\tilde{W}_c[n(n+1)/2+1 : (n(n+1)/2+nm])$ και $\tilde{Q}_{\text{xw}} := \text{mat}(\tilde{W}_c[n(n+1)/2+1 : (n(n+1)/2+np])$.

Η προαναφερθείσα μεθοδολογία μπορεί να συνοψιστεί στον παρακάτω αλγόριθμο. $rand(x)$ είναι η τυχαία συνάρτηση με μήκος διανύσματος x. `pointsOfTheTarget` είναι τα σημεία που αποτελούν τον στόχο, `targetCoords(x)` είναι το διάνυσμα των συντεταγμένων x και y του στόχου για x. `Critic` είναι η συνάρτηση που ενημερώνει τις παραμέτρους του κριτικού στο (3.41), ενώ `Actor1` είναι η συνάρτηση που εκτιμά τις παραμέτρους του πρώτου ηθοποιού στην (3.42) και `Actor2` αυτή που εκτιμά τις παραμέτρους του δεύτερου ηθοποιού, στη (3.43). Οι συναρτήσεις `AdaptControl` και `AdaptDisturbance` εκτιμούν τον έλεγχο και τη διαταραχή, αντίστοιχα, σύμφωνα με τις εξισώσεις (3.28), (3.29), αντίστοιχα. Η `AdaptQ` εκτιμά τις παραμέτρους του $\hat{Q}$ από την (3.25), ενώ η `Augment` χρησιμοποιείται απλώς για την επαύξηση της τρέχουσας κατάστασης κατά την επίλυση της διαφορικής εξίσωσης. Τέλος, το `trajectory's last` είναι το τελευταίο ζεύγος συντεταγμένων x-y σε κάθε $trajectory$.

Αλγόριθμος Υπολογισμού Reachable Set

$G \leftarrow \emptyset$·
$W_c(0) \leftarrow rand(10)$·
$W_{a_1}(0) \leftarrow rand(2)$·
$W_{a_2}(0) \leftarrow rand(2)$·
$points \leftarrow$ `pointsOfTheTarget`·
$allTrajectories \leftarrow$ [ ]·
for $i = 1$ `to` $points$ **do**
    $x_f \leftarrow$ `targetCoords(`i`);`
    for $t \in T$ **do**
        $\hat{W}_c \leftarrow$ `Critic(`$\hat{x}, \hat{u}, \hat{w}, a_c, M, R, F, \Delta t, P(T)$`);`
        $\hat{Q} \leftarrow$ `AdaptQ(`$\hat{x}, \hat{u}, \hat{w}, \hat{W}_c$`);`
        $\hat{W}_{a_1} \leftarrow$ `Actor1(`$\hat{x}, \hat{u}, \hat{Q}, a_a$`);`
        $\hat{W}_{a_2} \leftarrow$ `Actor2(`$\hat{x}, \hat{w}, \hat{Q}, a_a$`);`
        $\hat{\hat{u}} \leftarrow$ `AdaptControl(`$\hat{x}, \hat{u}, \hat{W}_{a_1}$`);`
        $\hat{\hat{w}} \leftarrow$ `AdaptDisturbance(`$\hat{x}, \hat{w}, \hat{W}_{a_2}$`);`
        $\dot{x} \leftarrow$ `Augment(`$\hat{W}_c, \hat{W}_{a_1}, \hat{W}_{a_2}$`);`
        `Return` $\hat{\hat{u}}, \hat{\hat{w}}$;
    end for
    $trajectory \leftarrow \dot{x}(t)$;
    $allTrajectories \leftarrow [allTrajectories, trajectory]$;
    $G \leftarrow G \bigcup$ {`trajectory's last`};
end for
Return G;
Calculate minimum distance from x_0 to target using Euclidean Norm;
Plot Trajectory from x_0 to the closest point of the target using $allTrajectories$;

Προσομοιώσεις

Οι προσομοιώσεις μέχρι το τέλος του κεφαλαίου αυτού βρίσκονται στο GitHub repository[1] μου.

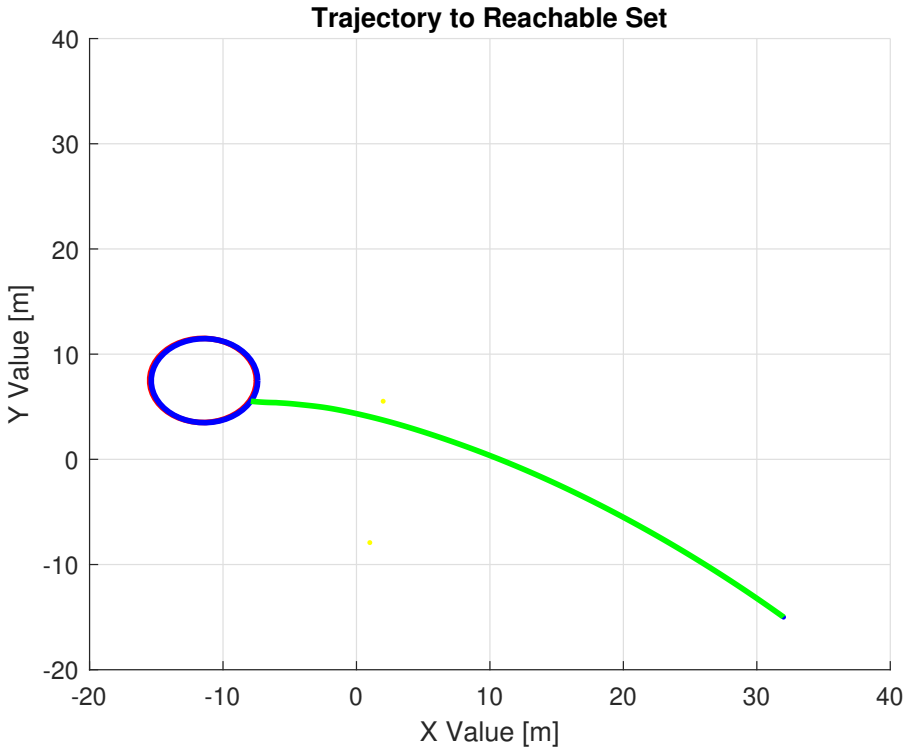
Σε αυτή την ενότητα, αποδεικνύεται η αποτελεσματικότητα του προτεινόμενου framework μέσω προσομοιώσεων. Γίνεται μελέτη απλών συστημάτων σημείου-στόχου, στα οποία ο πράκτορας-σημείο είναι εξοπλισμένος με τον online, χωρίς μοντέλο προτεινόμενο αλγόριθμο που στοχεύει στην προσέγγιση της βέλτιστης πολιτικής. Στη συνέχεια, υπολογίζεται η τροχιά προς το στόχο, αποφεύγοντας παράλληλα τυχόντα εμπόδια που μπορεί να υπάρχουν. Στις παρακάτω υποενότητες μπορεί κανείς να παρακολουθήσει την απόδοση του αλγορίθμου για συστήματα με 0, 1 και 3 εμπόδια, αντίστοιχα.

Όλες οι προσομοιώσεις χαρακτηρίζονται από πεπερασμένο χρονικό ορίζοντα T 6 sec, βήμα εσωτερικής δυναμικής $\Delta t = 0.05$ sec, καθώς και ρυθμούς σύγκλισης a_c, a_a 90 και 1.5 αντίστοιχα. Όσον αφορά στους πίνακες που ορίζονται από τον σχεδιαστή, ο πίνακας που κατάστασης είναι $M = I_2$, αυτός του ελέγχου είναι $R = 0.1$ και ο αντίστοιχος της διαταραχής είναι $F = 0.1$. Ο τελικός πίνακας Riccati είναι $P(T) = 0.5I_3$, ενώ οι τελικές δράσεις ελέγχου και διαταραχής $u(T), w(T)$, αντίστοιχα, είναι και οι δύο 0.005. Οι αρχικές τιμές των $\hat{W}_c, \hat{W}_{a_1}, \hat{W}_{a_2}$ επιλέγονται τυχαία, εκτός από τα δύο στοιχεία του $\hat{W}_c$ που αντιστοιχούν στα $Q_{\text{uu}}, Q_{\text{ww}}$, τα οποία πρέπει να διαφέρουν από το μηδέν, καθώς αντιστρέφονται στο (3.19) - (3.20).

3.0.2 Απουσία Εμποδίου

Έστω ένα απλό σύστημα σημείου-στόχου όπως αυτό που φαίνεται στην *Εικόνα 1*. Ο agent, που απεικονίζεται ως μπλε κουκκίδα, φτάνει με επιτυχία και σχεδόν βέλτιστα στο στόχο του, τον σκούρο μπλε κύκλο. Ο στόχος απεικονίζεται με κόκκινο χρώμα, αλλά το υπολογιζόμενο προσβάσιμο (μπλε) τον περιέχει τέλεια, οπότε δημιουργείται αυτό το σκούρο μπλε χρώμα.

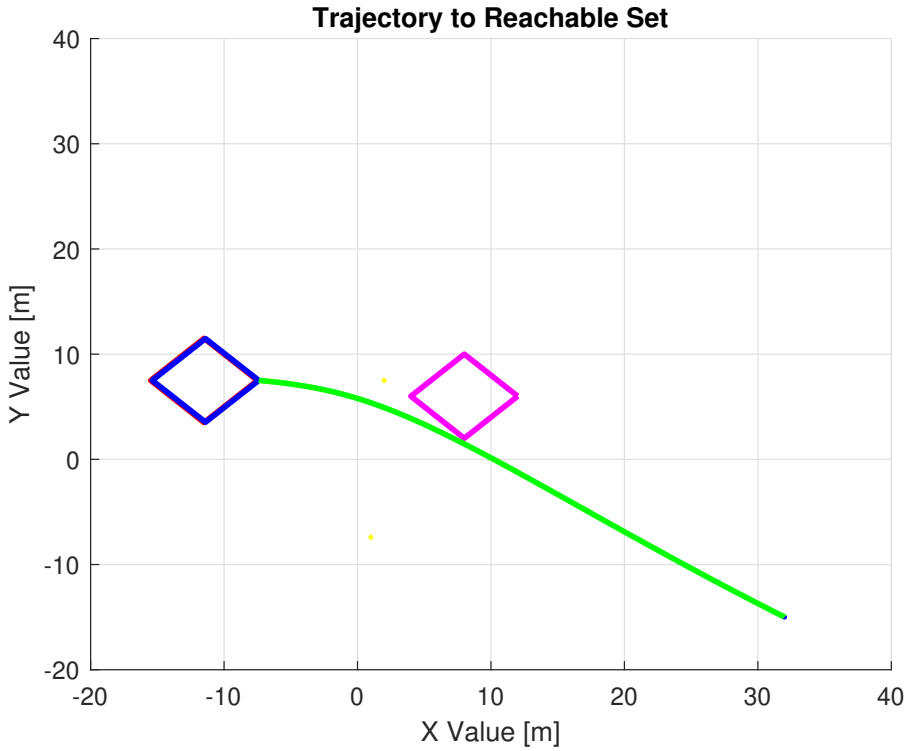
[1] https://github.com/Costopoulos/helperOC/tree/ReachabilityGame



Σχήμα 3.1: Τροχιά προς το Reachable Set. Το προσιτό σύνολο (μπλε) περιέχει απόλυτα τον στόχο (κόκκινο), εξ ου και το σκούρο μπλε χρώμα

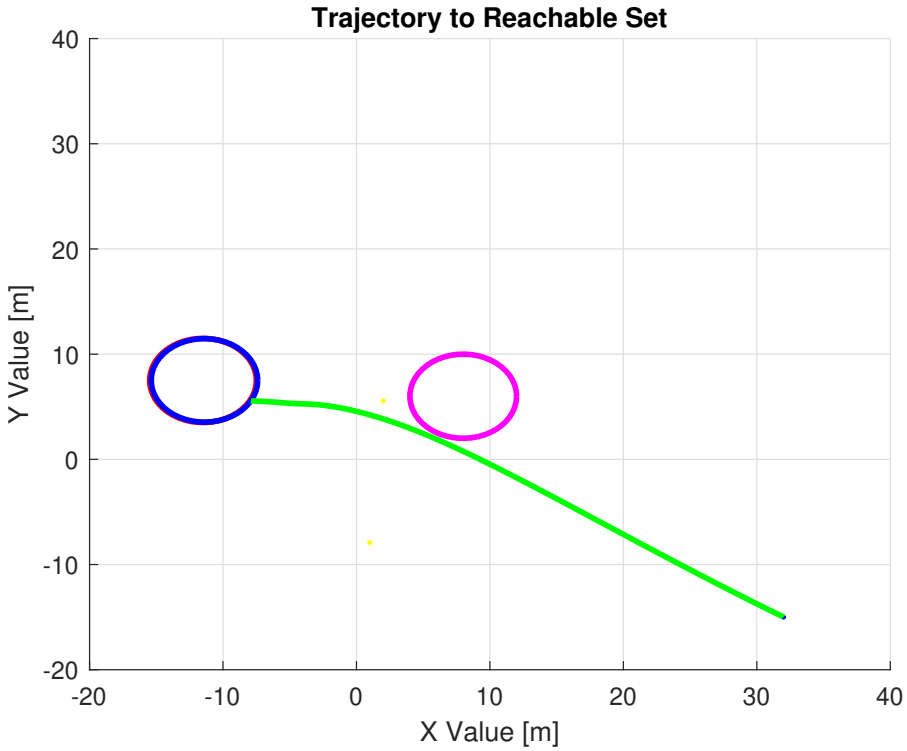
3.0.3 Μοναδικό Εμπόδιο

Εξετάζεται τώρα το ίδιο σύστημα, αλλά με ένα εμπόδιο που ανακόπτει τη διαδρομή. Στο *Σχήμα.2* τόσο ο στόχος όσο και το εμπόδιο είναι τετράγωνα και στο *Εικόνα.3* και τα δύο είναι κυκλικά, όπως ακριβώς και στο σχήμα 1.



Σχήμα 3.2: Τροχιά με ένα εμπόδιο σχήματος ρόμβου. Το προσιτό σύνολο (μπλε) περιέχει απόλυτα τον στόχο (κόκκινο), γι΄ αυτό και το σκούρο μπλε χρώμα. Το εμπόδιο είναι χρωματισμένο με μωβ χρώμα.

Είναι προφανές ότι ο πράκτορας φτάνει με επιτυχία στο στόχο, χάνοντας μόλις και μετά βίας το εμπόδιο. Σε μια ρομποτική εφαρμογή του πραγματικού κόσμου, κάποιος θα πρόσθετε διάφορους ελέγχους ασφαλείας για να διασφαλίσει την ασφάλεια του συστήματός του. Αυτή η επέκταση μπορεί εύκολα να εφαρμοστεί στο πλαίσιο που προτείνεται στην παρούσα εργασία.

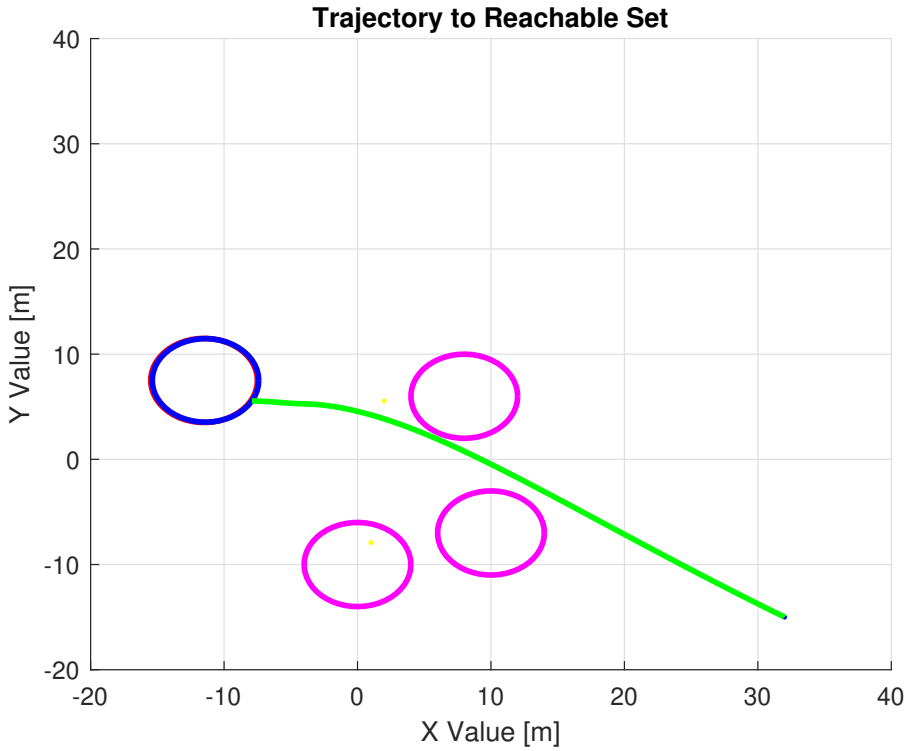


Σχήμα 3.3: Trajectory with One Obstacle. The reachable set (blue) perfectly contains the target (red), thus the dark blue color. Obstacle is colored in magenta.

Παρόμοια αποτελέσματα προκύπτουν και στην περίπτωση του ελλειψοειδούς (*Σχήμα 1.3*), όπου ο πράκτορας φτάνει και πάλι στο προσβάσιμο σύνολο και αποφεύγει το μωβ εμπόδιο.

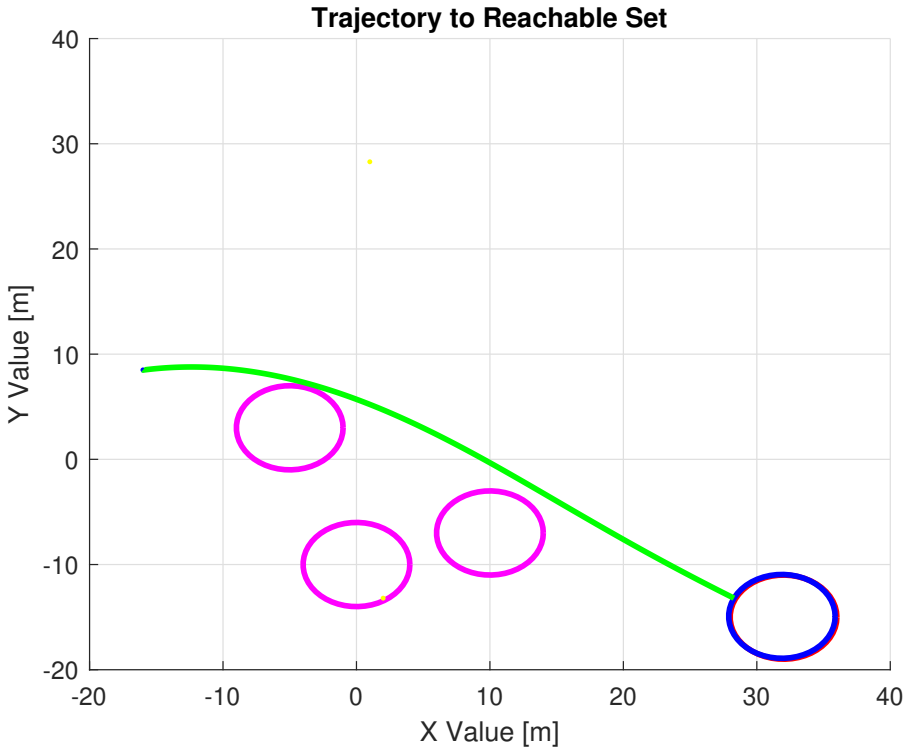
3.0.4 Τρία Εμπόδια

Το σύστημα τριών εμποδίων μπορεί εύκολα να γενικευτεί σε πολυάριθμα εμπόδια. Επομένως, θα δοκιμαστεί τώρα η αποτελεσματικότητα του εν λόγω αλγορίθμου για τα δύο διαφορετικά συστήματα των Σχημάτων 3.4 και 3.5.



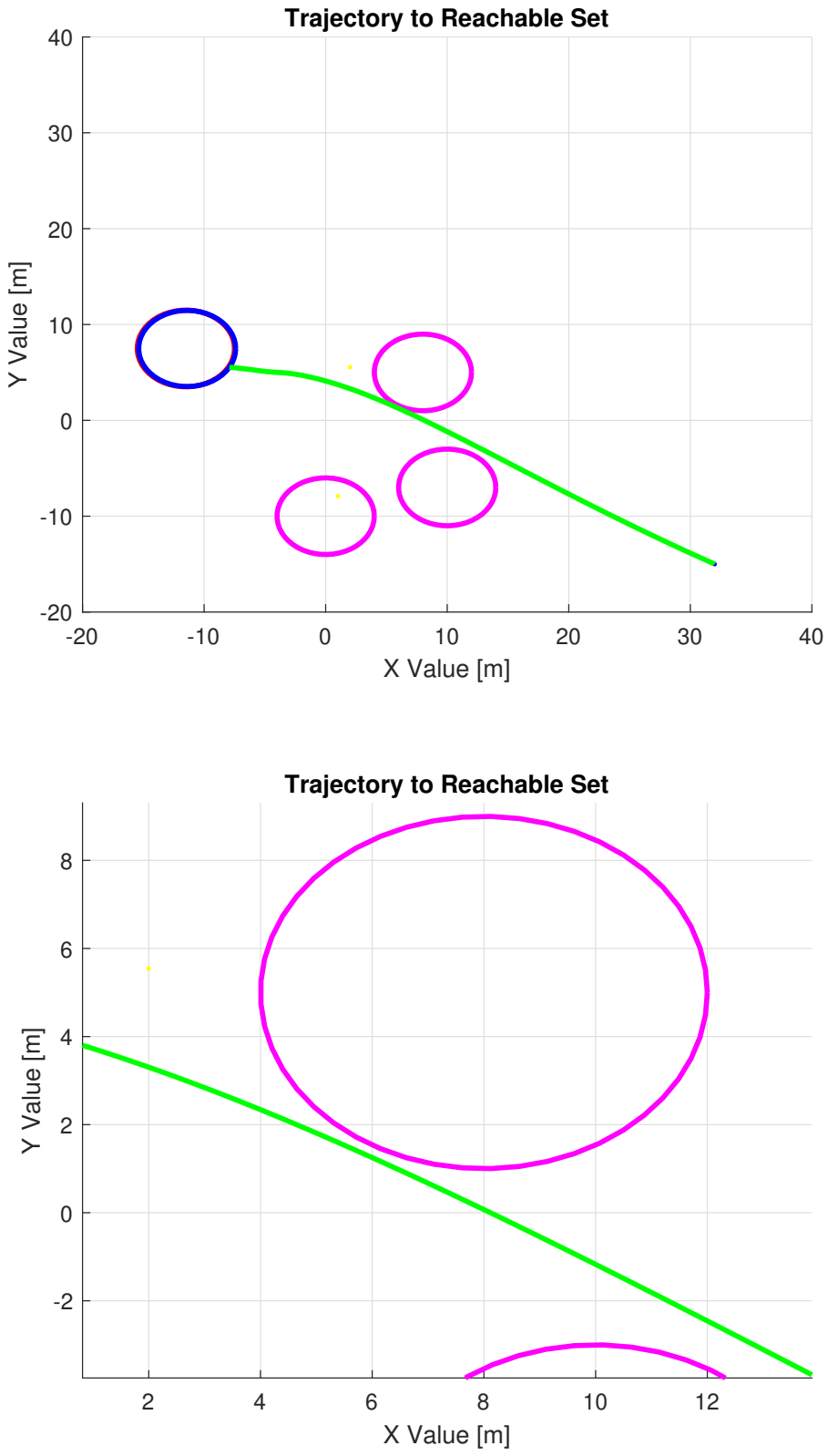
Σχήμα 3.4: Τροχιά με τρία εμπόδια, όλα έχοντας μωβ χρώμα.

Ο αλγόριθμος μπορεί να επιτύχει reachability και στις δύο διατάξεις, παρά την ύπαρξη διαφόρων αντικειμένων. Αξίζει να σημειωθεί εδώ ότι ο πράκτορας προσεγγίζει τα εμπόδια όσο το δυνατόν περισσότερο χωρίς να τα συναντήσει, για να επιτύχει τη μέγιστη δυνατή βελτιστότητα στον σχεδιασμό της τροχιάς. Για άλλη μια φορά, το προσιτό σύνολο περικλείει πλήρως τον στόχο, επιτυγχάνοντας έτσι προσιτότητα σε κάθε σημείο του στόχου.



Σχήμα 3.5: Τροχιά με τρία εμπόδια, όλα χρωματισμένα με μωβ, σε διαφορετική διάταξη.

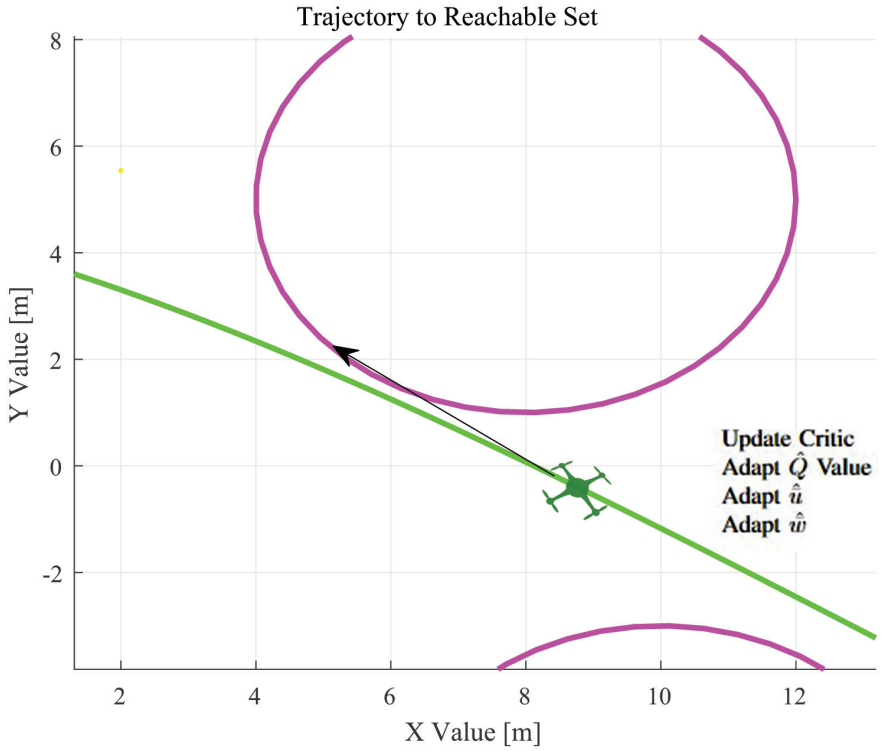
Θέτοντας πιο αυστηρά χωρικά περιθώρια στη διάταξη του Σχήματος 3.4 παράγονται τα αποτελέσματα που παρουσιάζονται στο Σχήμα 3.6.



Σχήμα 3.6: Αυστηρότερο περιβάλλον. Ο αλγόριθμος αποφεύγει κάθε εμπόδιο, κάνοντας ελιγμούς μεταξύ τους.

Το περιβάλλον στα παραπάνω σχήματα είναι αυστηρότερο από αυτό στο Σχήμα 3.4 από άποψη χώρου. Το εμπόδιο με την υψηλότερη τιμή του άξονα y έχει πλέον μετακινηθεί προς τα κάτω κατά μία μονάδα, περιορίζοντας έτσι τον διαθέσιμο χώρο κίνησης. Η τροχιά είναι τώρα πιο οξεία και έχει δύο ακμές, επειδή πρέπει να στριμωχτεί περισσότερο ανάμεσα στα εμπόδια.

Θα άξιζε να τονιστεί εδώ η διαδικασία που θα ακολουθούσε ένα drone-agent για το σύστημα Σχήμα 3.6β. Χωρίς το Q-network, το drone θα ακολουθούσε τυφλά την κατεύθυνση που ορίζει το μαύρο βέλος, με αποτέλεσμα να συγκρουστεί σίγουρα με το αντικείμενο. Ωστόσο, το critic update, η εκτίμηση της τιμής του Q και η προσαρμογή των νόμων ελέγχου ενημέρωσαν επιτυχώς το drone για τον κίνδυνο σε πραγματικό χρόνο, αλλάζοντας έτσι την πορεία του και εξασφαλίζοντας την ασφάλειά του.



Σχήμα 3.7: Μετά το critic update και την προσαρμογή των εκτιμώμενων τιμών, το μη επανδρωμένο αεροσκάφος αποφεύγει με επιτυχία το εμπόδιο. Η τελική τροχιά απεικονίζεται με πράσινο χρώμα.

Αποτελέσματα

Αυτή η ενότητα περιέχει λεπτομέρειες σχετικά με την υπολογιστική πολυπλοκότητα του προτεινόμενου αλγορίθμου. Επιπλέον, περιγράφονται τα όρια του αλγοριθμικού πλαισίου και παρέχονται οδηγίες για την εφαρμογή σε ένα πραγματικό σενάριο.

3.0.5 Υπολογιστική πολυπλοκότητα

Η χρονική πολυπλοκότητα της εκτίμησης της συνάρτησης Q καθορίζεται από τις εξισώσεις (3.25) και (3.41). Συγκεκριμένα, το critic network επεκτείνεται τετραγωνικά με το μέγεθος της επαυξημένης κατάστασης $\hat{U} \in \mathbb{R}^{n+m+p}$, το οποίο μεταφράζεται σε $\mathcal{O}((n+m+p)^2)$. Η εκτίμηση ελέγχου της εισόδου καθορίζεται από τις (3.28) και (3.42), όπου ο χρόνος αποδίδει $\mathcal{O}(nm)$. Ομοίως, η εκτίμηση της διαταραχής καθορίζεται από τις (3.29) και (3.43), όπου ο χρόνος αποδίδει $\mathcal{O}(np)$. Οι διαδικασίες εκτίμησης της προσέγγισης παραγώγων (gradient descent) που περιγράφονται στο [38] είναι ουσιαστικά αυτές στις οποίες στηρίζονται οι υπολογιστικές ανάγκες. Επομένως, η συνολική χρονική πολυπλοκότητα του online framework είναι $\mathcal{O}((n+m+p)^2 + n(m+p)) = \mathcal{O}((n+m+p)^2)$.

3.0.6 Περιορισμοί

Οι περιορισμοί του προτεινόμενου πλαισίου πρέπει να σημειωθούν για να αξιολογηθεί η εφαρμοσιμότητά του σε κάθε σενάριο. Αυτός ο model-free αλγόριθμος μπορεί να χρησιμοποιηθεί σε συνεχούς χρόνου, άγνωστα γραμμικά συστήματα. Αν και δεν αφορά τη μη γραμμική περίπτωση, αν ένα μη γραμμικό σύστημα μπορεί να γραμμικοποιηθεί γύρω από ένα σημείο ισορροπίας και αν το σύστημα είναι ανιχνεύσιμο και ελέγξιμο, η μεθοδολογία ισχύει.

3.0.7 Οδηγίες εφαρμογής

Το προτεινόμενο πλαίσιο δεν απαιτεί πολλές προδιαγραφές για να υλοποιηθεί σωστά. Ο online, model-free έλεγχος του ρομπότ χρειάζεται μια ακριβή γραμμική ανατροφοδότηση κατάστασης, εισόδου και διαταραχών για να τροφοδοτηθεί το σύστημα. Δεδομένου ότι το σύστημα είναι ανιχνεύσιμο, η κατάσταση μπορεί να υπολογιστεί σωστά αν η έξοδος είναι γνωστή.

Συμπεράσματα

Στην παρούσα διπλωματική εργασία προτάθηκε ένας online αλγόριθμος υπολογισμού προσιτών συνόλων χωρίς μοντέλο. Πιο συγκεκριμένα, η βέλτιστη πολιτική ενός συνεχούς ΓΧΑ συστήματος προσεγγίζεται μέσω Q-Learning και υπολογίζεται το reachable set. Στη συνέχεια, εκτελείται μια δειγματική τροχιά του αλγορίθμου. Αποδεικνύεται μαθηματικά ότι η ασυμπτωτική ευστάθεια είναι εγγυημένη για συστήματα άγνωστης/αβέβαιης δυναμικής. Αξιοποιώντας δεδομένα προσομοίωσης, η αποτελεσματικότητα του πλαισίου επικυρώθηκε περαιτέρω, όχι μόνο σε θεωρητικό επίπεδο.

Chapter 4

Introduction

Robotic systems' capabilities have significantly increased in recent years, thanks to quick developments in sensing and processing technology, as well as new optimization methods and data-driven techniques. The quantity and variety of robots functioning autonomously has increased at an unprecedented rate as a result of these advancements, from fully autonomous multi-agent systems of warehouse robots [39] to self-driving cars.

The safety and dependability of these systems depend critically on the decision-making components' proper operation. Although solid operational guarantees are necessary for the deployment of such robotic systems in high-stakes situations, these are the same situations where doing so is usually the most difficult since they frequently entail extremely complex, dynamic, and uncertain environments. One possible solution to this matter is *reachability analysis* [40].

For continuous-state dynamical systems, reachability analysis is a key mathematical technique for the construction and verification of controllers with safety properties. It can determine the appropriate control strategy for a dynamical system whose evolution can be influenced through a control input, driving the system's state toward a desired target configuration while averting undesirable failure states. A reach-avoid problem is more precisely understood as the conjunction of two reachability issues: one in the positive, where the controller aims to reach the target set, and one in the negative, where the controller aims to avoid the failure set. However, it is not enough to solve each problem independently and then integrate the computed solutions, since the optimal choice is coupled between the problems.

More precisely, the controller wants to minimize the outcome of the game, while the disturbance wants to maximize it. The goal is to solve a Hamilton-Jacobi (HJ) PDE, which will efficiently compute the value function [41] and, thus, lead to the reachable set [42], the set of points that the agent can reach while avoiding the obstacles that block its way. One must first establish the optimal cost function (value function) by solving the steady-state Hamilton-Jacobi-Bellman (HJB) problem [43], a nonlinear partial differential equation, to obtain the

optimal control policy. With very few exceptions, this solution cannot be obtained analytically in closed form. Because of this, general Hamilton-Jacobi algorithms eventually rely on state space discretization and are computationally demanding, necessitating computation and memory exponential in the state dimension [44], since they are solved with dynamic programming approaches [45]–[47]. Due to these computation requirements, those algorithms are typically not run online. The most common method is to limit the online section in only performing memory lookups of an offline precomputed controller, like the authors of [48] do.

Reinforcement Learning (RL) [49] techniques have recently been employed to solve such dynamic programming problems, introducing online methods as well [50], [51]. Model free approaches such as Tabular Q-Learning (TQ) and Double Deep Q-Network (DDQN) are widely utilized to approximate reachable sets [30], but often fail to do so in the finite-time horizon. The value function is approximated correctly, especially as the learning rate γ approaches 1, but the computational complexity can be rather heavy. On the other hand, model-based approaches such as [52] are rather niche, thus are not fully including real-world robotic application scenarios. In this work the authors present an online, model-free framework, that is computationally optimized.

Contribution The central theoretical contribution of this paper is the derivation of a novel, reinforcement learning method of constructing reachable sets in finite time. Unlike most popular frameworks that partially run the computations online, this framework can estimate the value function fully online and guarantee safety for the agent to succeed at its task. Furthermore, the proposed scheme relies on the use of only a critic network, allowing the simultaneous learning of the value function and the optimal strategy, thus leading to a less computationally expensive learning architecture. Finally, since the algorithm utilizes Q-learning, a model-free approach is introduced, allowing the user to apply it without having any information of the model's system dynamics.

Structure The remainder of the paper is structured as follows. Section 5 states the finite optimal control problem for general nonlinear dynamical systems that undergo disturbances, whereas Section 6 presents the optimal control two-point boundary value problem (TPBVP). In Section 7, an actor-critic learning framework is developed for learning online and in finite time the solution to the optimal control problem, as well as the reachable set. Section 8 provides illustrative numerical examples, through which the efficacy of our approach is proven. Section 9 touches computational complexity matters, implementation details and limitations of the proposed framework. Finally, Section 10 concludes the paper.

Notation The notation used in this paper is standard. Specifically, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^+$ the set of all positive real numbers, $\mathbb{R}^{a\times b}$ the set of

$a \times b$ real matrices and $\mathbb{N}$ denotes the set of natural numbers. $\lambda_{\max}(A)$ (resp., $\lambda_{\min}(A)$) denotes the maximum (resp., minimum) eigenvalue of the matrix A and $\sigma_{\max}(A)$ (resp., $\sigma_{\min}(A)$) denotes the maximum (resp., minimum) singular value of the matrix A. I_n signifies the $n \times n$ identity matrix, $(\cdot)^T$ and $(\cdot)^{-1}$ denote the transpose and inverse operator, respectively. $|| \cdot ||_i$ is the i-norm of the vector, whereas vech(A), vec(A) and mat(A) are the half-vectorization, the vectorization and the matrization of matrix A, respectively. The Kronecker product of two vectors K, M is denoted as $K \otimes M$, $\oplus$ is the Minkowski sum of two sets and $\wedge$ denotes the logical AND operation. $tr(A)$ is the trace of matrix A and $\in$ is the set operator "in". Lastly, $\bar{K}$, $|K|$ and ∂K denote the closure, the cardinality and the limit points of the set K, respectively, while $a \to b$ denotes that "a approaches b".

Chapter 5

Problem Formulation

Consider a dynamical system of a forward moving agent with a measurable state $x(t) \in X \subseteq \mathbb{R}^n$, control input $u(t) \in U \subseteq \mathbb{R}^m$, and disturbance $w(t) \in W \subseteq \mathbb{R}^p$, which evolves according to the following dynamical equation

$$\dot{x}(t) = Ax(t) + Bu(t) + Cw(t), x(0) = x_0, t \geq 0, \tag{5.1}$$

where $A \in \mathbb{R}^{n \times n}$ is the unknown plant matrix, $B \in \mathbb{R}^{m \times m}$ is the unknown input matrix, and $C \in \mathbb{R}^{p \times p}$ is the unknown disturbance matrix.

Note that the system (5.1) can either depend on the time or not. The method we propose in this paper is independent of time, but one can easily adjust our framework to time-varying systems.

The goal is to compute the Reachable Set G at the end of the time horizon $T \in \mathbb{R}_+$. The reachable set of (5.1) at the final time from a set of initial states $X_0 \subseteq \mathbb{R}^n$ is:

$$\begin{aligned} G(X_0) = \{s \in X \mid x_0 \in X_0, u(\tau) \in U, w(\tau) \in W, \\ \text{s.t. } x(T) = s \text{ and } x(\tau) \notin X_{obs}, \forall \tau \in [0, T]\} \end{aligned} \tag{5.2}$$

where X_{obs} is the set that describes all the obstacles in the environment.

In words, the final reachable set consists of all the states that can be reached at time T, from any initial state in X_0, for any admissible external input or disturbance, given that the agent will not come in contact with an obstacle.

Chapter 6

Finite Horizon Optimal Control Method

We denote the difference between the current state $x(t)$ and the final state $x(T)$ as our new state $\bar{x}(t) := x(t) - x(T)$. Defining the final state as $x_f := x(T)$, we can write $\bar{x}(t) := x(t) - x_f$. Consequently, the control and the disturbance become $\bar{u}(t) := u(t) - u_f$ and $\bar{w}(t) := w(t) - w_f$, respectively, considering $u_f := u(T)$ and $w_f := w(T)$. The updated dynamical system becomes

$$\begin{aligned}\dot{\bar{x}}(t) =& \dot{x}(t) - \dot{x_f} \\ =& A\bar{x}(t) + B\bar{u}(t) + C\bar{w}(t), \bar{x}(0) = x_0 - x_f, t \geq 0.\end{aligned} \tag{6.1}$$

Our goal is to obtain the optimal value function

$$V^\star(\bar{x}; t_0, T) := \max_{\bar{w}} \min_{\bar{u}} J(\bar{x}; \bar{u}; \bar{w}; t_0, T) \tag{6.2}$$

for the unknown system. $J(\bar{x}; \bar{u}; \bar{w}; t_0, T)$ is the cost functional defined as

$$\begin{aligned}J(\bar{x}; \bar{u}; \bar{w}; t_0, T) = \Phi(T) \\ + \frac{1}{2} \int_{t_0}^{T} (\bar{x}^{\mathrm{T}} M \bar{x} + \bar{u}^{\mathrm{T}} R \bar{u} - \bar{w}^{\mathrm{T}} F \bar{w}) d\tau\end{aligned} \tag{6.3}$$

where $\Phi(T) := (1/2)\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T)$ is the terminal cost with $P(T) := P_T \in \mathbb{R}^{n \times n} > 0$ being the last Ricatti matrix and $M \in \mathbb{R}^{n \times n} \geq 0$, $R \in \mathbb{R}^{m \times m} > 0$ and $F \in \mathbb{R}^{p \times p} > 0$ being the matrices that penalize the state, control input and disturbance, respectively, all defined by the controller. We will assume here that the system is detectable and controllable.

We are essentially interested in finding an optimal control $\bar{u}^\star$ and optimal disturbance $\bar{w}^\star$ such that $J(\bar{x}; \bar{u}^\star; t_0, T) \leq J(\bar{x}; \bar{u}; t_0, T), \forall \bar{x}, \bar{u}$ and $J(\bar{x}; \bar{w}^\star; t_0, T) \geq J(\bar{x}; \bar{w}; t_0, T), \forall \bar{x}, \bar{w}$, respectively, which can be described as the differential game shown in (6.2). The result of this game will produce the *Reachable Set* at the end

of the time horizon due to the boundary condition [31]:

$$V^{\star}(\bar{x}(T);T) = \bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T), \forall \bar{x}(T) \in G \tag{6.4}$$

where G will be the reachable set. Such a problem has a solution only if $x_f \approxeq 0$ [32]. In this paper, we will propose a model-free approach for finding the reachable set G.

In order to solve the finite horizon optimal control problem described by (6.2), we derive the following Hamilton-Jacobi-Bellman (HJB) equation:

$$\begin{aligned} -\frac{\partial V^{\star}}{\partial t} = &\frac{1}{2}(\bar{x}^{\mathrm{T}}M\bar{x} + \bar{u}^{*\mathrm{T}}R\bar{u}^{\star} - \bar{w}^{*\mathrm{T}}F\bar{w}^{\star}) \\ &+ \lambda^{\mathrm{T}}(A\bar{x} + B\bar{u}^{\star} + C\bar{w}^{\star}), \forall \bar{x} \end{aligned} \tag{6.5}$$

since the Hamiltonian $\mathcal{H}$ with regards to (6.1),(6.2) and (6.3) is defined as

$$\begin{aligned} \mathcal{H}(\bar{x};\bar{u};\bar{w};\lambda) = &\frac{1}{2}(\bar{x}^{\mathrm{T}}M\bar{x} + \bar{u}^{\mathrm{T}}R\bar{u} - \bar{w}^{\mathrm{T}}F\bar{w}) \\ &+ \lambda^{\mathrm{T}}(A\bar{x} + B\bar{u} + C\bar{w}), \forall \bar{x}, \bar{u}, \bar{w}, \lambda \end{aligned} \tag{6.6}$$

By making use of the sweep [33] method and $\lambda = \frac{\partial V^{\star}}{\partial x}$, equation (6.5) becomes

$$\begin{aligned} -\frac{\partial V^{\star}}{\partial t} = &\frac{1}{2}(\bar{x}^{\mathrm{T}}M\bar{x} + \bar{u}^{*\mathrm{T}}R\bar{u}^{\star} - \bar{w}^{*\mathrm{T}}F\bar{w}^{\star}) \\ &+ \frac{\partial V^{\star\,\mathrm{T}}}{\partial x}(A\bar{x} + B\bar{u}^{\star} + C\bar{w}^{\star}), \forall \bar{x} \end{aligned} \tag{6.7}$$

Taking into consideration that our system (5.1) is linear and the disturbance is maximizing the LQR cost functional (6.3), the value function will have a quadratic form in the state $\bar{x}$, shown below.

$$V^{\star}(\bar{x};t) = \frac{1}{2}\bar{x}^{\mathrm{T}}P(t)\bar{x}, \forall \bar{x}, t. \tag{6.8}$$

where $P(t) \in \mathbb{R}^{n \times n}$ is the solution [34] to the Riccati equation

$$\begin{aligned} -\dot{P}(t) = &M + P(t)A + A^{\mathrm{T}}P(t) \\ &+ P(t)(CF^{-1}C^{\mathrm{T}} - BR^{-1}B^{\mathrm{T}})P(t) \end{aligned} \tag{6.9}$$

Thus, the optimal control and disturbance are, respectively:

$$\bar{u}^{\star}(\bar{x};t) = -R^{-1}B^{\mathrm{T}}P(t)\bar{x}, \forall \bar{x}, t. \tag{6.10}$$

$$\bar{w}^{\star}(\bar{x};t) = F^{-1}C^{\mathrm{T}}P(t)\bar{x}, \forall \bar{x}, t. \tag{6.11}$$

If one wanted to handle the solution method of system (5.1) as a differential

game, they would have to rewrite the equations above as

$$\bar{u}^{\star}(\bar{x};t) = \arg\min_{u\in U}\max_{w\in W}\mathcal{H}(\bar{x};\bar{u};\bar{w};\lambda) \tag{6.12}$$

$$\bar{w}^{\star}(\bar{x};t) = \arg\max_{w\in W}\min_{u\in U}\mathcal{H}(\bar{x};\bar{u};\bar{w};\lambda) \tag{6.13}$$

Chapter 7

Model-free Actor-Critic Method

Let us define the following function $\mathcal{Q} : \mathbb{R}^{(n+m+p)} \times \mathbb{R}^{(n+m+p)\times(n+m+p)} \rightarrow \mathbb{R}^{+}$:

$$\begin{aligned}\mathcal{Q}(\bar{x};\bar{u};\bar{w};t) &:= V^{\star}(\bar{x};t) + \mathcal{H}(\bar{x};\bar{u};\bar{w};\frac{\partial V^{\star}}{\partial t},\frac{\partial V^{\star}}{\partial x}) \\ &= V^{\star}(\bar{x};t) + \frac{1}{2}\bar{x}^{\mathrm{T}}M\bar{x} + \frac{1}{2}\bar{u}^{\mathrm{T}}R\bar{u} \\ &- \frac{1}{2}\bar{w}^{\mathrm{T}}F\bar{w} + \bar{x}^{\mathrm{T}}P(t)(A\bar{x} + B\bar{u} + C\bar{w}) \\ &+ \frac{1}{2}\bar{x}^{\mathrm{T}}\dot{P}(t)\bar{x}, \forall \bar{x},\bar{u},\bar{w},t. \end{aligned} \tag{7.1}$$

We then define the augmented state $U := [\bar{x}^{\mathrm{T}}\bar{u}^{\mathrm{T}}\bar{w}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{(n+m+p)}$ to rewrite the action-dependent advantage function in a compact form [35]. More specifically,

$$\mathcal{Q}(\bar{x};\bar{u};\bar{w};t) = \frac{1}{2}U^{\mathrm{T}}\bar{\mathcal{Q}}(t)U \tag{7.2}$$

where

$$\bar{\mathcal{Q}}(t) = \begin{bmatrix} Q_{\mathrm{xx}}(t) & Q_{\mathrm{xu}}(t) & Q_{\mathrm{xw}}(t) \\ Q_{\mathrm{ux}}(t) & Q_{\mathrm{uu}}(t) & Q_{\mathrm{uw}}(t) \\ Q_{\mathrm{wx}}(t) & Q_{\mathrm{wu}}(t) & Q_{\mathrm{ww}}(t) \end{bmatrix} \tag{7.3}$$

Making use of the Riccati matrix's symmetrical properties, $\bar{x}^{\mathrm{T}}P(t)A\bar{x} = (1/2)\bar{x}^{\mathrm{T}}(P(t)A + A^{\mathrm{T}}P(t))\bar{x}$, and for the controls $\bar{x}^{\mathrm{T}}P(t)B\bar{u} = (1/2)\bar{x}^{\mathrm{T}}(P(t)B + B^{\mathrm{T}}P(t))\bar{u}$ and $\bar{x}^{\mathrm{T}}P(t)C\bar{w} = (1/2)\bar{x}^{\mathrm{T}}(P(t)C + C^{\mathrm{T}}P(t))\bar{w}$. Thus, each individual Q_i is as follows:

$Q_{\mathrm{xx}}(t) = \dot{P}(t) + P(t) + M + P(t)A + A^{\mathrm{T}}P(t) + P(t)B + P(t)C$, $Q_{\mathrm{xu}}(t) = Q_{\mathrm{ux}}(t)^{\mathrm{T}} = P(t)B$, $Q_{\mathrm{xw}}(t) = Q_{\mathrm{wx}}(t)^{\mathrm{T}} = P(t)C$, $Q_{\mathrm{uu}}(t) = R$, $Q_{\mathrm{uw}}(t) = Q_{\mathrm{wu}}(t)^{\mathrm{T}} = 0$, $Q_{\mathrm{uu}}(t) = F$.

$u^{\star}$ and $w^{\star}$ of equations (6.10) and (6.11) can be formulated in a model-free way by utilizing the stationary conditions $\partial\mathcal{Q}(\bar{x};\bar{u};\bar{w};t)/\partial\bar{u} = 0$ and $\partial\mathcal{Q}(\bar{x};\bar{u};\bar{w};t)/\partial\bar{w} = 0$. We obtain

$$\bar{u}^\star(\bar{x};t) = \arg\min_{\bar{u}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) = -Q_{\mathrm{uu}}^{-1}Q_{\mathrm{ux}}(t)\bar{x}. \tag{7.4}$$

$$\bar{w}^\star(\bar{x};t) = \arg\max_{\bar{w}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) = Q_{\mathrm{ww}}^{-1}Q_{\mathrm{wx}}(t)\bar{x}. \tag{7.5}$$

In this manuscript, a differential game approach will be proposed as the solution of system (5.1), therefore the equations above now become

$$\bar{u}^\star(\bar{x};t) = \arg\min_{\bar{u}}\max_{\bar{w}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) \tag{7.6}$$

$$\bar{w}^\star(\bar{x};t) = \arg\max_{\bar{w}}\min_{\bar{u}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t) \tag{7.7}$$

Lemma 1: The value of the game $\mathcal{Q}^\star(\bar{x};\bar{u}^\star;\bar{w}^\star;t) = \max_{\bar{w}}\min_{\bar{u}} \mathcal{Q}(\bar{x};\bar{u};\bar{w};t)$ is the same with the optimal value $V^\star$ in (6.8) of the differential game (6.2), where $P(t)$ is the Riccati matrix found from solving equation (6.9).

Proof. : Substitute equations (6.12) and (6.13) in the $\mathcal{Q}$ function (7.1) to obtain (6.9). Therefore, $\mathcal{Q}(\bar{x};\bar{u}^\star;\bar{w}^\star;t) = V^\star(\bar{x};t)$ □

7.0.1 Actor-Critic Architecture

By utilizing the symmetric properties of the $\bar{\mathcal{Q}}$ matrix, we can successfully calculate the Q function in (7.1) as

$$\mathcal{Q}^\star(\bar{x};\bar{u}^\star;\bar{w}^\star;t) = \frac{1}{2}U^{\mathrm{T}}\bar{\mathcal{Q}}(t)U = \frac{1}{2}vech(\bar{\mathcal{Q}}(t))^{\mathrm{T}}(U\otimes U) \tag{7.8}$$

where $vech(\bar{\mathcal{Q}}(t)) \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)}$ is the half-vectorization operation, which significantly reduces the computational complexity. We now define the term $v(t)^{\mathrm{T}}W_c := \frac{1}{2}vech(\bar{\mathcal{Q}}(t))$, that approximates the Q-function as

$$\mathcal{Q}^\star(\bar{x};\bar{u}^\star;\bar{w}^\star;t) = W_c^{\mathrm{T}}v(t)(U\otimes U) \tag{7.9}$$

with $W_c \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)}$ being the critic weight estimates and $v(t) \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)\times((n+m+p)(n+m+p+1)/2)}$ a bounded radial basis function, solely dependant on time.

Our goal is to find the ideal weight estimates, thus we apply an adaptive estimation technique that utilizes current weights, as shown in [36]. Consequently, we have

$$\hat{\mathcal{Q}}(\bar{x};\bar{u};\bar{w};t) = \hat{W}_c{}^{\mathrm{T}}v(t)(U\otimes U) \tag{7.10}$$

where $\hat{W}_c{}^{\mathrm{T}}v(t) := \frac{1}{2}vech(\bar{\hat{Q}}(t))$.

As for the actor structure, we create two actor instances, one for the control and one for the disturbance. Similar to the critic, we assign $\mu(t)^{\mathrm{T}}W_{a_1} := -Q_{\mathrm{uu}}^{-1}Q_{\mathrm{ux}}(t)\bar{x}$

and $\mu(t)^{\mathrm{T}} W_{a_2} := Q_{\mathbf{ww}}^{-1} Q_{\mathbf{wx}}(t)\bar{x}$ to write

$$\bar{u}^{\star}(\bar{x};t) = W_{a_1}^{\mathrm{T}}\mu(t)\bar{x}. \tag{7.11}$$

$$\bar{w}^{\star}(\bar{x};t) = W_{a_2}^{\mathrm{T}}\mu(t)\bar{x}. \tag{7.12}$$

where $W_{a_1} \in \mathbb{R}^{n\times m}$ are the controller actor's weight estimates, $W_{a_2} \in \mathbb{R}^{n\times p}$ the disturbance actor's weight estimates and $\mu(t) \in \mathbb{R}^{n\times n}$ is another bounded radial time-only dependent function. The actors then become

$$\hat{\bar{u}}(\bar{x};t) = \hat{W_{a_1}^{\mathrm{T}}}\mu(t)\bar{x}. \tag{7.13}$$

$$\hat{\bar{w}}(\bar{x};t) = \hat{W_{a_2}^{\mathrm{T}}}\mu(t)\bar{x}. \tag{7.14}$$

We must note here that this structure is for the whole space and not just a compact set. The reason for that is that the actor-critic approximators described in the equations (7.10) - (7.14) are not characterized by approximation errors, thus optimal policies are achieved.

The Bellman equation is represented using the integral reinforcement learning approach from [37] as

$$\begin{aligned} V^{\star}(\bar{x}(t);t) &= V^{\star}(\bar{x}(t-\Delta t);t-\Delta t) \\ &\quad -\frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^{\mathrm{T}}M\bar{x} + \hat{\bar{u}}^{\mathrm{T}}R\hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}}F\hat{\bar{w}})d\tau \end{aligned} \tag{7.15}$$

$$V^{\star}(\bar{x}(T);T) = \frac{1}{2}\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T), \tag{7.16}$$

with $\Delta t \in \mathbb{R}^{+}$ being a tiny standard value.

Using the Lemma above, where we have demonstrated that $\mathcal{Q}^{\star}(\bar{x};\hat{\bar{u}}^{\star};\hat{\bar{w}}^{\star};t) = V^{\star}(\bar{x};t)$, we can rewrite the equations above as

$$\begin{aligned} Q^{\star}(\bar{x}(t);\hat{\bar{u}}^{\star}(t);\hat{\bar{w}}^{\star}(t);t) &= Q^{\star}(\bar{x}(t-\Delta t);\hat{\bar{u}}^{\star}(t-\Delta t); \\ &\qquad \hat{\bar{w}}^{\star}(t-\Delta t);t-\Delta t) \\ &\quad -\frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^{\mathrm{T}}M\bar{x} + \hat{\bar{u}}^{\mathrm{T}}R\hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}}F\hat{\bar{w}})d\tau \end{aligned} \tag{7.17}$$

$$Q^{\star}(\bar{x}(T);\hat{\bar{u}}^{\star}(T);\hat{\bar{w}}^{\star}(T);T) = \frac{1}{2}\bar{x}^{\mathrm{T}}(T)P(T)\bar{x}(T) \tag{7.18}$$

From (6.4) and from the Lemma above, it is obvious that the reachable set G can be immediately acquired from $Q^{\star}(\bar{x}(T);\hat{\bar{u}}^{\star}(T);\hat{\bar{w}}^{\star}(T);T)$.

We then choose the errors e_{c1} and e_{c2}, which we want to reduce to zero by properly adjusting the critic weights of (7.10). Describe the initial critic error,

$e_{c1} \in \mathbb{R}$ as

$$
\begin{aligned}
e_{c1} &:= \hat{\mathcal{Q}}(\bar{x}; \hat{\bar{u}}; \hat{\bar{w}}; t) \\
&\quad - \hat{Q}(\bar{x}(t-\Delta t); \hat{\bar{u}}(t-\Delta t); \hat{\bar{w}}(t-\Delta t); t-\Delta t) \\
&\quad + \frac{1}{2}\int_{t-\Delta t}^{t} (\bar{x}^{\mathrm{T}} M \bar{x} + \hat{\bar{u}}^{\mathrm{T}} R \hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}} F \hat{\bar{w}}) d\tau \\
&\quad = \hat{W}_c^{\mathrm{T}} v(t)((\hat{U}(t) \otimes \hat{U}(t)) - (\hat{U}(t-\Delta t) \otimes \hat{U}(t-\Delta t)) \\
&\quad + \frac{1}{2}\int_{t-\Delta t}^{t} (\bar{x}^{\mathrm{T}} M \bar{x} + \hat{\bar{u}}^{\mathrm{T}} R \hat{\bar{u}} - \hat{\bar{w}}^{\mathrm{T}} F \hat{\bar{w}}) d\tau
\end{aligned} \tag{7.19}
$$

where $\hat{U} := [\bar{x}^{\mathrm{T}} \hat{\bar{u}}^{\mathrm{T}} \hat{\bar{w}}^{\mathrm{T}}]^{\mathrm{T}}$ is the augmented state that consists of the measurable full state vector and the estimated control and disturbance. As for the second critic error, this point on referred to as *final critic error*, it is defined as the real value

$$
e_{cf} := \frac{1}{2}\bar{x}^{\mathrm{T}}(T) P(T) \bar{x}(T) - \hat{W}_c^{\mathrm{T}} v(T)((\hat{U}(T) \otimes \hat{U}(T)) \tag{7.20}
$$

Next, we declare the actor approximation errors for our two actors, the control, and the disturbance, as $e_{a1} \in \mathbb{R}^m, e_{a2} \in \mathbb{R}^p$ respectively. More specifically,

$$
e_{a1} := \hat{W}_{a_1}^{\mathrm{T}} \mu(t) \bar{x} + \hat{Q}_{\mathrm{uu}}^{-1} \hat{Q}_{\mathrm{ux}}(t) \bar{x} \tag{7.21}
$$

$$
e_{a2} := \hat{W}_{a_2}^{\mathrm{T}} \mu(t) \bar{x} - \hat{Q}_{\mathrm{ww}}^{-1} \hat{Q}_{\mathrm{wx}}(t) \bar{x} \tag{7.22}
$$

with $\hat{Q}_{\mathrm{uu}}$, $\hat{Q}_{\mathrm{ux}}$, $\hat{Q}_{\mathrm{ww}}$, $\hat{Q}_{\mathrm{wx}}$ being derived from the estimation of the critic $\hat{W}_c$.

The squared norm of errors, following the adaptive control method of [36], can be obtained as

$$
N_1(\hat{W}_c, \hat{W}_c(T)) = \frac{1}{2}||e_{c1}||^2 + \frac{1}{2}||e_{cf}||^2 \tag{7.23}
$$

$$
N_2(\hat{W}_{a_1}) = \frac{1}{2}||e_{a1}||^2 \tag{7.24}
$$

$$
N_3(\hat{W}_{a_2}) = \frac{1}{2}||e_{a2}||^2 \tag{7.25}
$$

Thus, the learning framework can be implemented by employing a gradient

descent algorithm in (7.23), in the following normalized manner

$$\begin{aligned}\dot{\hat{W}}_c &= -\alpha_c \frac{\partial N_1}{\partial \hat{W}_c} \\ &= -\alpha_c \left(\frac{\sigma e_{c1}}{(1+\sigma^T\sigma)^2} + \frac{\sigma_f e_{c2}}{(1+\sigma_f^T\sigma_f)^2} \right)\end{aligned} \tag{7.26}$$

with $\alpha_c \in \mathbb{R}^+$ being the convergence rate that the designer specifies for the gradient descent and $\sigma(t) := v(t)(\hat{U}(t) \otimes \hat{U}(t) - (\hat{U}(t-\Delta t) \otimes \hat{U}(t-\Delta t))$, $\sigma(T) := v(T)(\hat{U}(T) \otimes \hat{U}(T)$. As e_{c1} and e_{c2} approach 0, the critic update (7.26) ensures that $\hat{W}_c \to W_c$ and $\hat{W}_c(T) \to W_c(T)$.

Similarly, by applying a normalized gradient descent approach in (7.24) - (7.25) the following actor tuning results are yielded

$$\dot{\hat{W}}_{a_1} = -\alpha_a \frac{\partial N_2}{\partial \hat{W}_{a_1}} = -\alpha_a \bar{x} e_{a1}^T \tag{7.27}$$

$$\dot{\hat{W}}_{a_2} = -\alpha_a \frac{\partial N_3}{\partial \hat{W}_{a_2}} = -\alpha_a \bar{x} e_{a2}^T \tag{7.28}$$

with $\alpha_a \in \mathbb{R}^+$ being another convergence rate that the designer specifies for the gradient descent. The actor tuning (7.27) and (7.28) guarantee that as $e_{a1}, e_{a2} \to 0$, then $\hat{W}_{a_1} \to W_{a_1}$ and $\hat{W}_{a_2} \to W_{a_2}$, respectively.

We now define the critic's estimation error as $\tilde{W}_c \in \mathbb{R}^{((n+m+p)(n+m+p+1)/2)}$ as $\tilde{W}_c := W_c - \hat{W}_c$. Similarly, the two actors' estimation errors $\tilde{W}_{a_1} \in \mathbb{R}^{n\times m}$, $\tilde{W}_{a_2} \in \mathbb{R}^{n\times p}$ are defined as $\tilde{W}_{a_1} := W_{a_1} - \hat{W}_{a_1}$ and $\tilde{W}_{a_2} := W_{a_2} - \hat{W}_{a_2}$, respectively.

The dynamics of the critic's estimation error are

$$\begin{aligned}\dot{\tilde{W}}_c &= \left(\frac{\partial W_c}{\partial e_{c1}} \frac{de_{c1}}{\mathrm{d}t} + \frac{\partial W_c}{\partial e_{cf}} \frac{de_{cf}}{\mathrm{d}t} \right) - \left(\frac{\partial \hat{W}_c}{\partial e_{c1}} \frac{de_{c1}}{\mathrm{d}t} + \frac{\partial \hat{W}_c}{\partial e_{cf}} \frac{de_{cf}}{\mathrm{d}t} \right) \\ &= \frac{\partial W_c}{\partial e_{c1}} \frac{de_{c1}}{\mathrm{d}t} - \frac{\partial \hat{W}_c}{\partial e_{c1}} \frac{de_{c1}}{\mathrm{d}t} \\ &= -\alpha_c \frac{\sigma\sigma^T \tilde{W}_c}{(1+\sigma^T\sigma)^2}\end{aligned} \tag{7.29}$$

and the respective dynamics of the control and disturbance actors are

$$\dot{\tilde{W}}_{a_1} = -\alpha_a \bar{x}\bar{x}^T \mu(t) \tilde{W}_{a_1} - \alpha_a \bar{x}\bar{x}^T \frac{\mu(t) \tilde{Q}_{\mathrm{xu}} R^{-1}}{||1+\mu(t)^T\mu(t)||^2} \tag{7.30}$$

$$\dot{\tilde{W}}_{a_2} = -\alpha_a \bar{x}\bar{x}^T \mu(t) \tilde{W}_{a_2} - \alpha_a \bar{x}\bar{x}^T \frac{\mu(t) \tilde{Q}_{\mathrm{xw}} F^{-1}}{||1+\mu(t)^T\mu(t)||^2} \tag{7.31}$$

where $\tilde{Q}_{\mathrm{xu}} := \mathrm{mat}(\tilde{W}_c[n(n+1)/2+1 : (n(n+1)/2+nm])$ and $\tilde{Q}_{\mathrm{xw}} := \mathrm{mat}(\tilde{W}_c[n(n+1)/2+1 : (n(n+1)/2+np])$.

The aforementioned methodology can be summarized in the algorithm below. $rand(x)$ is the random function of x digits in vector length. `pointsOfTheTarget` are the points that consist of the target, `targetCoords(x)` is the vector of the x and y coordinates of the target for x. `Critic` is the function that updates the critic parameters in (7.26), while `Actor1` is the function that estimates the first actor's parameters in (7.27) and `Actor2` the one that estimates the second actor's ones, in (7.28). The `AdaptControl` and `AdaptDisturbance` functions estimate the control and disturbance, respectively, according to the equations (7.13), (7.14), respectively. `AdaptQ` estimates $\hat{Q}$'s parameters from (7.10), whereas `Augment` is just used to augment the current state during the solving of the differential equation. Lastly, `trajectory's last` is the last pair of x-y coordinates in each $trajectory$.

Reachable Set Computation Algorithm

$G \leftarrow \emptyset$;
$W_c(0) \leftarrow rand(10)$;
$W_{a_1}(0) \leftarrow rand(2)$;
$W_{a_2}(0) \leftarrow rand(2)$;
$points \leftarrow$ `pointsOfTheTarget`;
$allTrajectories \leftarrow$ [ ];
for $i = 1$ `to` $points$ **do**
    $x_f \leftarrow$ `targetCoords(`i`)`;
    for $t \in T$ **do**
        $\hat{W}_c \leftarrow$ `Critic`$(\hat{x}, \hat{u}, \hat{w}, a_c, M, R, F, \Delta t, P(T))$;
        $\hat{Q} \leftarrow$ `AdaptQ`$(\hat{x}, \hat{u}, \hat{w}, \hat{W}_c)$;
        $\hat{W}_{a_1} \leftarrow$ `Actor1`$(\hat{x}, \hat{u}, \hat{Q}, a_a)$;
        $\hat{W}_{a_2} \leftarrow$ `Actor2`$(\hat{x}, \hat{w}, \hat{Q}, a_a)$;
        $\hat{\hat{u}} \leftarrow$ `AdaptControl`$(\hat{x}, \hat{u}, \hat{W}_{a_1})$;
        $\hat{\hat{w}} \leftarrow$ `AdaptDisturbance`$(\hat{x}, \hat{w}, \hat{W}_{a_2})$;
        $\dot{x} \leftarrow$ `Augment` $(\hat{W}_c, \hat{W}_{a_1}, \hat{W}_{a_2})$;
        `Return` $\hat{\hat{u}}, \hat{\hat{w}}$;
    end for
    $trajectory \leftarrow \dot{x}(t)$;
    $allTrajectories \leftarrow [allTrajectories, trajectory]$;
    $G \leftarrow G \bigcup$ {`trajectory's last`};
end for
Return G;
Calculate minimum distance from x_0 to target using Euclidean Norm;
Plot Trajectory from x_0 to the closest point of the target using $allTrajectories$;

Chapter 8

Simulations

The simulations until the end of this chapter can be found in my GitHub repository[1].

In this section, we demonstrate the efficiency of the proposed framework through simulations. We study simple point-target systems, in which the point-agent is equipped with our online, model-free algorithm that aims to approximate the optimal policy. Next, the trajectory to the target is calculated, while avoiding any obstacles that may exist. In the sections below, one can witness the performance of the algorithm for systems of 0, 1 and 3 obstacles, respectively.

All simulations are characterized by a finite time horizon *T* of 6 sec, an internal dynamics step $\Delta t = 0.05$ sec, as well as convergence rates a_c, a_a of 90 and 1.5 respectively. As for the designer-defined matrices, the matrix that penalizes the state is $M = I_2$, the one that penalizes the control is $R = 0.1$ and the one that penalizes the disturbance is $F = 0.1$. The final Riccati matrix is $P(T) = 0.5I_3$, while the final control and disturbance actions $u(T), w(T)$, respectively, are both 0.005. The initial values of $\hat{W}_c, \hat{W}_{a_1}, \hat{W}_{a_2}$ are randomly selected, except from $\hat{W}_c$'s two elements that correspond to $Q_{\text{uu}}, Q_{\text{ww}}$, which need to differ from zero, as they are inverted in (7.4) - (7.5).

8.0.1 Obstacle Absence

Consider a simple point-target system like the one shown in *Fig.1*. The agent, depicted as the blue dot, successfully and almost optimally reaches its target, the dark blue circle. The target is depicted in red, but the calculated reachable (blue) perfectly contains it, thus this dark blue color is created.

[1] https://github.com/Costopoulos/helperOC/tree/ReachabilityGame

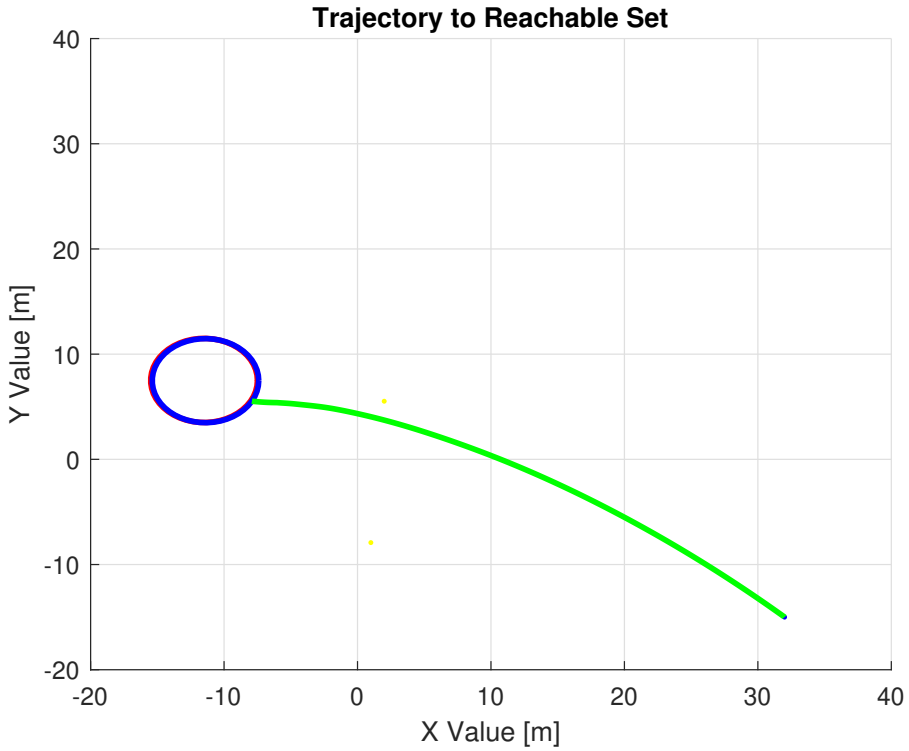


Figure 8.1: Trajectory to the Reachable Set. The reachable set (blue) perfectly contains the target (red), thus the dark blue color.

8.0.2 Single Obstacle

Let us now consider the same setup, but with an obstacle intercepting the path. In *Fig.2* both the target and the obstacle are square-like and in *Fig.3* both are circular, exactly as in figure 1.

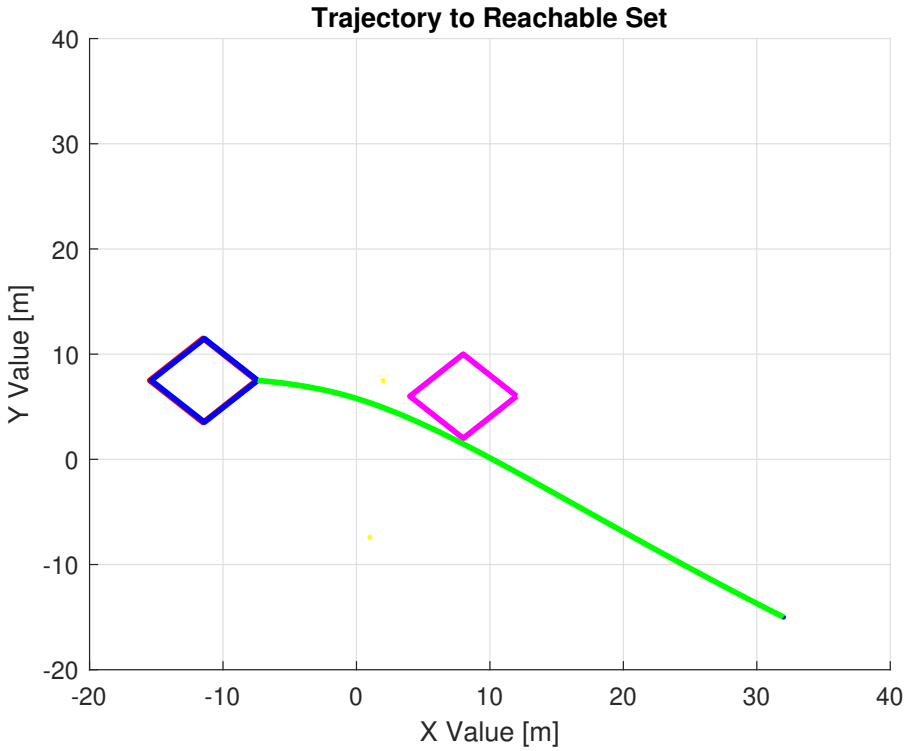


Figure 8.2: Trajectory with One Obstacle in a Rhombus-like Scenario. The reachable set (blue) perfectly contains the target (red), thus the dark blue color. Obstacle is colored in magenta.

It is clearly visible that the agent successfully reaches the target, just barely missing the obstacle. In a real-world robotic application, one would add various security checks to ensure the safety of their system. This extension can be easily applied in the framework proposed in this work.

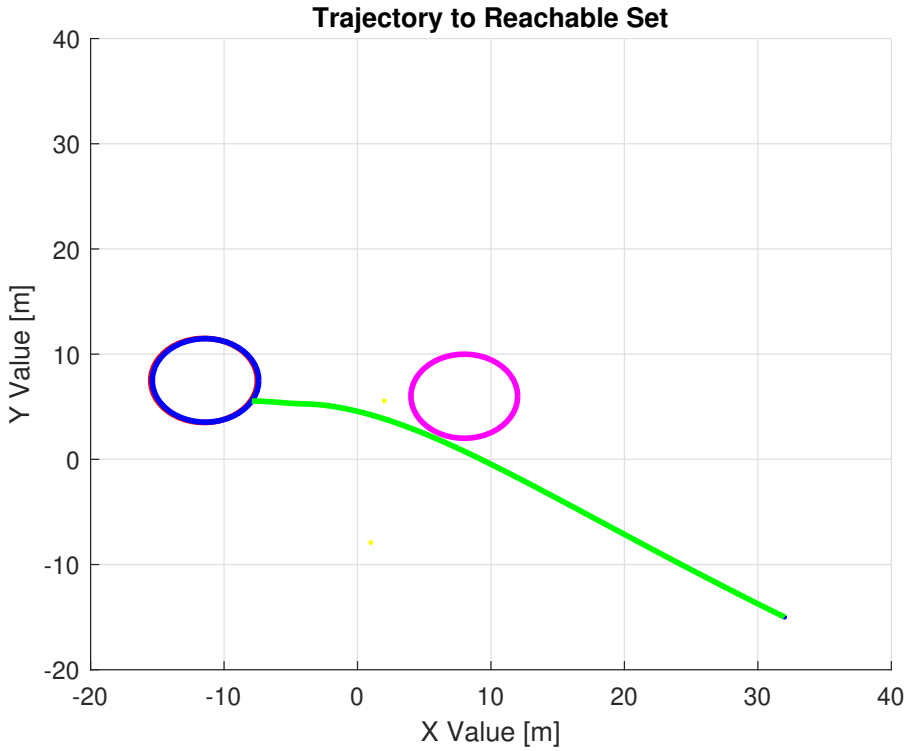


Figure 8.3: Trajectory with One Obstacle. The reachable set (blue) perfectly contains the target (red), thus the dark blue color. Obstacle is colored in magenta.

Similar results are acquired in the ellipsoid case (*Fig.3*), where the agent once more does reach the reachable set and avoids the magenta obstacle.

8.0.3 Three Obstacles

The setting of three obstacles can easily be generalized to numerous ones. Therefore, we will now test the efficacy of our algorithm for the two different settings of *Fig.* 8.4 and 8.5.

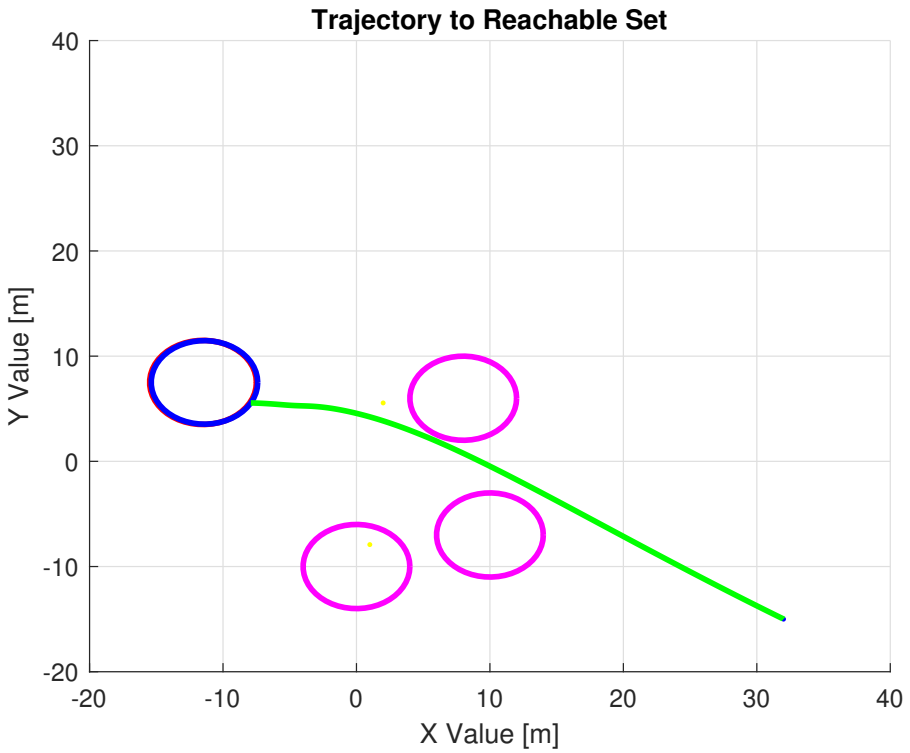


Figure 8.4: Trajectory with Three Obstacles, all colored in magenta.

The algorithm can achieve reachability in both set-ups, despite the existence of various objects. It is worth noting here that the agent approaches the obstacles as much as possible without encountering them, to achieve maximum optimality in the trajectory planning. Once again, the reachable set totally encloses the target, thus achieving reachability in every point of the target.

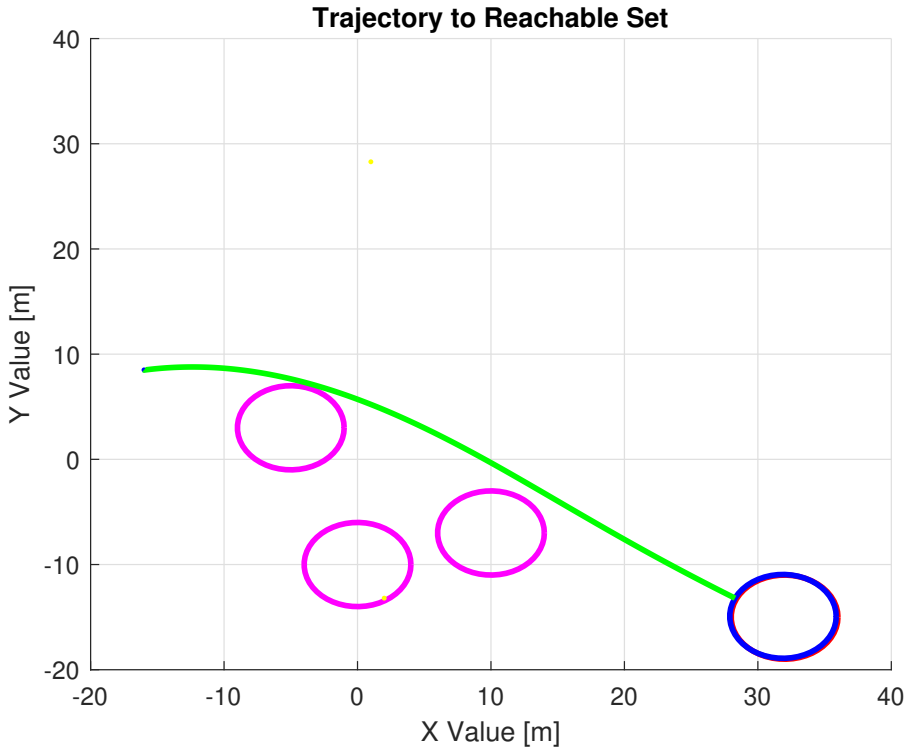
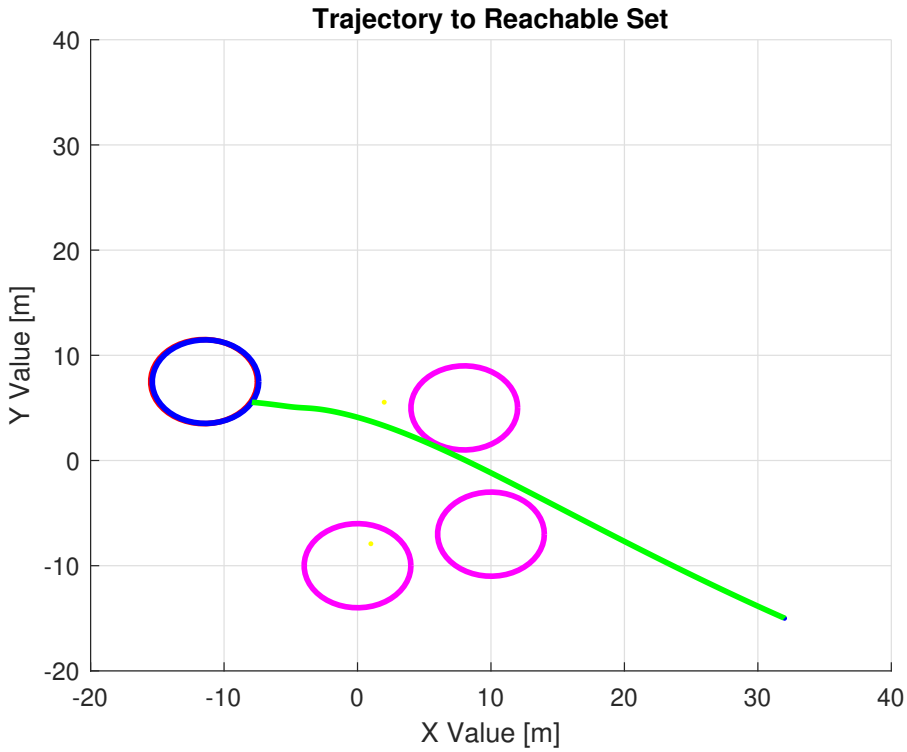


Figure 8.5: Trajectory with Three Obstacles, all colored in magenta, in an alternate setting.

Further strictifying Figure 8.4's setup produces the results demonstrated in Figure 8.6.



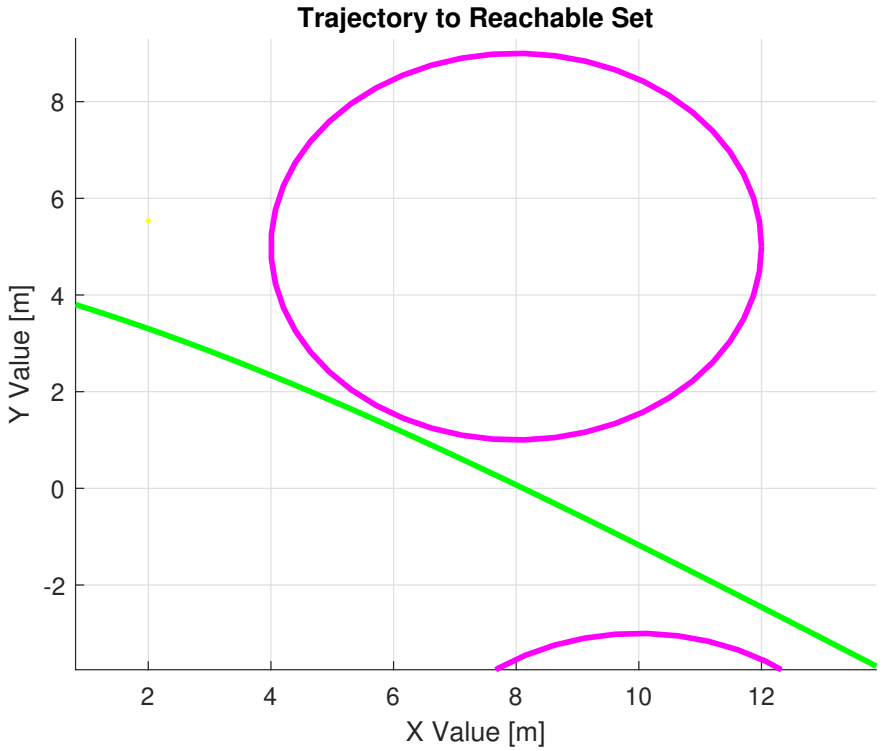


Figure 8.6: Stricter Environment. The algorithm dodges every obstacle, maneuvering between them.

The environment in the figures above is stricter than the one in *Fig.* 8.4 in terms of space. The obstacle with the highest y-axis value has now moved down by one point, thus limiting the available movement space. The trajectory now is sharper and has two edges, because it must squeeze more between the obstacles.

It would be worth highlighting here the process that a drone-agent would follow for the system of *Fig.* 8.6b. Without the Q-network, the drone would have blindly followed the direction determined by the black arrow, thus certainly colliding with the object. However, the critic's update, the estimation of the Q value and the adaptation of the control laws successfully informed the drone about the danger in real time, therefore changing its course and guaranteeing its safety.

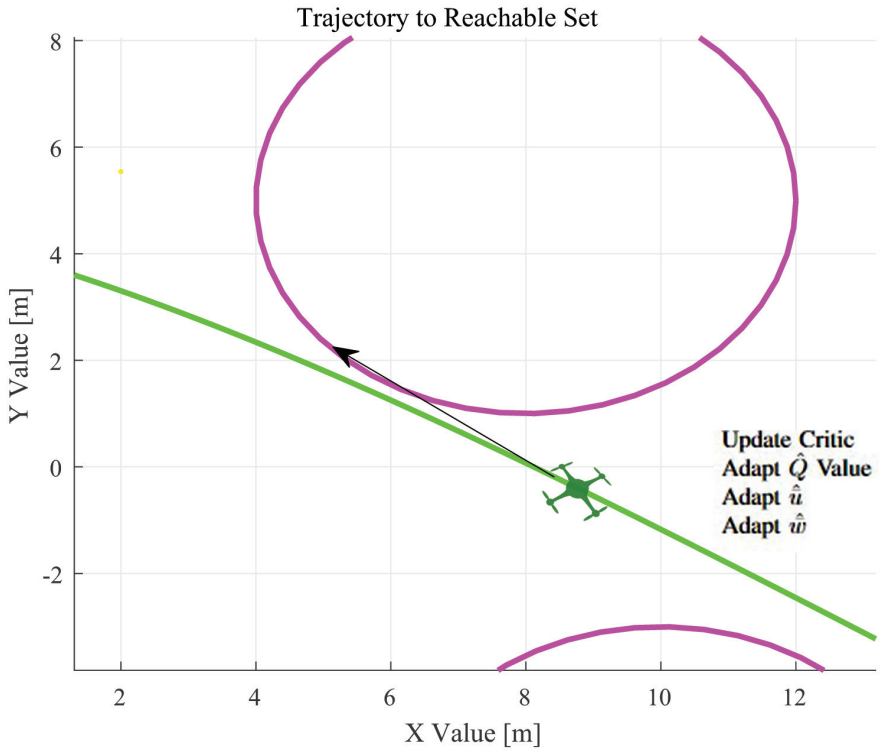


Figure 8.7: After the critic network is updated and the estimated values are adapted, the drone successfully avoids the obstacle. The final trajectory is depicted in green.

Chapter 9

Results

This section contains details regarding the computational complexity of the proposed algorithm. Furthermore, the algorithmic framework's limits are outlined and instructions for the implementation in a real-world scenario are provided.

9.0.1 Computational Complexity

The time complexity of the Q-function estimation is determined by the equations (7.10) and (7.26). In particular, the critic network expands quadratically with the size of the augmented state $\hat{U} \in \mathbb{R}^{n+m+p}$, which translates to $\mathcal{O}((n+m+p)^2)$. The input's control estimation is determined by (7.13) and (7.27), where the time yields $\mathcal{O}(nm)$. Similarly, the disturbance estimation is determined by (7.14) and (7.28), where the time yields $\mathcal{O}(np)$. The gradient descent estimation procedures outlined in [38] are essentially what the computational needs rely on. Therefore, the overall time complexity of the online framework is $\mathcal{O}((n+m+p)^2+n(m+p)) = \mathcal{O}((n+m+p)^2)$.

9.0.2 Limitations

The proposed framework's limitations must be noted to assess its applicability in each scenario. This model-free algorithm can be used in continuous-time, unknown linear systems. Although the nonlinear case is not concerned, if a nonlinear system can be linearized around an equilibrium point, and if the system is detectable and controllable, the methodology does hold.

9.0.3 Implementation Instructions

The proposed framework does not require many specifications to be properly implemented. The online, model-free control of the robot needs an accurate linear state, input, and disturbance feedback to be fed to the system. Since the system is detectable, the state can be properly computed if the output is known.

Chapter 10

Conclusions

In this thesis, the initial chapter delves into optimal control, reachability analysis, reach-avoid analysis and reachability decomposition, and their application to a 6D quadrotor system's simulations; first on the full system, then on the same system with Gaussian noise and, lastly, on the decomposed system. The computational speed of the latter's BRS was notably faster, solving the problem in 4 minutes compared to 12 minutes for the full systems on a PC with an Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz and 16GB RAM. The decomposition method was the traditional one, based on dynamic programming.

The second chapter explores the theory of reinforcement learning (RL) and its application in reachability analysis, demonstrated through Lunar Lander system simulations, again in the context of the full system and then the decomposed one, this time with a variant of Q-Learning. The decomposed system's reach-avoid set reconstruction was significantly faster, running for 2h25m on an NVIDIA GeForce GTX 1050 GPU (61% faster) and 4h52m on a Linode CPU with 8GB RAM and 4 cores (53.3% faster). However, the precision of the decomposed system was about 43%, which was expected for an approximate decomposition.

Chapter 3 provides a summary of Chapters 4-9 in Greek, while from Chapter 4 the main project unfolds. This project proposed an online, model-free reachable set computation algorithm. More specifically, the optimal policy of a continuous LTI system is approximated through Q-Learning and the reachable set is computed. Then, a sample trajectory of the algorithm is executed. The asymptotic stability is mathematically proven to be guaranteed for systems of unknown/uncertain dynamics. By leveraging simulation data, the efficacy of the framework was validated further, not only on the theoretical level.

Bibliography

[1] R. Vinter, *Optimal Control*. MA: Birkhauser Boston, 2010.

[2] L. Pontryagin, *Mathematical Theory of Optimal Processes* (Classics of Soviet Mathematics). Taylor & Francis, 1987.

[3] M. Athans and P. Falb, *Optimal Control: An Introduction to the Theory and Its Applications* (Dover Books on Engineering). Dover Publications, 2007.

[4] "Optimal control of continuous-time systems", in *Optimal Control*. John Wiley and Sons, Ltd, 2012, ch. 3, pp. 110–176.

[5] "Static optimization", in *Optimal Control*. John Wiley and Sons, Ltd, 2012, ch. 1, pp. 1–18.

[6] D. Kirk, *Optimal Control Theory: An Introduction* (Dover Books on Electrical Engineering Series). Dover Publications, 2004.

[7] "Differential games", in *Optimal Control*. John Wiley and Sons, Ltd, 2012, ch. 10, pp. 438–460.

[8] I. Chahma, *Set-valued discrete approximation of state-constrained differential inclusions* (Bayreuther mathematische Schriften). Mathematisches Institut der Universität Bayreuth, 2003.

[9] M. Dellnitz, O. Junge, M. Post, and B. Thiere, "On target for venus – set oriented computation of energy efficient low thrust trajectories", *Celestial Mechanics and Dynamical Astronomy*, vol. 95, no. 1, pp. 357–370, May 2006.

[10] R. Baier, M. Gerdts, and I. Xausa, "Approximation of reachable sets using optimal control algorithms", 2012.

[11] S. Bansal, M. Chen, S. L. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances", *CoRR*, vol. abs/1709.07523, 2017.

[12] I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by hamilton-jacobi projections", *Journal of Scientific Computing*, vol. 19, no. 1, pp. 323–346, Dec. 2003.

[13] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, *Reach-avoid problems with time-varying dynamics, targets and constraints*, 2014.

[14] M. Chen, S. Herbert, and C. J. Tomlin, "Fast reachable set approximations via state decoupling disturbances", in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, Dec. 2016.

[15] M. Chen, S. Herbert, and C. J. Tomlin, *Exact and efficient hamilton-jacobi-based guaranteed safety analysis via system decomposition*, 2016.

[16] S. Herbert. "Coupled control reconstruction array". (), [Online]. Available: https://images.squarespace-cdn.com/content/v1/5164717ee4b09befa7e93fbf/1494972105382-KP4A88VKG9O72MHGI4TX/coupled_control_table.png?format=1500w.

[17] S. Herbert. "Approximate decomposition method". (), [Online]. Available: https://images.squarespace-cdn.com/content/v1/5164717ee4b09befa7e93fbf/1494978689550-OOG5CJC1LEQYM6HVHBGP/approx.gif?format=2500w.

[18] I. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems", Jan. 2002.

[19] OpenAI. "Agent-environment interaction loop". (2018), [Online]. Available: https://spinningup.openai.com/en/latest/_images/rl_diagram_transparent_bg.png.

[20] J. Achiam. "Key concepts in rl". (2018), [Online]. Available: https://spinningup.openai.com/en/latest/spinningup/rl_intro.html (visited on 03/17/2023).

[21] C. Watkins, "Learning from delayed rewards", Jan. 1989.

[22] C. J. C. H. Watkins and P. Dayan, "Q-learning", *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Playing atari with deep reinforcement learning", *CoRR*, vol. abs/1312.5602, 2013.

[24] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation", in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12, MIT Press, 1999.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017.

[26] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning", *Machine learning*, vol. 8, pp. 229–256, 1992.

[27] W. R. J., "Toward a theory of reinforcement-learning connectionist systems", *Technical Report*, 1988.

[28] R. S. Sutton, “Integrated architectures for learning, planning, and reacting based on approximating dynamic programming”, in *Machine learning proceedings 1990*, Elsevier, 1990, pp. 216–224.

[29] G. Brockman, V. Cheung, L. Pettersson, *et al.*, *Openai gym*, 2016.

[30] Kai-Chieh Hsu and Vicenç Rúbies Royo and Claire J. Tomlin and Jaime F. Fisac, “Safety and Liveness Guarantees through Reach-Avoid Reinforcement Learning”, *CoRR*, vol. abs/2112.12288, 2021.

[31] Liberzon, Daniel, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. USA: Princeton University Press, 2011.

[32] Botan, C. and Onea, A., “A fixed end-point problem for an electrical drive system”, in *ISIE ’99. Proceedings of the IEEE International Symposium on Industrial Electronics (Cat. No.99TH8465)*, vol. 3, 1999, 1345–1349 vol.3.

[33] ARTHUR BRYSON, JR., “Applied Optimal Control: Optimization, Estimation, and Control (1st ed.)”, in *Guidance, Navigation and Control Conference*.

[34] ARTHUR BRYSON, JR. and ALAIN CARRIER, “A comparison of control synthesis using differential games (H-infinity) and LQR”, in *Guidance, Navigation and Control Conference*.

[35] Kontoudis, George P. and Vamvoudakis, Kyriakos G., “Kinodynamic Motion Planning With Continuous-Time Q-Learning: An Online, Model-Free, and Safe Navigation Framework”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3803–3817, 2019.

[36] Ioannou, Petros A and Sun, Jing, *Robust adaptive control*. Courier Corporation, 2012.

[37] Draguna L. Vrabie and Kyriakos G. Vamvoudakis and Frank L. Lewis, “Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles”, 2012.

[38] Baldi, P., “Gradient descent learning algorithm overview: a general dynamical systems perspective”, *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 182–195, 1995.

[39] B. Heater. “Amazon debuts a fully autonomous warehouse robot”. (2022), [Online]. Available: `https://techcrunch.com/2022/06/22/amazon-debuts-a-fully-autonomous-warehouse-robot/` (visited on 01/14/2023).

[40] D.P. Bertsekas and I.B. Rhodes, “On the minimax reachability of target sets and target tubes”, *Automatica*, vol. 7, no. 2, pp. 233–247, 1971.

[41] Liberzon, Daniel, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ: Princeton University Press, 2011.

[42] Bansal, Somil, S. Chen Mo and Herbert, and C. J. Tomlin, “Hamilton-Jacobi reachability: A brief overview and recent advances”, in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 2242–2253.

[43] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ: John Wiley & Sons, 2012.

[44] Ian M. Mitchell, “The Flexible, Extensible and Efficient Toolbox of Level Set Methods”, *Journal of Scientific Computing*, vol. 35, pp. 300–329, 2-3 Jun. 2008.

[45] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*. Hoboken, NJ: John Wiley & Sons, 2004, vol. 2.

[46] Powell, Warren B, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: John Wiley & Sons, 2007, vol. 703.

[47] F. Wang and H. Zhang and D. Liu, “Adaptive Dynamic Programming: An Introduction”, *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 39–47, 2009.

[48] Sylvia L. Herbert and Mo Chen and SooJean Han and Somil Bansal and Jaime F. Fisac and Claire J. Tomlin, “FaSTrack: A modular framework for fast and guaranteed safe motion planning”, in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, Dec. 2017.

[49] Sutton, Richard S and Barto, Andrew G, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT press, 2018.

[50] Vamvoudakis, Kyriakos G and Kokolakis, Nick-Marios T, *Synchronous Reinforcement Learning-Based Control for Cognitive Autonomy*. Boston - Delft: Now Publishers, 2020, vol. 8.

[51] Kokolakis, Nikolaos-Marios T. and Vamvoudakis, Kyriakos G., “Safety-Aware Pursuit-Evasion Games in Unknown Environments Using Gaussian Processes and Finite-Time Convergent Reinforcement Learning”, *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.

[52] I. Grondman, M. Vaandrager, L. Busoniu, R. Babuska, and E. Schuitema, “Efficient Model Learning Methods for Actor–Critic Control”, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 591–602, 2012.