



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ  
ΕΡΓΑΣΤΗΡΙΟ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Αυτόνομη Ρομποτική Πλοήγηση με χρήση της  
μεθόδου Koopman

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΑΝΔΡΕΑ Π ΚΕΛΙΡΗ

Επιβλέπων: ΧΑΡΑΛΑΜΠΟΣ Ε ΨΥΛΛΑΚΗΣ  
ΛΕΚΤΟΡΑΣ ΕΜΠ

Αθήνα, Μάρτιος 2024





**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ  
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ  
ΕΡΓΑΣΤΗΡΙΟ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Αυτόνομη Ρομποτική Πλοήγηση με χρήση της  
μεθόδου Koopman

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

**ΑΝΔΡΕΑ Π ΚΕΛΙΡΗ**

Επιβλέπων: ΧΑΡΑΛΑΜΠΟΣ Ε ΨΥΛΛΑΚΗΣ  
ΛΕΚΤΟΡΑΣ ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26 Μαρτίου.

.....  
ΧΑΡΑΛΑΜΠΟΣ Ε  
ΨΥΛΛΑΚΗΣ ΛΕΚΤΟΡΑΣ  
ΕΜΠ

.....  
ΚΩΝΣΤΑΝΤΙΝΟΣ Σ  
ΤΖΑΦΕΣΤΑΣ  
ΑΝΑΠΛΗΡΩΤΗΣ  
ΚΑΘΗΓΗΤΗΣ ΕΜΠ

.....  
ΣΑΒΒΑΣ Γ ΛΟΙΖΟΥ  
ΑΝΑΠΛΗΡΩΤΗΣ  
ΚΑΘΗΓΗΤΗΣ ΤΕΠΑΚ

Αθήνα, Μάρτιος 2024.

.....  
**ΑΝΔΡΕΑΣ Π ΚΕΛΙΡΗΣ**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Copyright © Κελίρης Ανδρέας, 2024.

Με επιφύλαξη παντός δικαιώματος, All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Η σχεδίαση νόμων ελέγχου για την πλοήγηση αυτόνομων ρομπότ, είναι ένα πρόβλημα το οποίο έχει απασχολήσει έντονα τα τελευταία χρόνια την ερευνητική κοινότητα. Η ευρεία χρήση τους σε πολυάριθμους τομείς της επιστήμης, καθιστά την ανάπτυξη διαφορετικών μεθοδολογιών πλοήγησης, μείζονος σημασίας. Σε αυτήν την διπλωματική, θα αναζητήσουμε λύσεις, στο πρόβλημα σχεδίασης ελέγχου πλοήγησης ενός αυτόνομου ρομπότ, μέσα σε ένα καρτεσιανό χώρο με στατικά εμπόδια, από ένα σημείο εκκίνησης μέχρι το τελικό σημείο-στόχος. Θα αναπτύξουμε μεθοδολογίες κατασκευής τεχνητών αρμονικών πεδίων και δυναμικών ροών γύρω από τα εμπόδια, καθώς και από το σημείο τερματισμού. Οι δυναμικές ροές γύρω από τα εμπόδια πρέπει να δημιουργούν μία απωστική δύναμη πάνω στο ρομπότ, ενώ όπως είναι φυσιολογικό, από το σημείο τερματισμού πρέπει να παράγεται μία ελκτική δύναμη που θα καθοδηγεί το ρομπότ προς αυτό. Αυτή η προσέγγιση φυσικά, δεν είναι δημιουργία του συγγραφέως και αρκετά ερευνητικά κείμενα έχουν αναπτύξει νόμους ελέγχου με βάση αυτή την μεθοδολογία. Στο [8] με βάση όλα τα παραπάνω, σχεδιάζεται ένας νόμος ελέγχου για έναν καρτεσιανό χώρο εργασίας με σημειακά εμπόδια, ενώ στο [6] προτείνεται ένας μετασχηματισμός του χώρου εργασίας στον σημειακό χώρο (point world). Το καινοτόμο στοιχείο αυτής της εργασίας, είναι η υλοποίηση των αρμονικών πεδίων και δυναμικών ροών με τη βοήθεια ενός μαθηματικού τελεστή, γνωστός σήμερα ως τελεστής Koopman. Η βασική ιδιότητα αυτού του τελεστή, ο οποίος έχει ονομαστεί προς τιμήν του Αμερικάνου μαθηματικού Bernard Osgood Koopman, είναι η μετατροπή μη γραμμικών δυναμικών συστημάτων σε αντίστοιχα γραμμικά, έστω και προσεγγιστικά, χωρίς φυσικά την απώλεια πληροφορίας των συστημάτων. Η υλοποίηση του νόμου ελέγχου με τη βοήθεια του τελεστή Koopman αποφεύγει πολλά προβλήματα που σχετίζονται με τις φυσικές διαστάσεις των εμποδίων, τα οποία θα αναλυθούν λεπτομερώς στην συνέχεια της διπλωματικής, και παράγει έναν στιβαρό νόμο πλοήγησης.

Στα επόμενα κεφάλαια θα παρουσιάσουμε τις ιδιότητες του τελεστή Koopman καθώς και όλες τις λεπτομέρειες της πλοήγησης αυτόνομων ρομπότ. Θα επιστημονοποιήσουμε τα προβλήματα που δημιουργούνται και θα προτείνουμε μία μεθοδολογία για την κατασκευή του κατάλληλου νόμου ελέγχου.

### Λέξεις κλειδιά:

Πλοήγηση, ρομπότ, τελεστής, δυναμικές ροές, Koopman, αυτόνομη, έλεγχος, αρμονικότητα, εμπόδια, χώρος εργασίας, φυσικές διαστάσεις, μεθοδολογία, τεχνητά πεδία.

## Abstract

Designing a control law for the navigation of autonomous robots is an important problem that has attracted significant research interest in recent years. Their widespread use in numerous fields of science makes the development of various navigation methodologies of great importance. In this thesis, we will seek solutions to the problem of designing control for the navigation of an autonomous robot, within the boundaries of a cartesian space filled with static obstacles, from an initial starting point to a final destination point. We will develop methodologies for constructing artificial harmonic fields and dynamic flows around obstacles, as well as towards the destination point. The dynamic flows around obstacles should create a repulsive force on the robot, while, as expected, for the destination point, an attractive force should guide the robot towards it. This approach, of course, is not new, and several research papers have developed control laws based on this methodology. In [8], based on all of the above, a control law is designed for a Cartesian workspace with point obstacles, while in [6], a transformation of the workspace into the point space (point world) is proposed. The innovative element of this work is the implementation of harmonic fields and dynamic flows with the help of a mathematical operator, known as the Koopman operator. The fundamental property of this operator, named in honor of the American mathematician Bernard Osgood Koopman, is the conversion of finite-dimensional nonlinear dynamic systems into corresponding infinite-dimensional linear ones, albeit approximately, without, of course, losing information from the systems. Implementing the control law with the help of the Koopman operator avoids many problems related to the physical dimensions of obstacles, which will be analyzed in detail later in the thesis, and produces a robust navigation law. In the following chapters, we will present the properties of the Koopman operator as well as all the details of autonomous robot navigation. We will identify the problems that arise and propose a methodology for constructing the appropriate control law.

### *Keywords:*

Navigation, robot, operator, potential flows, Koopman, autonomous, control, harmonicity, obstacles, workspace, physical dimensions, methodology, artificial fields

# Περιεχόμενα

<b>1</b>	<b>Πλοήγηση Αυτόνομου Ρομπότ</b>	<b>4</b>
1.1	Εισαγωγή στο πρόβλημα πλοήγησης . . . . .	4
1.2	Αρμονικές συναρτήσεις και δυναμικές ροές . . . . .	5
1.3	Πλοήγηση σε χώρο με σημειακά εμπόδια . . . . .	6
<b>2</b>	<b>Μεθοδολογία Koopman</b>	<b>9</b>
2.1	Εισαγωγή στη μέθοδο Koopman . . . . .	10
2.2	Τελεστής Koopman . . . . .	11
2.3	Ιδιότητες . . . . .	11
2.4	Ιδιοσυναρτήσεις και ιδιοτιμές . . . . .	12
2.5	Αναπαράσταση στον χώρο των ιδιοσυναρτήσεων . . . . .	13
2.6	Προσέγγιση του τελεστή Koopman με αριθμητικές μεθόδους . . . . .	14
<b>3</b>	<b>Εφαρμογή μεθόδου Koopman για την κατασκευή συναρτήσεων πλοήγησης</b>	<b>16</b>
3.1	Εισαγωγή . . . . .	16
3.2	Περιβάλλον εργασίας και ρομποτικό μοντέλο . . . . .	16
3.3	Δυναμικές ροές . . . . .	17
3.4	Ιδιοσυναρτήσεις του δυναμικού συστήματος πλοήγησης . . . . .	19
3.5	Panel Method . . . . .	20
3.6	Προσομοίωση με διαφορετικό πλήθος ιδιοσυναρτήσεων . . . . .	23
3.6.1	Χρήση 5 ιδιοσυναρτήσεων . . . . .	23
3.6.2	Χρήση 30 ιδιοσυναρτήσεων . . . . .	24
3.6.3	Χρήση 50 ιδιοσυναρτήσεων . . . . .	26
<b>4</b>	<b>Εφαρμογές και αποτελέσματα</b>	<b>28</b>
4.1	Εμπόδιο ευθύγραμμου τμήματος . . . . .	28
4.2	Περιβάλλον με δύο εμπόδια ευθύγραμμου τμήματος . . . . .	30
4.3	Περιβάλλον με ένα μη κυρτό εμπόδιο . . . . .	32
4.4	Λαβύρινθος . . . . .	36
4.5	Συμπεράσματα . . . . .	37

<b>5</b>	<b>Παράρτημα</b>	<b>39</b>
5.1	Παράρτημα I . . . . .	39
5.2	Παράρτημα II . . . . .	44
5.3	Παράρτημα III . . . . .	46
5.4	Παράρτημα IV . . . . .	47
5.5	Παράρτημα V . . . . .	49



# Κεφάλαιο 1

## Πλοήγηση Αυτόνομου Ρομπότ

### 1.1 Εισαγωγή στο πρόβλημα πλοήγησης

Το κύριο θέμα αυτής της διπλωματικής είναι η σχεδίαση νόμου ελέγχου για την αυτόνομη πλοήγηση ενός ρομπότ μέσα σε έναν καρτεσιανό χώρο εμπλουτισμένο με εμπόδια φυσικών διαστάσεων. Θεωρούμε ότι τα εμπόδια είναι στατικά και η θέση τους είναι γνωστή στο ρομπότ. Επιπρόσθετα το σημείο εκκίνησης ανήκει στο σύνολο  $\mathbb{S}$  το οποίο περιέχει όλα τα σημεία του καρτεσιανού χώρου, εκτός από το τελικό σημείο και τα σημεία στα οποία βρίσκονται τα εμπόδια. Το τελικό σημείο-στόχος είναι προκαθορισμένο. Αυτή η συνθήκη αυξάνει τον βαθμό δυσκολίας της σχεδίασης του επιθυμητού νόμου ελέγχου, εφόσον το σημείο εκκίνησης δεν είναι συγκεκριμένο. Επομένως, μία απλή μέθοδος σχεδίασης τροχιάς δεν μπορεί να μας οδηγήσει στο επιθυμητό αποτέλεσμα. Συνεπώς, πρέπει να αναζητήσουμε μία εναλλακτική προσέγγιση στο πρόβλημα, η οποία θα εξασφαλίζει ότι το ρομπότ θα αποφεύγει οποιαδήποτε σύγκρουση με τα εμπόδια και παράλληλα θα καταφθάνει στο τελικό σημείο. Μία από τις μεθόδους επίλυσης αυτού του προβλήματος είναι η κατασκευή **συναρτήσεων πλοήγησης**. Αυτή η έννοια εισήχθη για πρώτη φορά από τους Daniel E. Koditschek και Elon Rimon στο [4]. Η βασική ιδέα, είναι η κατασκευή μίας βαθμωτής συνάρτησης, η οποία θα παράγει ένα διανυσματικό πεδίο το οποίο θα οδηγεί όλες τις τροχιές, ανεξαρτήτως των αρχικών συνθηκών, στον επιθυμητό προορισμό. Για να το πετύχουμε αυτό, η συνάρτηση πλοήγησης θα πρέπει να παρουσιάζει ένα και μόνο ένα (ολικό) ελάχιστο στο σημείο προορισμού. Η ύπαρξη τοπικών ελαχίστων σε διάφορα σημεία του χώρου μπορούν να οδηγήσουν το ρομπότ σε άδιέξοδο και τότε ο νόμος ελέγχου έχει αποτύχει. Οι συναρτήσεις που ταιριάζουν απόλυτα στο ιδανικό μοντέλο πλοήγησης, ονομάζονται **αρμονικές συναρτήσεις**.

## 1.2 Αρμονικές συναρτήσεις και δυναμικές ροές

Όπως προαναφέραμε στο προηγούμενο κεφάλαιο, η επιθυμητή συνάρτηση πλοήγησης πρέπει να εμφανίζει ένα ολικό ελάχιστο στο τελικό σημείο και φυσικά κανένα τοπικό ελάχιστο στα υπόλοιπα σημεία του χώρου. Επιπρόσθετα οφείλει να εξασφαλίζει ότι το ρομπότ, υπό οποιαδήποτε συνθήκη, δε θα έρθει σε σύγκρουση με τα εμπόδια που βρίσκονται στο χώρο. Στην ουσία το διανυσματικό πεδίο που παράγεται από την βαθμίδα της συνάρτησης πλοήγησης, επιβάλλεται να δημιουργεί απωστικές δυνάμεις γύρω από τα εμπόδια και ελκτική δύναμη από το σημείο προορισμού. Με πρώτη ματιά, η κατασκευή τέτοιας συνάρτησης δεν φαίνεται τόσο εύκολη. Ειδικότερα, όσο αυξάνεται ο αριθμός των εμποδίων τόσο μεγαλώνει ο βαθμός δυσκολίας του προβλήματος. Η λύση βρίσκεται στην διάσπαση της συνάρτησης πλοήγησης σε μικρότερα κομμάτια, όπου το κάθε ένα θα ικανοποιεί μία συνθήκη απώθησης ή έλξης. Πιο συγκεκριμένα, χρειαζόμαστε μία συνάρτηση που θα δημιουργεί ελκτική ροή και θα αντιστοιχεί στο σημείο προορισμού και διαφορετικές συναρτήσεις για κάθε εμπόδιο, όπου η κάθε μία θα παράγει απωστικές ροές από το εμπόδιο που της αντιστοιχεί. Φυσικά, η τελική συνάρτηση πλοήγησης θα προκύψει από την υπέρθεση όλων αυτών των συναρτήσεων. Εδώ όμως παρουσιάζεται ένα πρόβλημα. Η υπέρθεση όλων των συναρτήσεων δεν εξασφαλίζει ότι η ολική συνάρτηση θα κληρονομήσει όλες τις ιδιότητες των υπο-συναρτήσεων. Αν το σκεφτούμε διαισθητικά, υπάρχει κίνδυνος για εμφάνιση τοπικών ελαχίστων στον καρτεσιανό χώρο που θα οφείλονται στην αλληλεπίδραση των ροών που αντιστοιχούν σε διαφορετικά εμπόδια. Επομένως, για την εύρυθμη λειτουργία της πλοήγησης, όταν το ρομπότ βρίσκεται κοντά σε ένα εμπόδιο, τότε θα πρέπει να 'ιώθει' μόνο την απωστική δύναμη αυτού του εμποδίου και να μην λαμβάνει υπόψιν τις ροές των υπόλοιπων εμποδίων. Ένας τύπος συνάρτησης που μας εξασφαλίζει όλα τα παραπάνω, είναι οι **αρμονικές συναρτήσεις**.

Εξ ορισμού, αρμονικές συναρτήσεις ονομάζονται οι συναρτήσεις οι οποίες είναι δύο φορές διαφορίσιμες και ταυτόχρονα ικανοποιούν την διαφορική εξίσωση *Laplace* η οποία στον καρτεσιανό χώρο είναι:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0 \quad (1.1)$$

Αυτές οι συναρτήσεις λοιπόν, παρουσιάζουν διάφορες ιδιότητες οι οποίες ταιριάζουν απόλυτα στις απαιτήσεις που έχουμε για την συνάρτηση πλοήγησης. Αρχικά, εάν θεωρήσουμε ότι η αρμονική συνάρτηση  $f$  είναι περιορισμένη σε ένα υποσύνολο  $K \subseteq U$ , όπου το  $U$  είναι συνεκτικό, τότε η  $f$  δεν παρουσιάζει τοπικά ακρότατα. Επομένως η συγκεκριμένη ιδιότητα, μας εξασφαλίζει ότι η αρμονική συνάρτηση δεν έχει τοπικά ελάχιστα, κάτι το οποίο είναι απολύτως ευνοϊκό για την κατασκευή της συνάρτησης πλοήγησης. Όμως το πιο σημαντικό στοιχείο των αρμονικών συναρτήσεων, είναι ότι ικανοποιούν την συνθήκη της υπέρθεσης. Δηλαδή, ο γραμμικός συνδυασμός δύο ή περισσότερων αρμονικών συναρτήσεων, είναι πάλι μία αρμονική συνάρτηση. Επιλέγοντας λοιπόν, αρμονικές υπο-συναρτήσεις εξασφαλίζουμε ότι και η ολική συνάρτηση

πλοήγησης θα κληρονομήσει ανεξάρτητα όλες τις ιδιότητες των υπο-συναρτήσεων, με αποτέλεσμα οι ροές που παράγονται από κάθε υπο-συνάρτηση να μην αλληλεπιδρούν μεταξύ τους.

Ας επικεντρωθούμε τώρα στις ροές που παράγουν οι αρμονικές συναρτήσεις. Όπως γνωρίζουμε από την ρευστοδυναμική, η βαθμίδα μίας βαθμωτής συνάρτησης ονομάζεται **δυναμική ροή** και μαθηματικά περιγράφεται από τη εξίσωση  $V = \nabla\phi$ . Επιπλέον αυτές οι ροές παρουσιάζουν μία πολύ σημαντική ιδιότητα, έχουν μηδενικό στροβιλισμό. Δηλαδή, εάν τοποθετήσουμε μία άμορφη μάζα μέσα στην δυναμική ροή και την αφήσουμε να κινηθεί ελεύθερα, θα παρατηρήσουμε ότι δεν θα τελέσει καμία περιστροφική κίνηση. Ας υποθέσουμε τώρα, ότι έχουμε μία δυναμική ροή  $\phi$  η οποία παράγεται από μία **βαθμωτή αρμονική συνάρτηση**, τότε αυτή:

1. Είναι αστρόβιλη,

$$\nabla \times \nabla\phi = 0 \quad (1.2)$$

2. Είναι ασυμπίεστη,

$$\nabla \cdot \nabla\phi = 0 \quad (1.3)$$

3. Είναι στατική,

$$\frac{\partial\phi}{\partial t} = 0 \quad (1.4)$$

Δηλαδή, η επιλογή μίας αρμονικής  $\phi$  δημιουργεί μία δυναμική ροή με δύο επιπρόσθετες ιδιότητες, οι οποίες μας εξασφαλίζουν ότι οι τροχιές του διανυσματικού πεδίου θα είναι ασυμπίεστες και σταθερές στον χρόνο. Τα τρία παραπάνω χαρακτηριστικά, καθιστούν τις δυναμικές ροές αρμονικών συναρτήσεων, ως ένα υπερπολύτιμο εργαλείο σε τομείς όπως η ρομποτική πλοήγηση και η αεροδυναμική. Παρόμοια με τις αρμονικές συναρτήσεις, η γραμμική υπέρθεση των δυναμικών ροών, έχει ως αποτέλεσμα μία καινούρια δυναμική ροή η οποία εξακολουθεί να διαθέτει τις ιδιότητες (1.2), (1.3) και (1.4). Επομένως, αν θέλουμε να κατευθύνουμε το ρομπότ γύρω από ένα εμπόδιο, μπορούμε να συνθέσουμε μία κατάλληλη δυναμική ροή με τη μέθοδο της υπέρθεσης, που θα καθοδηγεί το ρομπότ στην επιθυμητή τροχιά. Αυτή η προσέγγιση ταυτίζεται ουσιαστικά με την μέθοδο της υπέρθεσης των αρμονικών συναρτήσεων, που προαναφέραμε, υπό την εξής έννοια. Είτε θα σχεδιάσουμε τις κατάλληλες αρμονικές συναρτήσεις και θα αποκτήσουμε τις κατάλληλες δυναμικές ροές από την βαθμίδα τους (gradient), είτε θα ακολουθήσουμε την δεύτερη προσέγγιση και θα σχεδιάσουμε απευθείας τις δυναμικές ροές.

### 1.3 Πλοήγηση σε χώρο με σημειακά εμπόδια

Στο κύριο αντικείμενο αυτής της διπλωματικής, θα αναφερθούμε σε καρτεσιανούς χώρους με εμπόδια φυσικών διαστάσεων. Παρόλα αυτά, η απλοποίηση των διαστάσεων των εμποδίων θα μας βοηθήσει να κατανοήσουμε καλύτερα το πρόβλημα της πλοήγησης. Μάλιστα, στο [6] παρουσιάζεται μία μεθοδολογία μετατροπής ενός

άμορφου χώρου με εμπόδια φυσικών διαστάσεων, σε έναν κυκλικό χώρο με σημειακά εμπόδια. Όπως έχει αποδειχθεί, στη δεύτερη περίπτωση έχουμε να αντιμετωπίσουμε ευνοϊκότερες συνθήκες. Ας θεωρήσουμε λοιπόν, ένα ρομποτικό κινηματικό μοντέλο:

$$\dot{x}(t) = u(t) \quad (1.5)$$

και έναν χώρο εργασίας  $\mathcal{W}$  μέσα στον οποίο βρίσκονται σημεία 'νάρκες' (εμπόδια) και ένα επιθυμητό σημείο στο οποίο θέλουμε να οδηγήσουμε το ρομπότ. Σύμφωνα με το [8] ορίζουμε τις συναρτήσεις αρμονικού δυναμικού των εμποδίων ως:

$$\phi_i(h) = \ln(\|h - P_i\|^2) \quad (1.6)$$

και την συνάρτηση αρμονικού δυναμικού του σημείου προορισμού ως:

$$\phi_d(h) = \ln(\|h - P_d\|^2) \quad (1.7)$$

Η τελική συνάρτηση δυναμικού θα προκύψει ως υπέρθεση των δυναμικών που οφείλονται στα εμπόδια και στο σημείου προορισμού.

$$\phi_k(h) = \phi_d(h) - \frac{1}{k} \sum_{i=1}^M \phi_i(h) \quad (1.8)$$

όπου  $\phi_k : \mathcal{W} \rightarrow \mathcal{R}$ ,  $k$  μία θετική παράμετρος και  $M$  ο αριθμός των εμποδίων. Σύμφωνα με τα προαναφερθέντα, η  $\phi_k$  είναι μία αρμονική συνάρτηση καθώς προκύπτει από γραμμική υπέρθεση αρμονικών συναρτήσεων. Επομένως η βαθμίδα της θα κληρονομήσει τις τρεις ιδιότητες που αναφέραμε στην προηγούμενη παράγραφο και ουσιαστικά θα αποτελείται από δύο είδη δυναμικών ροών. Από μία ροή 'sink' που οφείλεται στο σημείο προορισμού και ελκύει το ρομποτικό σώμα, και σε  $M$  ροές 'source' που απωθούν το ρομπότ, για την αποφυγή οποιασδήποτε σύγκρουσης με τα εμπόδια. Επιπλέον παρατηρούμε ότι ισχύει  $-\lim_{h \rightarrow P_i} \phi_i(h) = \infty$  και  $\lim_{h \rightarrow P_d} \phi_d = -\infty$ , γεγονός που υποδηλώνει ότι όσο πλησιάζουμε τα εμπόδια ή το σημείο προορισμού, τόσο μεγαλώνουν οι δυνάμεις απώθησης και έλξης αντίστοιχα. Επιπρόσθετα, εφόσον η συνάρτηση  $\phi_k$  διατηρεί την αρμονικότητα των συναρτήσεων εμποδίων και σημείου προορισμού, δεν εμφανίζει κανένα τοπικό ελάχιστο, μία συνθήκη που μας εξασφαλίζει ότι το ρομπότ δεν θα 'κολλήσει' σε κανένα σημείο του χώρου εργασίας. Το [8] παρουσιάζει αναλυτικά την μαθηματική απόδειξη της παραπάνω πρότασης. Για την ολοκληρωμένη μορφή της συνάρτησης πλοήγησης, είναι αναγκαία η συμπίεση του συνόλου τιμών στο διάστημα  $[0,1]$ . Η συνάρτηση πλοήγησης προκύπτει από την σύνθεση:  $\Phi = \sigma \circ \phi_k$  όπου:

$$\sigma(x) = \frac{e^x}{1 + e^x} \quad (1.9)$$

Ας προχωρήσουμε τώρα στη περίπτωση όπου τα εμπόδια δε συμβολίζονται ως σημεία στο χώρο αλλά έχουν φυσικές διαστάσεις, στη προκειμένη διπλωματική δύο στον αριθμό τους καθώς θεωρούμε ως χώρο εργασίας τον καρτεσιανό χώρο. Τι μας εμποδίζει να χρησιμοποιήσουμε την παραπάνω μεθοδολογία σχεδίασης; Η απάντηση

κρύβεται στην απώλεια της αρμονικότητας ελέω των φυσικών διαστάσεων των εμποδίων. Πλέον δεν μας ενδιαφέρει μόνο ένα συγκεκριμένο σημείο αλλά ένα σύνολο σημείων τα οποία αποτελούν το σύνορο της επιφάνειας των εμποδίων. Οι συναρτήσεις  $\phi_i$  μας εξασφάλιζαν αρμονικότητα γύρω από ένα σημείο  $P_i$ , συνθήκη η οποία ήταν αρκετή στο προηγούμενο πρόβλημα. Τώρα όμως πρέπει να αναζητήσουμε μία καινούρια συνάρτηση η οποία θα εξασφαλίζει, όσο πιο προσεγγιστικά γίνεται, αρμονικότητα σε όλα τα σημεία  $x_d$  τα οποία ανήκουν στο σύνορο  $\partial\mathcal{O}$  του κάθε εμποδίου. Σε αυτό το σημείο να υπενθυμίσουμε ότι η προσέγγιση των [6] και [8] δίνει λύση στο πρόβλημα καθώς με τη βοήθεια κατάλληλου μετασχηματισμού μεταβαίνουμε σε ένα σημειακό χώρο εργασίας όπου δεν αντιμετωπίζουμε προβλήματα αρμονικότητας. Σε κάποιες περιπτώσεις όμως, η εύρεση του κατάλληλου μετασχηματισμού δεν είναι τόσο εύκολη, ιδιαίτερα όταν τα εμπόδια έχουν μη κυρτό σχήμα. Η προσέγγιση αυτής της διπλωματικής, αφορά την σχεδίαση κατάλληλων δυναμικών ροών γύρω από τα εμπόδια με την βοήθεια του τελεστή Koopman. Στο επόμενο κεφάλαιο θα παρουσιάσουμε αναλυτικά όλες τις ιδιότητες και τις μαθηματικές έννοιες που συνδέονται με τον τελεστή, καθώς και κάποιες εφαρμογές του σε μη γραμμικά συστήματα.

## Κεφάλαιο 2

# Μεθοδολογία Koopman

Η μαθηματική έννοια του τελεστή Koopman εισήχθη για πρώτη φορά από τον αμερικάνο μαθηματικό Bernard Osgood Koopman στο έργο[5]. Η βασική ιδέα του τελεστή ήταν η γραμμικοποίηση ενός μη γραμμικού δυναμικού συστήματος, αυξάνοντας τις διαστάσεις του διανύσματος κατάστασης. Πιο συγκεκριμένα, όλοι οι γραμμικοί οροι του διανυσματικού πεδίου  $f$  προστίθενται στο διάνυσμα κατάστασης έτσι ώστε η συνάρτηση  $f$  να μπορεί να εκφραστεί γραμμικά από το νέο διάνυσμα κατάστασης. Όμως, η εισαγωγή αυτού του όρου στο διάνυσμα κατάστασης, ως επί το πλείστον των περιπτώσεων, παράγει νέους μη γραμμικούς όρους στο διανυσματικό πεδίο. Στο παρακάτω παράδειγμα έχουμε επιλέξει ένα σύστημα με εύνοια χαρακτηριστικά για να αναδείξουμε την λογική της διαδικασίας. Για περισσότερες λεπτομέρειες όσον αφορά το συγκεκριμένο δυναμικό σύστημα ανατρέξτε στο [2].

$$\begin{aligned}\dot{x}_1 &= \mu x_1 \\ \dot{x}_2 &= \lambda(x_2 - x_1^2)\end{aligned}$$

Όπως παρατηρούμε, στη δεύτερη εξίσωση εμφανίζεται ο μη γραμμικός όρος  $x_1^2$ . Σύμφωνα με την θεωρία του Koopman θα προσθέσουμε τον όρο στο διάνυσμα κατάστασης. Το νέο σύστημα θα έχει την μορφή:

$$\frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & -\lambda \\ 0 & 0 & 2\mu \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (2.1)$$

όπου  $y_1 = x_1$ ,  $y_2 = x_2$  και  $y_3 = x_1^2$ . Αυξάνοντας την διάσταση του διανύσματος κατάστασης καταφέραμε να μετατρέψουμε το αρχικό μη γραμμικό δυναμικό σύστημα σε ένα σύστημα της μορφής  $\dot{x} = Ax$ . Βέβαια στις περισσότερες περιπτώσεις, ο πίνακας  $A$  δεν θα έχει κλειστή μορφή όπως στο (2.1). Ας δούμε για παράδειγμα το παρακάτω σύστημα:

$$\dot{x} = x^2 \quad (2.2)$$

Επαναλαμβάνουμε την διαδικασία που ακολουθήσαμε στο (2.1).

$$\frac{d}{dt} \begin{bmatrix} x \\ y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 2 & 0 & \dots \\ 0 & 0 & 0 & 3 & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \end{bmatrix} \begin{bmatrix} x \\ y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} \quad (2.3)$$

όπου  $y_1 = x^2$ ,  $y_2 = x^3$ ,  $y_3 = x^4$ . Είναι ξεκάθαρο ότι ο πίνακας  $A$  δεν έχει κλειστή μορφή. Κάθε φορά που επεκτείνουμε το διάνυσμα κατάστασης δημιουργούμε νέους μη γραμμικούς όρους στο δεξί πλευρό της εξίσωσης. Επομένως σε αυτήν την περίπτωση αναζητούμε την καλύτερη δυνατή **προσέγγιση** για το διάνυσμα κατάστασης, έτσι ώστε το προκύπτον σύστημα να παρουσιάζει, σε μεγάλο βαθμό, γραμμική συμπεριφορά..

Σε αυτό το κεφάλαιο παραθέτουμε ένα ολοκληρωμένο μαθηματικό υπόβαθρο για τον τελεστή Koopman και των ιδιοτήτων του. Θα παρουσιάσουμε βασικές μαθηματικές έννοιες και ορισμούς. Θα αναφερθούμε στο φάσμα ιδιοτιμών και στο ιδιοσύστημα του τελεστή, αναλύοντας τα ιδιαίτερα χαρακτηριστικά τους που καθιστούν τον τελεστή ως ισχυρό όπλο γραμμικοποίησης μη γραμμικών δυναμικών συστημάτων. Επιπλέον θα παρουσιάσουμε διάφορες αριθμητικές μεθόδους προσέγγισης του τελεστή. Τέλος θα δείξουμε κάποιες εφαρμογές σε μη γραμμικά συστήματα ελέγχου.

## 2.1 Εισαγωγή στη μέθοδο Koopman

Η ιδέα του μετασχηματισμού μη γραμμικών δυναμικών συστημάτων σε γραμμικά, απασχολεί εδώ και έναν αιώνα την επιστημονική κοινότητα. Η χαοτική φύση των μη γραμμικών δυναμικών συστημάτων καθιστά δύσκολη την εφαρμογή κλασικών μεθόδων ελέγχου. Όπως θα περίμενε κανείς, οι ελεγκτές που προσδίδουν την κατάλληλη απόκριση σε αυτού του τύπου τα συστήματα, θα πρέπει να έχουν και αυτοί μη γραμμικά ή και χαοτικά χαρακτηριστικά. Γεγονός το οποίο, καθιστά δύσκολη την εύρεση μεθοδευμένης σχεδίασης που στηρίζεται μόνο σε κλασικά εργαλεία της θεωρίας ελέγχου. Αντιθέτως, τα γραμμικά δυναμικά συστήματα έχουν αναλυθεί εκτενώς τα τελευταία χρόνια και ο έλεγχος τους αποτελεί μία πιο απλοϊκή και μεθοδευμένη διαδικασία, αφού η θεωρία ελέγχου προσφέρει ποικίλα μαθηματικά εργαλεία και ένα ολοκληρωμένο θεωρητικό υπόβαθρο. Επομένως, ο επιτυχής μετασχηματισμός ενός μη γραμμικού χώρου κατάστασης σε γραμμικό, έστω μέσω μίας ικανοποιητικής προσέγγισης, απλοποιεί σημαντικά το αρχικό μας πρόβλημα. Ας θεωρήσουμε ένα συνεχές αυτόνομο δυναμικό σύστημα της μορφής:

$$\dot{x} = f(x(t)), \quad (2.4)$$

όπου  $x \in \mathcal{X} \subseteq \mathcal{R}^n$  είναι το  $n$ -διάστατο διάνυσμα κατάστασης και  $f$  είναι ένα μη γραμμικό, διανυσματικό πεδίο που περιγράφει την δυναμική του συστήματος. Η ιδέα

της γραμμικοποίησης στηρίζεται στην εύρεση ενός μετασχηματισμού  $z = \phi(x)$ , ο οποίος θα μπορεί να περιγράψει την δυναμική ροή του συστήματος ως:

$$\dot{z} = Lz \quad (2.5)$$

Πλέον, το δυναμικό σύστημα είναι γραμμικό ως προς το διάνυσμα κατάστασης  $z$  και ταυτόχρονα διατηρείται η μη γραμμικότητα ως προς το διάνυσμα κατάστασης  $x$ . Στην πραγματικότητα όμως, στα περισσότερα δυναμικά συστήματα, αυτού του είδους οι μετασχηματισμοί σπάνια μπορούν να υλοποιηθούν. Η πρόταση του Koopman, ήταν ότι όσο επεκτείνουμε την διάσταση του  $z$ , πλησιάζουμε όλο και περισσότερο σε ένα δυναμικό σύστημα με γραμμική συμπεριφορά. Επομένως, μέσω της μεθόδου Koopman φάχνουμε την καλύτερη δυνατή προσέγγιση που θα μας φέρει στην μορφή (2.5). Αυτή η μεθοδολογία έμεινε 'θαμμένη' για δεκαετίες καθώς δεν ήταν εύκολη η εύρεση μίας ικανοποιητικής προσέγγισης με την αποκλειστική χρήση μαθηματικών εργαλείων. Όμως η ραγδαία ανάπτυξη των υπολογιστικών συστημάτων δημιούργησε ευνοϊκές συνθήκες για την υλοποίηση της μεθόδου Koopman. Τελικά, επανήλθε στην επιφάνεια μέσω των έργων [9] και [10] όπου αναφέρθηκε για πρώτη φορά το φάσμα Koopman αποτελούμενο από τις ιδιοσυναρτήσεις και ιδιοτιμές του τελεστή.

## 2.2 Τελεστής Koopman

Ας περιγράψουμε με πιο αυστηρές και μαθηματικές έννοιες τον τελεστή Koopman. Ας θεωρήσουμε ότι έχουμε ένα γραμμικό δυναμικό σύστημα το οποίο περιγράφεται από τον μη γραμμικό μετασχηματισμό  $S : \mathcal{X} \rightarrow \mathcal{X}$ . Επομένως οι τροχιές του συστήματος περιγράφονται ως  $S^k(x_0) \big|_{k=0}^{\infty}$ . Θεωρούμε ένα χώρο Banach  $\mathbb{F}$  'παρατηρήσιμων' συναρτήσεων  $f : \mathcal{X} \rightarrow \mathbb{C}$  του διανύσματος κατάστασης. Ο τελεστής Koopman  $U : \mathbb{F} \rightarrow \mathbb{F}$  που συσχετίζεται με τον μετασχηματισμό  $S$  ορίζεται ως:

$$Uf = f \circ S \quad (2.6)$$

Επομένως ο τελεστής  $U$  δεν δρα πάνω στο διάνυσμα κατάστασης, όπως ο τελεστής  $S$ , αλλά δρα πάνω σε **συναρτήσεις του διανύσματος κατάστασης**. Δηλαδή, περιγράφει την εξέλιξη όλων των συναρτήσεων που ανήκουν στον χώρο  $\mathcal{X}$ . Συνοπτικά, ας παρουσιάσουμε την ουσιαστική διαφορά των δύο τελεστών:

$$x_{k+1} = F(x_k) \quad (2.7)$$

$$g(x_{k+1}) = Ug(x_k) \quad (2.8)$$

## 2.3 Ιδιότητες

Ο τελεστής Koopman, ως γραμμικός τελεστής χαρακτηρίζεται από κάποιες πολύ σημαντικές ιδιότητες. Η πιο βασική από αυτές πηγάζει φυσικά από την γραμμική του φύση.



**1η Ιδιότητα:** Για δύο παρατηρήσιμες συναρτήσεις  $g_1$  και  $g_2$  ισχύει ότι:

$$K * [a_1 g_1(x) + a_2 g_2(x)] = a_1 K g_1(x) + a_2 K g_2(x) \quad (2.9)$$

Αυτή η ιδιότητα έχει μεγάλη αξία καθώς μας επιτρέπει να συνδυάσουμε αποτελέσματα προβλημάτων, τα οποία έχουμε αντιμετωπίσει ξεχωριστά. Συγκεκριμένα, σε αυτή την διπλωματική θα κατασκευάσουμε μια global συνάρτηση δυναμικής ροής ως αποτέλεσμα υπέρθεσης πολλών τοπικών ροών.

**2η Ιδιότητα:** Ας υποθέσουμε ότι η συνάρτηση δυναμικού του συστήματος είναι συνεχής και επαρκώς λεία. Τότε μπορούμε να ορίσουμε την απειροελάχιστη γενέτειρα  $\mathbb{L}$  του τελεστή Koopman ως [3]:

$$Lg := \lim_{t \rightarrow 0} \frac{Kg - g}{t} \quad (2.10)$$

Η γενέτειρα αυτή ονομάζεται και **τελεστής Lie** καθώς είναι η παράγωγος Lie της συνάρτησης  $g$  πάνω στο διανυσματικό πεδίο  $f(x)$ , όταν έχουμε ένα δυναμικό πεδίο της μορφής (2.4). Αυτό φυσικά αποδεικνύεται εύκολα εάν επεκτείνουμε την χρονική παράγωγο της συνάρτησης  $g$ :

$$\frac{dg(x(t))}{dt} = \nabla g \cdot \dot{x}(t) = \nabla g \cdot f(x(t)), \quad (2.11)$$

και μέσω της σχέσης (2.10) μπορούμε να γράψουμε:

$$Lg = \nabla g \cdot f \quad (2.12)$$

Άρα σε συνεχή και λεία δυναμικά μπορούμε να συνδέσουμε τον τελεστή  $L$  με το μη γραμμικό διανυσματικό πεδίο  $f$ , μία ιδιότητα που μας επιτρέπει να υπολογίσουμε τις ιδιοσυναρτήσεις και ιδιοτιμές του τελεστή.

**3η ιδιότητα:** Ο τελεστής Koopman έχει έναν συζυγή τελεστή  $\mathbb{U}^*$  ο οποίος δρα στο συζυγές χώρο  $\mathbb{F}^*$  τέτοιος ώστε:

$$\xi(Uf) = (U^*\xi)(f) \quad (2.13)$$

Επίσης ισχύει ότι  $UU^* = I$ , όπου  $I$  ο μοναδιαίος τελεστής.

## 2.4 Ιδιοσυναρτήσεις και ιδιοτιμές

Ο τελεστής Koopman όπως κάθε γραμμικός τελεστής όταν δρα πάνω σε ένα τυχαίο διάνυσμα του διανυσματικού χώρου, κατά κανόνα, προκαλεί αλλαγές στον προσανατολισμό και στο μέτρο του. Όμως αυτός ο κανόνας έχει εξαιρέσεις καθώς υπάρχουν διανύσματα τα οποία κρατούν άθικτο τον προσανατολισμό τους. Αυτά τα διανύσματα ονομάζονται ιδιοδιανύσματα ή στην δική μας περίπτωση, εφόσον αναφερόμαστε σε

τελεστή που δρα πάνω σε συναρτήσεις, ιδιοσυναρτήσεις. Η δράση του τελεστή πάνω στις ιδιοσυναρτήσεις αλλάζει ουσιαστικά μόνο το μετρο τους πολλαπλασιάζοντας το με μία τιμή  $\lambda$  η οποία ονομάζεται ιδιοτιμή της ιδιοσυνάρτησης. Αυτή η ιδιαίτερη συμπεριφορά μπορεί να εκφραστεί και μέσω της εξίσωσης των ιδιοτιμών:

$$K\phi(x_k) = \lambda\phi(x_k) \quad (2.14)$$

για διακριτό σύστημα και:

$$L\phi(x) = \mu\phi(x) \quad (2.15)$$

για συνεχές σύστημα. Μάλιστα αν λάβουμε υπόψη την εξίσωση (2.12) μπορούμε να προχωρήσουμε στην εξής συνεπαγωγή:

$$\nabla\phi(x) \cdot f(x) = \lambda\phi(x) \quad (2.16)$$

Η παραπάνω εξίσωση έχει ευμεγέθη βαρύτητα καθώς συσχετίζει τα ιδιοδιανύσματα  $\phi$  με το διανυσματικό πεδίο  $f$ . Επομένως, ανάγουμε το πρόβλημα της εύρεσης των ιδιοσυναρτήσεων, στην επίλυση μίας διαφορικής εξίσωσης. Βέβαια, στις περισσότερες περιπτώσεις η επίλυση της εξίσωσης (2.16) με καθαρά και μόνο μαθηματικά εργαλεία, αποδεικνύεται από πολύ μεγάλη πρόκληση έως και άλυτος γρίφος. Για αυτό τον λόγο, καταφεύγουμε πολύ συχνά σε μεθόδους επεξεργασίας δεδομένων. Παρόλα αυτά, εάν το διανυσματικό πεδίο  $f$  έχει κάποια βολική μορφή, τότε η εξίσωση (2.16) μπορεί να λυθεί και αναλυτικά. Στο παρόν έργο ευτυχήσαμε να έχουμε μία κατάλληλη συνάρτηση δυναμικού από την οποία θα προκύψουν αναλυτικές εκφράσεις για τις ιδιοσυναρτήσεις.

## 2.5 Αναπαράσταση στον χώρο των ιδιοσυναρτήσεων

Μια ισχυρή ιδιότητα των ιδιοσυναρτήσεων Koopman, είναι η αναπαράσταση του διανύσματος κατάστασης και συναρτήσεων του σε ένα διανυσματικό χώρο του οποίου η βάση αποτελείται από τις κύριες ιδιοσυναρτήσεις. Άρα κάθε συνάρτηση του διανύσματος κατάστασης μπορεί να αναπαρασταθεί ως γραμμική υπέρθεση των ιδιοσυναρτήσεων, γεγονός το οποίο προσθέτει εναλλακτικούς τρόπους προσέγγισης των αυτόνομων μη γραμμικών δυναμικών συστημάτων της μορφής (2.4). Για παράδειγμα, εάν θέλουμε να υπολογίσουμε μία τροχιά του συστήματος από μία αρχική κατάσταση  $x_0$ , τότε αντί να εξετάσουμε απευθείας το σύστημα (2.4), έχουμε την δυνατότητα να εξετάσουμε την τροχιά στον χώρο των ιδιοσυναρτήσεων και μέσω της ιδιότητας της γραμμικής υπέρθεσης, να καταλήξουμε στο ίδιο αποτέλεσμα. Φυσικά όπως καταλαβαίνει κανείς, στο παραπάνω παράδειγμα παρακολούθησης της κίνησης του διανύσματος κατάστασης, η μεταφορά από τον αρχικό χώρο κατάστασης στον χώρο των ιδιοσυναρτήσεων, μάλλον θα μπορούσε να χαρακτηριστεί ως χάσιμο χρόνου καθώς θα μπορούσαμε να παρακολουθήσουμε απευθείας την τροχιά του από την εξίσωση (2.4). Η σπουδαιότητα του μετασχηματισμού στον χώρο των ιδιοσυναρτήσεων γίνεται ξεκάθαρη όταν εισάγουμε έλεγχο στο σύστημα. Ο νέος χώρος των ιδιοσυναρτήσεων

αποτελεί πρόσφορο έδαφος για την εφαρμογή ενός νόμου ελέγχου καθώς παρουσιάζει γραμμική συμπεριφορά ως προς τις ιδιοσυναρτήσεις και έτσι μπορούμε να αξιοποιήσουμε κλασσικές μεθόδους ελέγχου γραμμικών συστημάτων. Φυσικά, για να πάρουμε το επιθυμητό τελικό αποτέλεσμα εφαρμόζουμε τον αντίστροφο μετασχηματισμό στον αρχικό χώρο κατάστασης. Ας δούμε όμως πιο συγκεκριμένα πως συνδέεται το διάνυσμα κατάστασης με τις ιδιοσυναρτήσεις. Μία συνάρτηση του διανύσματος κατάστασης  $x$  μπορεί να επεκταθεί ως:

$$g(x) = \sum_{j=1}^{\infty} v_j \phi_j(x) \quad (2.17)$$

όπου  $v$  οι συντελεστές Koopman (modes), οι οποίοι απεικονίζουν το διάνυσμα κατάστασης στο χώρο των ιδιοσυναρτήσεων  $\phi$ . Μάλιστα αν επιθυμούμε να εξετάσουμε την τροχιά του συστήματος από ένα αρχικό σημείο  $x_0$ , εφαρμόζουμε τον τελεστή Koopman πάνω στην συνάρτηση  $g(x)$ . Λαμβάνοντας υπόψιν την εξίσωση (2.14) έχουμε:

$$g(x(t)) = Kg(x_0) = K \sum_{j=1}^{\infty} \phi_j(x_0) v_j = \sum_{j=1}^{\infty} K \phi_j(x_0) v_j = \sum_{j=1}^{\infty} \lambda_j \phi_j(x_0) v_j \quad (2.18)$$

Η τελική εξίσωση ισοδυναμεί με την δράση του τελεστή  $\mathbb{F}$  στο σύστημα (2.4). Επομένως η εξίσωση (2.18) μας παρέχει ακόμα έναν εναλλακτικό τρόπο για να παρακολουθήσουμε τις τροχιές του συστήματος και οι συντελεστές Koopman επιτρέπουν την μετάβαση από τον ένα χώρο κατάστασης στον άλλο.

## 2.6 Προσέγγιση του τελεστή Koopman με αριθμητικές μεθόδους

Στο κύριο αντικείμενο αυτής της πτυχιακής, δηλαδή στην πλοήγηση αυτόνομου ρομπότ, οι ιδιοσυναρτήσεις Koopman του δυναμικού συστήματος που περιγράφει το πρόβλημα μας, μπορούν να υπολογιστούν με καθαρά μαθηματικά εργαλεία. Παρόλα αυτά, στην συντριπτική πλειοψηφία των δυναμικών συστημάτων η έκφραση των ιδιοσυναρτήσεων σε αναλυτική μορφή καθίσταται αδύνατη. Αντ' αυτού γίνεται χρήση διαφόρων αριθμητικών μεθόδων με στόχο την καλύτερη δυνατή προσέγγιση των ιδιοσυναρτήσεων. Για λόγους πληρότητας της διπλωματικής, θα παρουσιάσουμε τις μεθόδους EDMD και DMD.

### **DMD και EDMD :**

Ο αλγόριθμος DMD (Dynamic Mode Decomposition) είναι ένας από τους πρωταρχικούς τρόπους υπολογισμού του τελεστή Koopman. Αναπτύχθηκε το 2008 από τους ερευνητές Peter J. Schmid και Joern Sesterhenn [12]. Ο PJ Schmid αργότερα δημοσίευσε και ένα ερευνητικό κείμενο με εφαρμογές του αλγορίθμου σε μη γραμμικά συστήματα [13]. Το κύριο χαρακτηριστικό του, είναι ότι δέχεται ως είσοδο μία βάση

δεδομένων, η οποία συνήθως προκύπτει από πειραματική προσομοίωση του δυναμικού συστήματος. Με άλλα λόγια ο αλγόριθμος δέχεται ένα σύνολο με στιγμιότυπα του δυναμικού συστήματος  $(x_k, y_k)_{(k=1)}^M$  έτσι ώστε να ισχύει  $S(x_k) = y_k$ , όπου  $S$  ο τελεστής εξέλιξης όπως ορίστηκε στην ενότητα 2.2. Στη συνέχεια ορίζουμε μία βιβλιοθήκη συναρτήσεων ως:

$$P_x = \begin{bmatrix} \psi_1(x_1) & \psi_2(x_1) & \dots & \psi_N(x_1) \\ \psi_1(x_2) & \psi_2(x_2) & \dots & \psi_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(x_k) & \psi_2(x_k) & \dots & \psi_N(x_k) \end{bmatrix}, P_y = \begin{bmatrix} \psi_1(y_1) & \psi_2(y_1) & \dots & \psi_N(y_1) \\ \psi_1(y_2) & \psi_2(y_2) & \dots & \psi_N(y_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(y_k) & \psi_2(y_k) & \dots & \psi_N(y_k) \end{bmatrix} \quad (2.19)$$

όπου στην μέθοδο DMD θεωρούμε  $\psi_j(x) = x_j$  (η ουσιώδης διαφορά με την μέθοδο EDMD, στην οποία έχουμε την δυνατότητα να επιλέξουμε διαφορετικές βιβλιοθήκες  $P_x, P_y$ ). Η προσέγγιση της μήτρας Koopman υπολογίζεται μέσω της εξίσωσης:

$$K = G^+ A \quad (2.20)$$

, όπου  $G^+$  είναι ο ψευδοαντίστροφος πίνακας του  $G$  και:

$$G = \frac{1}{M} \sum_{m=1}^M \Psi(x_m)^* \Psi(x_m) \quad (2.21)$$

$$A = \frac{1}{M} \sum_{m=1}^M \Psi(x_m)^* \Psi(y_m) \quad (2.22)$$

, όπου  $G, A, K \in \mathbb{C}^{\mathbb{K} \times \mathbb{K}}$ . Μετά τον υπολογισμό της μήτρας  $K$  μπορούμε να εξαγάγουμε τις ιδιοσυναρτήσεις Koopman μέσω της σχέσης:

$$\phi_j = \Psi \xi_j \quad (2.23)$$

, όπου  $\Psi$  είναι η βιβλιοθήκη που έχουμε ορίσει και  $\xi_j$  είναι το  $j$ -ιστό ιδιοδιάνυσμα της  $K$ .

Ο αλγόριθμος EDMD (Extended Dynamic Mode Decomposition) [14],[16],[11],[12] είναι στην ουσία μία βελτίωση του αλγόριθμου DMD καθώς η επιλογή της βιβλιοθήκης  $\Psi$  είναι ελεύθερη. Η επιλογή της  $\Psi$  για να θεωρείται πετυχημένη πρέπει να προσεγγίζει όσο περισσότερο δυνατόν την βάση των ιδιοσυναρτήσεων Koopman. Φυσικά, αυτή η επιλογή δεν είναι τόσο προφανής, καθώς δεν γνωρίζουμε εξάρχης την βάση. Η μορφή της συνάρτησης διανυσματικού πεδίου συνήθως μπορεί να οδηγήσει σε μία στοχευμένη επιλογή. Επιπλέον, υπάρχουν διάφορες συναρτήσεις, όπως οι RBFs (Radial Basis Functions) [15] και τα Hermite πολυώνυμα, οι οποίες προσφέρουν ικανοποιητική προσέγγιση ανεξαρτήτως της δυναμικής του συστήματος. Πάντως η ιδανική επιλογή της  $\Psi$  εξακολουθεί να είναι σοβαρό αντικείμενο έρευνας. Στο παράρτημα I παρουσιάζεται μία εφαρμογή της μεθόδου EDMD σε ένα μη γραμμικό δυναμικό σύστημα, όπως αναλύεται στο [16], για σκοπούς πληρότητας της πτυχιακής.

## Κεφάλαιο 3

# Εφαρμογή μεθόδου Koopman για την κατασκευή συναρτήσεων πλοήγησης

### 3.1 Εισαγωγή

Στο δεύτερο κεφάλαιο παρουσιάσαμε το πρόβλημα της αυτόνομης ρομποτικής πλήγησης και διάφορες μεθόδους αντιμετώπισης του. Μιλήσαμε για την σημασία των αρμονικών συναρτήσεων και ροών. Αναλύσαμε την απλουστευμένη εκδοχή των σημειακών εμποδίων καθώς και διάφορες τεχνικές μετασχηματισμών [6], [8] στον σημειακό χώρο. Σε αυτό το κεφάλαιο θα επεκταθούμε ακόμα περισσότερο και θα προτείνουμε μία διαφορετική προσέγγιση στο πρόβλημα η οποία μπορεί να εφαρμοστεί και σε ένα καρτεσιανό χώρο με δισδιάστατα εμπόδια. Με την χρήση του **τελεστή Koopman** και των ιδιοτήτων που παρουσιάστηκαν στο 3ο κεφάλαιο, καθώς και με την βοήθεια των **panel methods**, θα δημιουργήσουμε δυναμικές ροές που προσεγγίζουν τις αρμονικές ροές του 2ου κεφαλαίου, έτσι ώστε να εξασφαλίζεται η αποφυγή οποιασδήποτε σύγκρουσης καθώς και η άφιξη του ρομπότ στο σημείο προορισμού. Εφόσον ολοκληρώσουμε την θεωρητική ανάλυση θα παρουσιάσουμε απλές και περίπλοκες εφαρμογές, όπου υλοποιώντας όλα τα παραπάνω έχουμε καταλήξει σε επιθυμητά αποτελέσματα.

### 3.2 Περιβάλλον εργασίας και ρομποτικό μοντέλο

Ας ξεκινήσουμε με τον μαθηματικό ορισμό του χώρου κίνησης του αυτόνομου ρομποτικού συστήματος καθώς και με το ρομποτικό μοντέλο ελέγχου. Θεωρούμε έναν συμπαγές και συνεκτικό χώρο εργασίας  $W \subseteq \mathbb{R}^2$  με ένα εξωτερικό σύνορο  $\mathbb{O}_0$ . Μέσα στο εσωτερικό του χώρου  $W$  είναι τοποθετημένα  $M$  συμπαγή και στατικά εμπόδια τα οποία έχουν σύνορα επιφάνειας τα οποία συμβολίζονται με  $\mathbb{O}_i$  όπου  $i = 1, 2, \dots, M$ .

Σε αυτό το σημείο να αναφέρουμε ότι θεωρούμε πως το ρομπότ έχει επίγνωση των θέσεων των εμποδίων προτού ξεκινήσει η πλοήγηση. Ο επιθυμητός προορισμός ορίζεται ως ένα σημείο το οποίο ανήκει στο σύνολο  $\mathbb{W}$ , εξαιρουμένων φυσικά των σημείων που ανήκουν στα σύνολα  $\mathbb{O}_i$  καθώς και στο εσωτερικό των εμποδίων. Το σημείο εκκίνησης, όπως επεξηγήθηκε στο κεφάλαιο 2, δεν είναι προκαθορισμένο. Οι τροχιές της κίνησης του ρομπότ θα πρέπει να καταλήγουν στο σημείο προορισμού χωρίς να έρχονται σε επαφή με τα εμπόδια, ανεξαρτήτως των αρχικών συνθηκών. Ας αναφερθούμε τώρα στο ρομποτικό μοντέλο. Για απλούστευση του προβλήματος, θα θεωρήσουμε ότι το αυτόνομο ρομπότ ορίζεται και κινείται ως ένα σημείο  $x \in \mathbb{W}$ . Αυτή η υπόθεση απλοποιεί σε ένα βαθμό το πρόβλημα της πλοήγησης αλλά ταυτόχρονα δεν μας απομακρύνει ολοσχερώς από την πραγματικότητα. Υπάρχουν τετριμμένοι τρόποι μετασχηματισμού των νόμων πλοήγησης για σημειακό ρομπότ, σε αντίστοιχους νόμους για δισδιάστατα ή τρισδιάστατα αυτόνομα ρομπότ. Φυσικά διατηρώντας όλες τις επιθυμητές ιδιότητες οι οποίες εξασφαλίζουν σθεναρά αποτελέσματα. Θεωρούμε λοιπόν το εξής κινηματικό μοντέλο ελέγχου:

$$\dot{x} = u(t) \quad (3.1)$$

όπου  $x \in \mathbb{R}^2$  είναι το διάνυσμα θέσης του ρομπότ και  $u(t) \in \mathbb{R}^2$  είναι ο συνεχής νόμος ελέγχου που καλούμαστε να σχεδιάσουμε. Για έναν επιτυχημένο ελεγκτή θα πρέπει να ληφθούν υπόψη οι θέσεις των εμποδίων καθώς και η θέση του σημείου προορισμού. Το ρομπότ κινείται αυτόνομα και δεν μπορούμε να παρέμβουμε κατά την εκτέλεση της κίνησης του.

### 3.3 Δυναμικές ροές

Μέχρι αυτό το σημείο της διπλωματικής, έχουμε παρουσιάσει κάποιες βασικές αρχές και μεθόδους της αυτόνομης ρομποτικής πλοήγησης, καθώς και ένα ολοκληρωμένο μαθηματικό υπόβαθρο για τον τελεστή Koopman. Η παρούσα ενότητα, όπως και οι επόμενες ενότητες της διπλωματικής, αποτελούν την καινοτόμο προσφορά του συγγραφέα και για αυτόν τον λόγο είναι εξαιρετικής σημασίας. Αρχικά, ας αναφέρουμε συνοπτικά τις συνθήκες του προβλήματος πλοήγησης, οι οποίες έχουν μία ειδοποιό διαφορά σε σχέση με την απλουστευμένη εκδοχή του δεύτερου κεφαλαίου. Καλούμαστε να σχεδιάσουμε έναν νόμο πλοήγησης για ένα αυτόνομο σημειακό ρομπότ, ο οποίος θα το καθοδηγήσει από ένα τυχαίο σημείο εκκίνησης σε ένα σημείο προορισμού αποφεύγοντας παράλληλα όλα τα **εμπόδια φυσικών διαστάσεων** στο χώρο. Η τελευταία συνθήκη, δηλαδή η ύπαρξη εμποδίων με φυσικές διαστάσεις, διαφοροποιεί εντελώς την φύση του προβλήματος που παρουσιάστηκε στο 2ο κεφάλαιο. Τότε, είχαμε αναφερθεί στην υπέρθεση αρμονικών συναρτήσεων ως μέσο υλοποίησης του νόμου ελέγχου. Κάθε αρμονική συνάρτηση δημιουργούσε μία δυναμική ροή η οποία εξασφάλιζε ότι οι τροχιές του ρομποτικού συστήματος δεν θα διαπερνούσαν ποτέ τα εμπόδια, αποκλείοντας έτσι οποιαδήποτε πιθανότητα σύγκρουσης. Η τελική συνάρτηση ελέγχου η οποία ήταν αποτέλεσμα της υπέρθεσης των αρμονικών συναρτήσεων, παρήγαγε την ολική δυναμική ροή η οποία ικανοποιούσε όλες τις απαιτήσεις

του προβλήματος και καθοδηγούσε το ρομπότ στο σημείο προορισμού. Στο 2ο κεφάλαιο όμως είχαμε θεωρήσει ότι τα εμπόδια αναπαρίστανται από μεμονωμένα σημεία στον καρτεσιανό χώρο. Αυτή η υπόθεση, διευκόλυνε ιδιαίτερα την κατασκευή των κατάλληλων δυναμικών ροών καθώς η επιλογή των αρμονικών συναρτήσεων ήταν πιο προφανής. Συγκεκριμένα, οι δυναμικές ροές γύρω από τα εμπόδια προκύπταν από τις συναρτήσεις:

$$\phi_i(h) = \ln(\|h - P_i\|^2) \quad (3.2)$$

όπου  $P_i$  το  $i$ -σημειακό εμπόδιο και  $h$  η απόσταση από αυτό. Η συγκεκριμένη συνάρτηση εξασφαλίζει την αρμονικότητα σε όλο τον καρτεσιανό χώρο και παράλληλα στο σημείο  $P_i$  ισχύει:

$$\lim_{h \rightarrow P_i} \phi_i(h) = -\infty \quad (3.3)$$

Η τελευταία συνοριακή συνθήκη είναι αναγκαία καθώς δημιουργεί την επιθυμητή απωθητική δύναμη από τα εμπόδια και εξασφαλίζει ότι οι δυναμικές ροές δε θα τα διαπερνούν.

Ας προχωρήσουμε τώρα στην εκδοχή των εμποδίων με φυσικές διαστάσεις. Η λογική που ακολουθήθηκε στο 2ο κεφάλαιο δεν θα εγκαταλειφθεί, αντιθέτως θα θέσουμε τις ίδιες προαπαιτήσεις για τις συναρτήσεις των εμποδίων. Δηλαδή ψάχνουμε μία συνάρτηση  $\phi$  έτσι ώστε να ικανοποιεί την εξίσωση Laplace :

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \quad (3.4)$$

και επιπρόσθετα για κάποιο σημείο  $r \in \partial\mathbb{O}_i$  του εμποδίου  $P_i$ , όπου  $\partial\mathbb{O}_i$  το σύνορο του εμποδίου, πρέπει να ισχύει:

$$\lim_{h \rightarrow r} \phi_i(h) = -\infty \quad (3.5)$$

Η τελευταία συνθήκη πλέον δεν αναφέρεται σε ένα σημείο αλλά σε ένα σύνολο σημείων στο σύνορο του εμποδίου. Γεγονός το οποίο δυσκολεύει κατά σημαντικό βαθμό την επιλογή κατάλληλης συνάρτησης που ικανοποιεί τις δύο απαιτήσεις. Για παράδειγμα η συνάρτηση (3.2), διατηρεί την αρμονικότητα σε όλο το καρτεσιανό επίπεδο αλλά όσον αφορά την δεύτερη συνθήκη δεν είναι επαρκής. Είναι προφανές ότι στο σημείο  $P_i$  η συνάρτηση θα απειρίζεται, αλλά κατά μήκος του συνόρου η ιδιότητα δε θα διατηρείται. Μάλιστα η εύρεση μίας αρμονικής συνάρτησης που να ικανοποιεί τα παραπάνω δεν είναι διόλου εύκολη. Συνήθως όταν η συνάρτηση εξασφαλίζει την μία ιδιότητα, αποτυγχάνει να ικανοποιήσει την άλλη. Για αυτό τον λόγο, θα προσπαθήσουμε να **κατασκευάσουμε απευθείας τις επιθυμητές δυναμικές ροές**, χωρίς να υπολογίσουμε τις αρμονικές συναρτήσεις που τις παράγουν. Σε αυτήν την περίπτωση θα χρειαστεί να αναπροσαρμόσουμε τις συνθήκες (3.4), (3.5), έτσι ώστε να αφορούν δυναμικές ροές. Οι καινούριες συνθήκες για μια ροή  $\vec{V}$  θα είναι:

$$\nabla \vec{V} = 0 \quad (3.6)$$

καθώς ισχύει  $\nabla\phi = \vec{V}$  και :

$$\lim_{h \rightarrow r} V_x = \pm\infty, \quad \lim_{h \rightarrow r} V_y = \pm\infty \quad (3.7)$$

όπου  $r \in \partial\mathbb{O}_i$  (σύνορο του εμποδίου  $P_i$ ) και το πρόσημο του απείρου εξαρτάται από την πλευρά που προσεγγίζουμε το σημείο.

Εφόσον μας ενδιαφέρει η σχεδίαση μίας κατάλληλης δυναμικής ροής, ας εξετάσουμε την ροή που παράγει η αρμονική συνάρτηση (3.2):

$$\vec{V} = \nabla\phi = \left[ \begin{array}{c} \frac{x-x_i}{(x-x_i)^2+(y-y_i)^2} \\ \frac{y-y_i}{(x-x_i)^2+(y-y_i)^2} \end{array} \right] \quad (3.8)$$

Όπως έχουμε αναφέρει, αυτή η μορφή δυναμικού αντιστοιχεί σε σημειακό εμπόδιο και αποτυγχάνει να ικανοποιήσει την συνοριακή συνθήκη (3.5). Παρόλα αυτά, η συνάρτηση αυτή δεν μας είναι αδιάφορη. Με κατάλληλους χειρισμούς μπορούμε να αξιοποιήσουμε την καθολική αρμονικότητα της  $\vec{V}$  και προσεγγιστικά να καταλήξουμε σε μία δυναμική ροή που να ικανοποιεί σε αρκετά μεγάλο βαθμό την συνθήκη (3.5). Σε αυτό το σημείο λοιπόν, ήρθε η ώρα να συνδέσουμε την θεωρία του **τελεστή Koopman** που παρουσιάστηκε στο 3ο κεφάλαιο, με το πρόβλημα της πλοήγησης. Ορίζουμε λοιπόν το εξής μη γραμμικό δυναμικό σύστημα:

$$\left[ \begin{array}{c} \dot{x} \\ \dot{y} \end{array} \right] = \left[ \begin{array}{c} \frac{x-x_i}{(x-x_i)^2+(y-y_i)^2} \\ \frac{y-y_i}{(x-x_i)^2+(y-y_i)^2} \end{array} \right] \quad (3.9)$$

όπου  $x, y \in \mathbb{R}$ . Διαισθητικά το σύστημα (3.9) περιγράφει την κίνηση μίας ελεύθερης μάζας μέσα στο δυναμικό πεδίο  $\vec{V}$ . Με τη βοήθεια του τελεστή θέλουμε να υπολογίσουμε τις ιδιοσυναρτήσεις Koopman που συσχετίζονται με το δυναμικό (3.8). Τότε επιλέγοντας ορισμένες από αυτές, και με τη μέθοδο της υπέρθεσης, θα προσπαθήσουμε να δημιουργήσουμε μία συνάρτηση ροής η οποία προσεγγιστικά θα ικανοποιεί τις δύο συνθήκες (3.6),(3.7).

## 3.4 Ιδιοσυναρτήσεις του δυναμικού συστήματος πλοήγησης

Ας εφαρμόσουμε τώρα την μεθοδολογία Koopman στο δυναμικό σύστημα (3.9). Όπως αναφέραμε στο 3ο κεφάλαιο, ο υπολογισμός ιδιοτιμών και ιδιοσυναρτήσεων συνήθως πραγματοποιείται μέσω κάποιων αριθμητικών αλγορίθμων όπως DMD, EDMD . Σε αυτήν την πτυχιακή θα αποκλίνουμε από αυτήν την προσέγγιση, καθώς οι παραπάνω μέθοδοι υπολογίζουν τις τιμές των ιδιοτιμών και ιδιοσυναρτήσεων για κάποια συγκεκριμένη αρχική συνθήκη του διανύσματος κατάστασης  $\begin{bmatrix} x \\ y \end{bmatrix}$ . Στις προδιαγραφές του νόμου ελέγχου όμως, απαιτείται η καθοδήγηση του ρομποτικού συστήματος



από οποιοδήποτε σημείο εκκίνησης στο σημείο προορισμού. Επομένως η καλή λειτουργία του νόμου ελέγχου για μία μεμονωμένη τροχιά δεν είναι επαρκές κριτήριο για την επιτυχία της σχεδίασης μας. Για να μπορέσουμε να αξιοποιήσουμε τους αριθμητικούς αλγόριθμους, πρέπει να εξετάζουμε για κάθε αρχική συνθήκη τις τιμές των ιδιοσυναρτήσεων. Με αυτόν τον τρόπο όμως, δε θα καταλήγαμε σε κάποιο σθεναρό αποτέλεσμα ενώ παράλληλα θα οδηγούμασταν σε μία εκτεταμένη σπατάλη πόρων.

Το ιδανικό σενάριο θα ήταν ο υπολογισμός των ιδιοσυναρτήσεων σε αναλυτική μορφή, δηλαδή συναρτήσεϊ του διανύσματος κατάστασης. Στις περισσότερες περιπτώσεις κάτι τέτοιο δεν είναι εφικτό, αλλά στο συγκεκριμένο μη-γραμμικό σύστημα, η συνάρτηση δυναμικού πεδίου αποδεικνύεται ιδιαίτερα ευνοϊκή. Αρχικά ας επανέλθουμε στην διαφορική εξίσωση ιδιοσυναρτήσεων Koopman (2.16) και ας εισάγουμε την δυναμική του συστήματος (3.9). Τότε θα έχουμε την διαφορική εξίσωση:

$$\frac{x - x_i}{(x - x_i)^2 + (y - y_i)^2} \cdot \frac{\partial \phi}{\partial x} + \frac{y - y_i}{(x - x_i)^2 + (y - y_i)^2} \cdot \frac{\partial \phi}{\partial y} = \lambda \phi \quad (3.10)$$

Με προσεκτική επισκόπηση της εξίσωσης συμπεραίνουμε ότι οι συναρτήσεις της μορφής:

$$\phi(x, y, x_i, y_i, k) = e^{k[(x-x_i)^2+(y-y_i)^2]} \quad (3.11)$$

όπου  $k$  ρητός αριθμός, είναι λύσεις της εξίσωσης (3.10) και ιδιοσυναρτήσεις του συστήματος (3.9). Διαισθητικά αυτό το αποτέλεσμα είναι πολύ ικανοποιητικό, καθώς πλέον έχουμε αναλυτική μορφή για όλες τις ιδιοσυναρτήσεις και παράλληλα αυτές, παράγουν έναν απειροδιάστατο χώρο (εφόσον υπάρχουν άπειρες στον αριθμό ιδιοσυναρτήσεις), γεγονός το οποίο συμβαδίζει με την θεωρία που παρουσιάστηκε στο 2ο κεφάλαιο. Σημαντικό να αναφέρουμε σε αυτό το σημείο ότι οι ιδιοσυναρτήσεις είναι συνεχείς σε όλο το  $\mathbb{R}^2$  και δεν εμφανίζουν κανένα ανώμαλο σημείο, σε αντίθεση με την ροή (3.8). Το χαρακτηριστικό αυτό, μας επιτρέπει να σχεδιάσουμε πιο εύελικτες δυναμικές ροές έτσι ώστε να ικανοποιήσουμε σε σημαντικό βαθμό τις κρίσιμες συνθήκες (3.6), (3.7). Φυσικά, οι ιδιοσυναρτήσεις από μόνες τους, δεν μπορούν να παρέχουν το επιθυμητό αποτέλεσμα. Εκ φύσεως περιγράφουν την δυναμική του συστήματος (3.9), το οποίο όπως έχουμε αναφέρει εξασφαλίζει την συνθήκη της αρμονικότητας (3.6), αλλά δεν εξασφαλίζει την συνθήκη (3.7) παρά μόνο στο σημείο  $(x_i, y_i)$ . Με κάποιο τρόπο επιθυμούμε να επεκτείνουμε το σημείο  $(x_i, y_i)$  κατά μήκος του συνόρου του κάθε εμποδίου, έτσι ώστε να ικανοποιείται η συνθήκη (3.7) και παράλληλα να διατηρείται σε έναν σημαντικό βαθμό η αρμονικότητα. Για την επίτευξη όλων των παραπάνω, θα αξιοποιήσουμε μία μέθοδο η οποία είναι ιδιαίτερα διαδεδομένη στην επιστήμη της αεροδυναμικής, και ονομάζεται μέθοδος σήραγγας (panel methods).

## 3.5 Panel Method

Η μέθοδος σήραγγας (panel method) είναι μία υπολογιστική μέθοδος η οποία χρησιμοποιείται κυρίως στην επιστήμη της αεροδυναμικής για την επίλυση προβλημάτων που εμπλέκουν ροές γύρω από στερεά σώματα. Για παράδειγμα, συνηθίζεται

να χρησιμοποιείται για την σταθεροποίηση των πτερύγων των αεροσκαφών και άλλων αεροδυναμικών αντικειμένων. Στην ουσία η μέθοδος αυτή, υπολογίζει την ροή (δυναμική ροή στην αεροδυναμική ή ροή ρευστού στην ρευστοδυναμική) γύρω από ένα αντικείμενο. Στη δική μας περίπτωση, θα αξιοποιήσουμε αυτήν την μέθοδο για να δημιουργήσουμε την επιθυμητή ροή γύρω από τα εμπόδια η οποία θα εξασφαλίζει τις συνθήκες (3.6) και (3.7).

Ας αναλύσουμε λοιπόν με μεγαλύτερη λεπτομέρεια την διαδικασία υπολογισμού της ροής με την χρήση της μεθόδου. Η βασική ιδέα, είναι η **διακριτοποίηση** της επιφάνειας του αντικειμένου σε πολλά μικροσκοπικά τμήματα, τα οποία προσεγγιστικά μπορούν να θεωρηθούν ως ευθύγραμμα. Στη συνέχεια προσπαθούμε να κατασκευάσουμε την επιθυμητή δυναμική ροή γύρω από ένα μεμονωμένο τμήμα και με την ιδιότητα της υπέρθεσης θα συνδυάσουμε όλες τις επιμέρους δυναμικές ροές των τμημάτων για να υπολογίσουμε την ολική δυναμική ροή. Είναι σημαντικό να αναφέρουμε, ότι εάν οι δυναμικές ροές των τμημάτων δεν ικανοποιούν την εξίσωση (3.6) έστω και προσεγγιστικά, η υπέρθεση τους δεν θα φέρει σωστό αποτέλεσμα.

Ας επικεντρωθούμε τώρα στην επίλυση του προβλήματος σε ένα μεμονωμένο τμήμα. Ένα ικανοποιητικό αποτέλεσμα σε ένα 'πάνελ' είναι αρκετό για να μας εξασφαλίσει ότι και το ολικό αποτέλεσμα θα είναι το επιθυμητό. Στην επιστήμη της αεροδυναμικής, συνηθίζεται να επιλέγεται ένα σημείο πηγής  $P_d(x_d, y_d)$ , το οποίο εκπέμπει την αρμονική συνάρτηση  $\phi(x, y, P_d)$ . Η συνολική συνάρτηση δυναμικού  $\Phi(x, y)$  του τμήματος υπολογίζεται ως:

$$\Phi(x, y) = \frac{1}{2\pi} \int \lambda * \phi(x, y, x_d, y_d) dS \quad (3.12)$$

όπου το  $\lambda$  είναι η κατανομή δύναμης πάνω στο τμήμα. Η δυναμική ροή του τμήματος υπολογίζεται όπως είναι γνωστό από την βαθμίδα της συνάρτησης δυναμικού  $\vec{V}_p = \nabla\Phi$ . Στο δικό μας πρόβλημα τη συνάρτηση  $\phi(x, y, x_d, y_d)$  θα αποτελεί έναν γραμμικό συνδυασμό των ιδιοσυναρτήσεων Koozman που υπολογίσαμε στην ενότητα 3.4. Αυτές οι ιδιοσυναρτήσεις όμως, διατηρούν την δυναμική του συστήματος (3.9), δηλαδή της δυναμικής ροής που παράγεται από την βαθμωτή συνάρτηση (3.2). Επομένως, δεν θα θεωρήσουμε κάποια βαθμωτή συνάρτηση, αντάυτου θα κινηθούμε πλήρως στον 'κόσμο των δυναμικών ροών'. Η παραλλαγή της (3.12) την οποία και θα αξιοποιήσουμε θα υπολογίζει την συνολική δυναμική ροή του τμήματος  $\vec{V}_p = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$  και θα έχει την μορφή:

$$V_x(x, y) = \iint \lambda_x(x, x_d) * v(x, y, x_d, y_d) dx_d dy_d \quad (3.13)$$

$$V_y(x, y) = \iint \lambda_y(y, y_d) * v(x, y, x_d, y_d) dx_d dy_d \quad (3.14)$$

Επειδή όμως ακολουθούμε αυτήν την διαδικασία σε μία μεμονωμένη λωρίδα (ή αλλιώς τμήμα), μπορούμε να θεωρήσουμε μία παραμετροποίηση της μορφής  $x_d = x_d(t), y_d = y_d(t)$  και να ανάγουμε τις δύο εξισώσεις σε δύο ισοδύναμες:

$$V_x(x, y) = \int_C \lambda_x(x, t) * v(x, y, t) dt \quad (3.15)$$

$$V_y(x, y) = \int_C \lambda_y(y, t) * v(x, y, t) dt \quad (3.16)$$

Οι συναρτήσεις  $\lambda_x$  και  $\lambda_y$  θα επιλεγθούν κατάλληλα έτσι ώστε η τελική δυναμική ροή  $\vec{V}_p$  να εμφανίζει τις επιθυμητές ιδιότητες. Συνήθως χρησιμοποιούμε τις συναρτήσεις  $\lambda_x = \text{sign}(x - x_d)$  και  $\lambda_x = \sin(x - x_d)$ , όπου  $\text{sign}$  είναι η γνωστή συνάρτηση προσήμου, και τις αντίστοιχες για το  $\lambda_y$ . Οι συναρτήσεις αυτές όπως θα φανεί στην ενότητα των εφαρμογών, συμβάλουν στην δημιουργία ροών  $V_x, V_y$  οι οποίες εμφανίζουν σε μεγάλο βαθμό τις δύο κρίσιμες ιδιότητες (3.6), (3.7). Όσον αφορά την συνάρτηση δυναμικής ροής  $v$  αυτή θα προκύψει ως γραμμικός συνδυασμός των ιδιοσυναρτήσεων:

$$v(x, y, x_d, y_d) = \sum_{n=-k}^0 c_n * \phi(x, y, x_d, y_d, n) \quad (3.17)$$

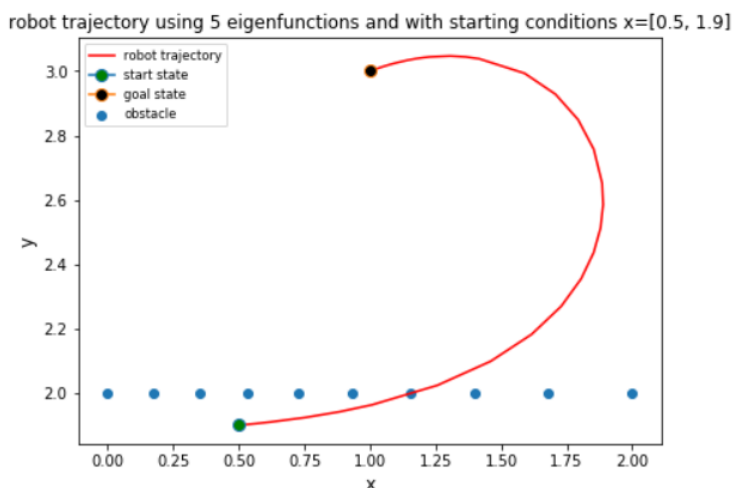
Οι συντελεστές  $c_n$  επιλέγονται με κατάλληλο τρόπο έτσι ώστε το διανυσματικό πεδίο  $\vec{V}$  να διατηρεί προσεγγιστικά αρμονικές ιδιότητες. Ουσιαστικά, επιχειρούμε να 'χαλαρώσουμε' την αυστηρότητα της συνθήκης (3.6) με σκοπό να ικανοποιήσουμε την συνθήκη (3.7) καθώς ρεαλιστικά, η αυστηρή τήρηση και των δύο αποτελεί μία εξαιρετικά δύσκολη αποστολή. Επιπρόσθετα επιλέγουμε  $k$  ιδιοσυναρτήσεις με αρνητικό εκθέτη. Αυτή η επιλογή βασίζεται στο γεγονός ότι μακριά από τα εμπόδια οι ιδιοσυναρτήσεις προσεγγίζουν το μηδέν, ιδιότητα η οποία εξασφαλίζει ότι το ρομπότ δεν θα επηρεάζεται από μία δυναμική ροή ενός απομακρυσμένου εμποδίου. Στο σύνορο του εμποδίου από την άλλη, δηλαδή για  $x = x_d, y = y_d$  οι ιδιοσυναρτήσεις λαμβάνουν την μέγιστη τιμή  $\phi(x_d, y_d) = 1$ . Φυσικά, είναι εύλογο να αναρωτηθεί κανείς, πως θα εξασφαλίσουμε τον απειρισμό της συνθήκης (3.7); Εκ φύσεως, η κάθε ιδιοσυνάρτηση δεν μπορεί να εξασφαλίσει το άπειρο πάνω στο σύνορο του εμποδίου, αλλά η υπέρθεση ενός πολυ μεγάλου αριθμού ιδιοσυναρτήσεων θα έχει ως αποτέλεσμα μία μεγάλη τιμή στο σύνορο, ότι ακριβώς επιζητούμε. Αντιθέτως, εάν επιλέξουμε έναν μικρό αριθμό ιδιοσυναρτήσεων  $k = \nu$  το άθροισμα (3.17) θα πάρει την τιμή  $\nu$  η οποία δεν είναι επαρκής για να μας εξασφαλίσει την εγκυρότητα της εξίσωσης (3.7). Επιπρόσθετα, η αξιοποίηση μεγάλου αριθμών ιδιοσυναρτήσεων ικανοποιεί την συνθήκη αρμονικότητας (3.6) σε απόλυτο βαθμό σε όλα τα σημεία του χώρου **εκτός από τα σημεία στο σύνορο του εμποδίου**, πάνω στο οποίο όπως εξηγήσαμε, είναι αδύνατο να πετύχουμε χωρίς απώλειες και τις δύο συνθήκες (3.6), (3.7). Το μόνο μειονέκτημα της χρήσης μεγάλου αριθμού ιδιοσυναρτήσεων σχετίζεται με την πολυπλοκότητα του αλγορίθμου και με το γεγονός ότι μετά απο κάποιο ορισμένο σημείο η περαιτέρω αύξηση αυτού του αριθμού δεν φαίνεται κάποια ουσιαστική διαφορά. Για να συνοψίσουμε, προσπαθούμε να κατασκευάσουμε την δυναμική ροή  $\vec{V} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$  με τέτοιο τρόπο έτσι ώστε να ικανοποιούνται οι συνθήκες (3.6), (3.7). Για την κατασκευή αυτή θα χρησιμοποιήσουμε τις εξισώσεις (3.15) και (3.16), επιλέγοντας τις κατάλληλες συναρτήσεις  $\lambda$  και υπολογίζοντας ταυτόχρονα την συνάρτηση  $v$  από την εξίσωση ιδιοσυναρτήσεων (3.17). Στην επόμενη ενότητα συγκρίνουμε τις διαφορές στην προσομοίωση ενός προβλήματος πλοήγησης, με την χρήση μικρού και μεγάλου αντίστοιχα αριθμού ιδιοσυναρτήσεων.

## 3.6 Προσομοίωση με διαφορετικό πλήθος ιδιοσυναρτήσεων

Ορίζουμε έναν καρτεσιανό χώρο με ένα στατικό εμπόδιο ευθύγραμμου τμήματος από το σημείο (0,2) μέχρι το σημείο (2,2). Ως αρχική συνθήκη επιλέγουμε το διάνυσμα κατάστασης [0.5, 1.9], το οποίο βρίσκεται αρκετά κοντά στα τοιχώματα του εμποδίου, και το διάνυσμα κατάστασης [1,3] για το επιθυμητό σημείο προορισμού.

### 3.6.1 Χρήση 5 ιδιοσυναρτήσεων

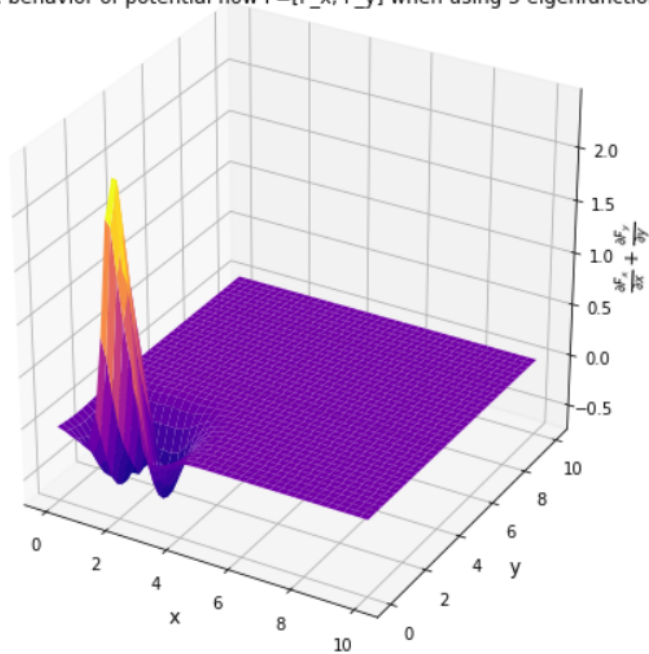
Αρχικά επιλέγουμε έναν μικρό αριθμό ιδιοσυναρτήσεων έτσι ώστε να εξετάσουμε εάν αυτός είναι επαρκής για την σωστή λειτουργία του νόμου ελέγχου. Το αποτέλεσμα καταγράφεται στο παρακάτω διάγραμμα:



Σχήμα 3.1: Αποτέλεσμα πλοήγησης με χρήση 5 ιδιοσυναρτήσεων.

Όπως είναι προφανές, το αποτέλεσμα δεν είναι το επιθυμητό. Η χρήση μόνο πέντε ιδιοσυναρτήσεων δεν είναι αρκετή για να ικανοποιήσει την συνθήκη (3.7) έστω και προσεγγιστικά. Για αυτό το λόγο, το ρομπότ δεν αναγνωρίζει το τοίχωμα του εμποδίου, καθώς δεν αισθάνεται μία μεγάλη αύξηση δυναμικού, και για αυτόν τον λόγο το διαπερνά. Επιπλέον, το τεχνητό δυναμικό που έχουμε κατασκευάσει με αυτές τις ιδιοσυναρτήσεις εμφανίζει κενά και όσον αφορά την συνθήκη (3.6). Στο παρακάτω διάγραμμα φαίνεται ότι η βαθμίδα του δυναμικού παίρνει μη μηδενικές τιμές σε αρκετά σημεία κοντά στο εμπόδιο.

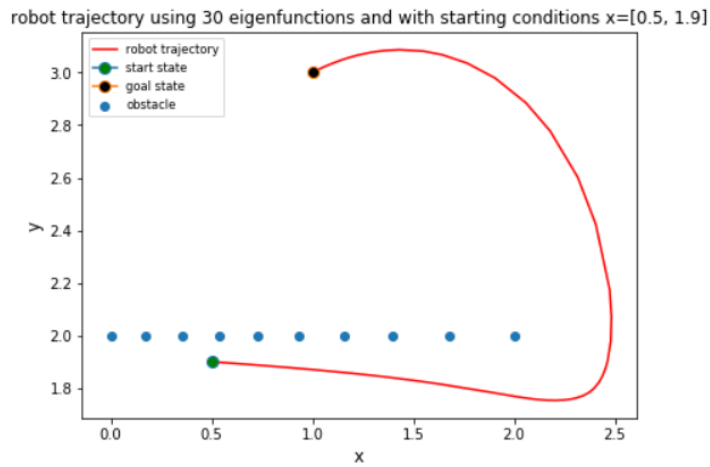
harmonic behavior of potential flow  $F=[F_x, F_y]$  when using 5 eigenfunctions



Σχήμα 3.2: Βαθμιά της δυναμικής ροής με χρήση 5 ιδιοσυναρτήσεων

### 3.6.2 Χρήση 30 ιδιοσυναρτήσεων

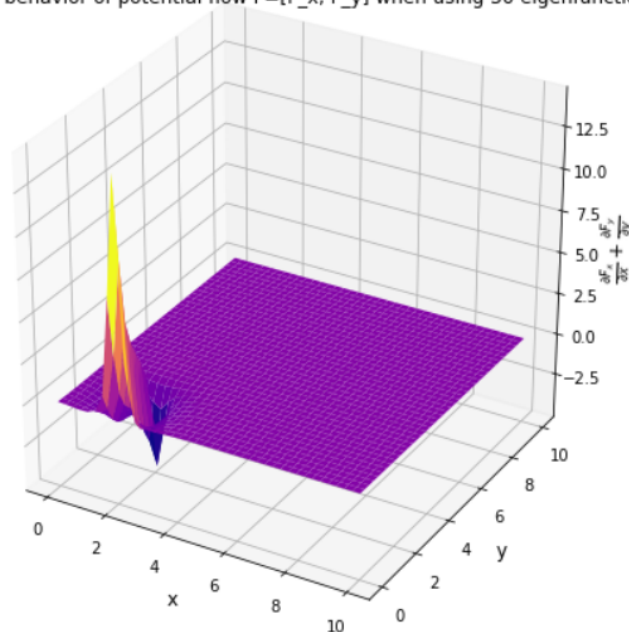
Τώρα θα αυξήσουμε σημαντικά τον αριθμό των ιδιοσυναρτήσεων έτσι ώστε η δυναμική ροή που θα σχεδιάσουμε να ικανοποιεί σε αρκετά μεγαλύτερο βαθμό τις συνθήκες (3.6),(3.7). Στο παρακάτω διάγραμμα φαίνεται το αποτέλεσμα της πλοήγησης με χρήση 30 ιδιοσυναρτήσεων.



Σχήμα 3.3: Αποτέλεσμα πλοήγησης με χρήση 30 ιδιοσυναρτήσεων

Όπως παρατηρούμε το αποτέλεσμα είναι αρκετά ικανοποιητικό καθώς το ρομπότ αποφεύγει το εμπόδιο, γεγονός το οποίο υποδεικνύει την τήρηση της συνοριακής συνθήκης (3.6), και καταλήγει στο σημείο προορισμού. Ας εξετάσουμε τώρα τον νόμο ελέγχου ως προς την αρμονική του συμπεριφορά.

harmonic behavior of potential flow  $F=[F_x, F_y]$  when using 30 eigenfunctions

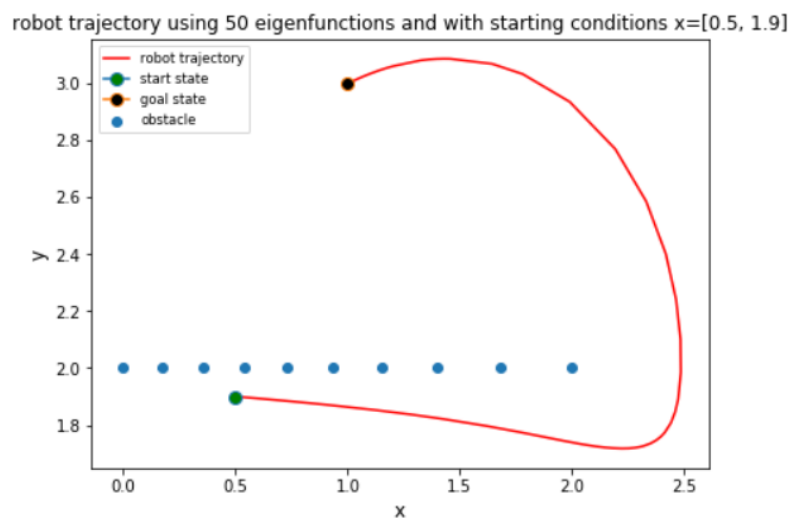


Σχήμα 3.4: Βαθμίδα της δυναμικής ροής με χρήση 30 ιδιοσυναρτήσεων

Όπως φαίνεται υπάρχει σημαντική βελτίωση (σε σχέση με την χρήση 5 ιδιοσυναρτήσεων) και όσον αφορά την συνθήκη (3.7). Η βαθμίδα της δυναμικής ροής μηδενίζεται σχεδόν σε όλα τα σημεία του καρτεσιανού χώρου, με μοναδικές εξαιρέσεις αυτά πάνω στα τοιχώματα του εμποδίου.

### 3.6.3 Χρήση 50 ιδιοσυναρτήσεων

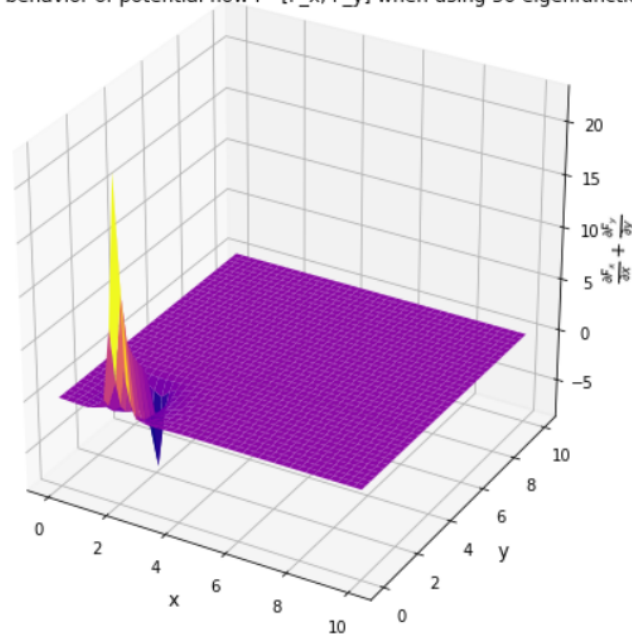
Αυξάνουμε περαιτέρω τον αριθμό των ιδιοσυναρτήσεων για να εξετάσουμε κατά πόσο αυτή η αύξηση θα βελτιώσει το αποτέλεσμα της πλοήγησης. Στο παρακάτω διάγραμμα φαίνεται το αποτέλεσμα της πλοήγησης με χρήση 50 ιδιοσυναρτήσεων.



Σχήμα 3.5: Αποτέλεσμα πλοήγησης με χρήση 50 ιδιοσυναρτήσεων

Όπως παρατηρούμε το αποτέλεσμα της πλοήγησης είναι παρόμοιο με εκείνο της προηγούμενης υποενοτήτας όπου είχαμε χρησιμοποιήσει 30 ιδιοσυναρτήσεις. Δηλαδή και σε αυτήν την περίπτωση η συνθήκη (3.7) ικανοποιείται σε πολύ μεγάλο βαθμό. Ας δούμε τώρα τι συμβαίνει με την συνθήκη αρμονικότητας (3.6).

harmonic behavior of potential flow  $F=[F_x, F_y]$  when using 50 eigenfunctions



Σχήμα 3.6: Βαθμίδα της δυναμικής ροής με χρήση 50 ιδιοσυναρτήσεων

Σε αυτό το διάγραμμα μπορούμε να διακρίνουμε μία διαφορά, έστω και μικρή, σε σχέση με την προηγούμενη εκδοχή των 30 ιδιοσυναρτήσεων. Η βαθμίδα της δυναμικής ροής μηδενίζεται σε όλα τα σημεία μακριά από το εμπόδιο, ενώ στα σημεία που βρίσκονται κοντά στα τοιχώματα του εμποδίου, παρουσιάζει μία πιο απότομη συμπεριφορά σε σχέση με τις δύο προηγούμενες εκδοχές. Αυτό σημαίνει ότι το σύνολο των σημείων του καρτεσιανού χώρου, στα οποία ικανοποιείται η εξίσωση (3.6), είναι ακόμα μεγαλύτερο. Επομένως σε αυτήν την περίπτωση, μπορούμε να σχεδιάσουμε έναν πιο στυβαρό νόμο ελέγχου που θα έχει πολύ μικρές πιθανότητες εμφάνισης τοπικών ελαχίστων στο χώρο, γεγονός το οποίο γίνεται ιδιαίτερα σημαντικό όταν προστεθούν και άλλα εμπόδια στον χώρο εργασίας. Βεβαίως όμως, η αύξηση των ιδιοσυναρτήσεων φέρνει και μία αύξηση στην πολυπλοκότητα του αλγορίθμου κάτι το οποίο πρέπει πάντα να λαμβάνουμε υπόψιν.



## Κεφάλαιο 4

# Εφαρμογές και αποτελέσματα

### 4.1 Εμπόδιο ευθύγραμμου τμήματος

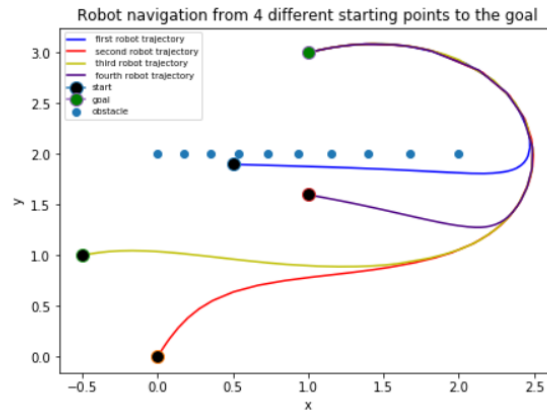
Σε αυτό το κομμάτι της διπλωματικής θα εφαρμόσουμε την θεωρία του προηγούμενου κεφαλαίου σε διάφορα προβλήματα πλοήγησης. Αρχικά ας εξετάσουμε ένα απλοϊκό περιβάλλον εργασίας με ένα ευθύγραμμο τμήμα ως εμπόδιο. Επιλέγουμε τις συναρτήσεις κατανομής έντασης ως  $\lambda_x(x) = \cos(x - x_d)$  και  $\lambda_y(y) = \sin(y - y_d)$ . Καταλήξαμε σε αυτήν την επιλογή, καθώς όπως αποδεικνύεται η ολική δυναμική ροή διατηρεί μία μεγάλη τιμή στο σύνορο του εμποδίου και ταυτόχρονα ικανοποιείται σε αρκετά μεγάλο βαθμό η συνθήκη αρμονικότητας. Επιπρόσθετα, επειδή οι συναρτήσεις έχουν πεδίο τιμών το  $[-1,1]$ , η προκύπτουσα δυναμική ροή  $V$  εμφανίζει παραπλήσιες τιμές πάνω στο τοίχωμα του εμποδίου. Στη συνέχεια προχωράμε στον υπολογισμό της ολικής δυναμικής ροής, η οποία σχετίζεται στα εμπόδια που βρίσκονται στο περιβάλλον εργασίας, μέσω των σχέσεων (3.13) και (3.14). Φυσικά χρειαζόμαστε και μία συνάρτηση 'βρύσης' (sink) στο σημείο προορισμού η οποία θα παράγει μία δυναμική ροή που θα ελκύει το ρομπότ. Εφόσον ο τελικός στόχος συμβολίζεται από ένα σημείο στο περιβάλλον εργασίας, θα αξιοποιήσουμε την θεωρία του 2ου κεφαλαίου και θα θεωρήσουμε μία δυναμική ροή της μορφής (3.8), όπου  $(x_i, y_i)$  το σημείο προορισμού. Ο τελικός νόμος ελέγχου θα είναι:

$$u(x, y) = -V_{goal}(x, y) + V_{obst}(x, y) \quad (4.1)$$

και οι τροχιές του συστήματος προκύπτουν από την διαφορική εξίσωση του ρομποτικού κινηματικού μοντέλου:

$$\dot{h} = u \quad (4.2)$$

Στο παρακάτω γράφημα παρουσιάζονται τέσσερις διαφορετικές τροχιές με ξεχωριστές αρχικές συνθήκες. Όλες καταλήγουν φυσικά στο επιθυμητό σημείο προορισμού.

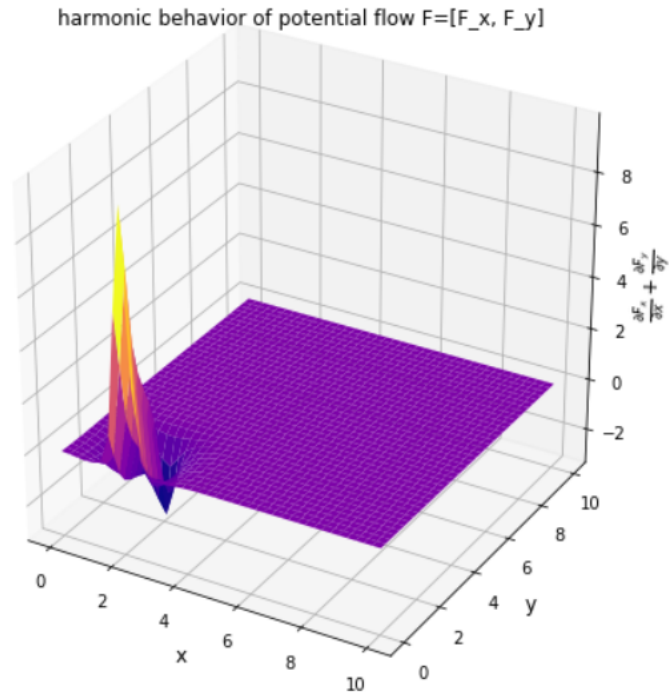


Σχήμα 4.1: Ρομποτική πλοήγηση από τέσσερα διαφορετικά σημεία εκκίνησης στο σημείο προορισμού

Επιπρόσθετα μας ενδιαφέρει να υπολογίσουμε την συνάρτηση προσέγγισης αρμονικότητας:

$$h(x, y) = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} \quad (4.3)$$

Ιδανικά επιθυμούμε η παραπάνω συνάρτηση να μηδενίζεται σε όλο το  $\mathcal{W}$ , καθώς αυτή η συμπεριφορά ταυτίζεται με μία δυναμική ροή παραχθείσα από βαθμωτή αρμονική συνάρτηση. Όμως όπως έχουμε εξηγήσει, η ανάγκη ικανοποίησης της συνθήκης (3.7) οδηγεί σε μία ασθενέστερη τήρηση της συνθήκης αρμονικότητας (3.6). Ρεαλιστικά, αναμένουμε πως στα τοιχώματα του εμποδίου η συνάρτηση  $h$  θα παίρνει μη μηδενικές τιμές, ενώ στα υπόλοιπα σημεία του χώρου  $\mathcal{W}$  θα μηδενίζεται. Όλα αυτά παρουσιάζονται στο παρακάτω γράφημα.



Σχήμα 4.2: Βαθμίδα της συνολικής δυναμικής ροής συναρτήσει των  $x, y$  συντεταγμένων

Αναλυτική παρουσίαση του κώδικα στο παράρτημα II.

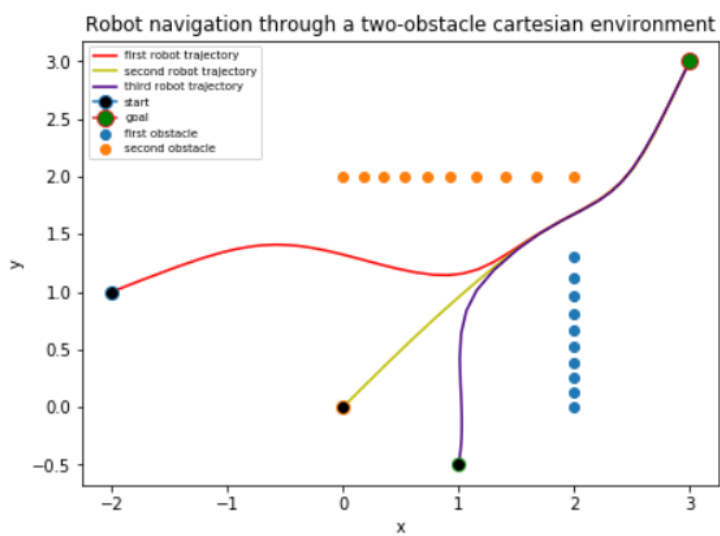
## 4.2 Περιβάλλον με δύο εμπόδια ευθύγραμμου τμήματος

Αυτό το παράδειγμα εμφανίζει πολλές ομοιότητες με το πρώτο, με την ουσιαστική διαφορά ότι το ρομπότ θα χρειαστεί να κινηθεί σε πιο στενούς χώρους. Η ύπαρξη δεύτερου εμποδίου αυξάνει τον κίνδυνο της εμφάνισης τοπικού ελαχίστου, γεγονός το οποίο μπορεί να οδηγήσει το ρομπότ σε άδιέξοδο και να καταστρέψει το σύστημα πλοήγησης. Παρόλα αυτά εάν οι δυναμικές ροές των εμποδίων εξασφαλίζουν σε μεγάλο βαθμό τις συνθήκες (3.6), (3.7) τότε ο νόμος ελέγχου θα φέρει επιθυμητό αποτέλεσμα. Η συνάρτηση  $u(x, y)$  θα συμπεριλαμβάνει αυτή τη φορά και μία δεύτερη δυναμική ροή εμποδίου:

$$u(x, y) = -V_{goal}(x, y) + V_{obst1}(x, y) + V_{obst2}(x, y) \quad (4.4)$$

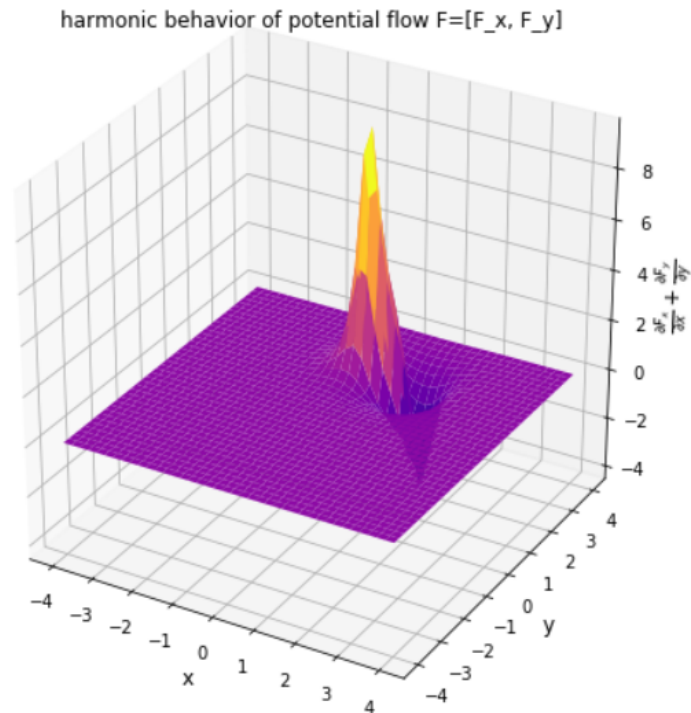
Στο παρακάτω γράφημα παρουσιάζεται η πλοήγηση του ρομπότ σε ένα περιβάλλον εργασίας με δύο εμπόδια ευθύγραμμου τμήματος, από τρία διαφορετικά σημεία εκκίνησης

μέχρι το σημείο προορισμού.



Σχήμα 4.3: Πλοήγηση αυτόνομου ρομπότ σε ένα περιβάλλον εργασίας με δύο εμπόδια. Έχουν επιλεγθεί τρία διαφορετικά σημεία εκκίνησης.

Όπως και στο προηγούμενο παράδειγμα μας ενδιαφέρει το πεδίο τιμών της συνάρτησης αρμονικότητας (4.3).



Σχήμα 4.4: Βαθμίδα της συνολικής δυναμικής ροής συναρτήσει των  $x, y$  συντεταγμένων.

Όπως θα περιμέναμε η συνάρτηση αρμονικότητας μηδενίζεται σε όλα τα σημεία του χώρου εργασίας εκτός από τα τοιχώματα των δύο εμποδίων. Στο παράρτημα III μπορεί να βρεθεί ο σχετικός κώδικας που παράγει τα παραπάνω αποτελέσματα.

### 4.3 Περιβάλλον με ένα μη κυρτό εμπόδιο

Τώρα ας προχωρήσουμε σε μία πιο περίπλοκη περίπτωση. Η ύπαρξη ενός μη κυρτού εμποδίου στον χώρο εργασίας απαιτεί πολύ μεγαλύτερη ακρίβεια στην κατασκευή του νόμου ελέγχου πλοήγησης. Όπως αναφέραμε στο 2ο κεφάλαιο, έχουν αναπτυχθεί διάφορες μεθοδολογίες μετασχηματισμού ενός καρτεσιανού χώρου εργασίας, σε έναν σημειακό χώρο εργασίας με σημειακά εμπόδια. Η ύπαρξη όμως μη κυρτών εμποδίων, δυσκολεύει σημαντικά αυτήν την διαδικασία και αυτός είναι ένας από τους λόγους αναζήτησης μίας διαφορετικής προσέγγισης, όπως παρουσιάζεται σε αυτήν την διπλωματική. Αρχικά χωρίζουμε το μη-κυρτό σχήμα σε ξεχωριστά ευθύγραμμα τμήματα και υπολογίζουμε τις δυναμικές ροές για κάθε τμήμα. Ουσιαστικά υπολογίζουμε μία δυναμική ροή για κάθε πλευρά του σχήματος. Η μεγαλύτερη δυσκολία σε αυτό το πρόβλημα συνήθως εμφανίζεται στα σημεία τομής των πλευρών, όπου οι συνθήκες για

εμφάνιση τοπικών ελαχίστων είναι πολύ ευνοϊκές. Για αυτό τον λόγο θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί στην επιλογή των συντελεστών  $c$  της εξίσωσης (3.15) έτσι ώστε να αποφύγουμε τυχόν ανεπιθύμητο αποτέλεσμα.

Οι συναρτήσεις συνολικής δυναμικής ροής  $V_x$  και  $V_y$  θα έχουν την μορφή (3.13) και (3.14) αντίστοιχα. Η συνάρτηση δυναμικής ροής  $v$  θα υπακούει στην εξίσωση (3.15) ενώ για τις συναρτήσεις κατανομής δύναμης  $\lambda_x, \lambda_y$  θα έχουμε:

$$\lambda_x = \text{sign}(x - x_d) \quad (4.5)$$

$$\lambda_y = \text{sign}(y - y_d) \quad (4.6)$$

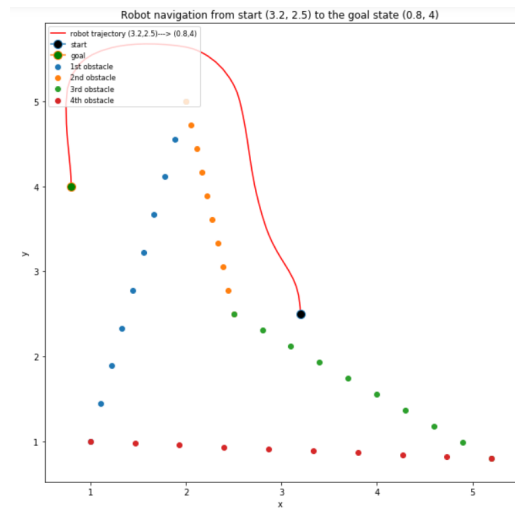
όπου η συνάρτηση  $\text{sign}$  είναι η γνωστή συνάρτηση προσήμου:

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (4.7)$$

Διαλέξαμε αυτές τις συναρτήσεις διαισθητικά, καθώς έχουν μοναδιαίο μέτρο και έτσι οι τιμές των δυναμικών  $V_x, V_y$  πάνω στο σύνορο του εμποδίου είναι ομοιόμορφα κατανεμημένες. Επιπρόσθετα μέσω των διαφορετικών προσήμων προσδίδουμε στις δυναμικές ροές τον κατάλληλο προσανατολισμό. Όπως και στην ενότητα 5.1 εισάγουμε τον έλεγχο:

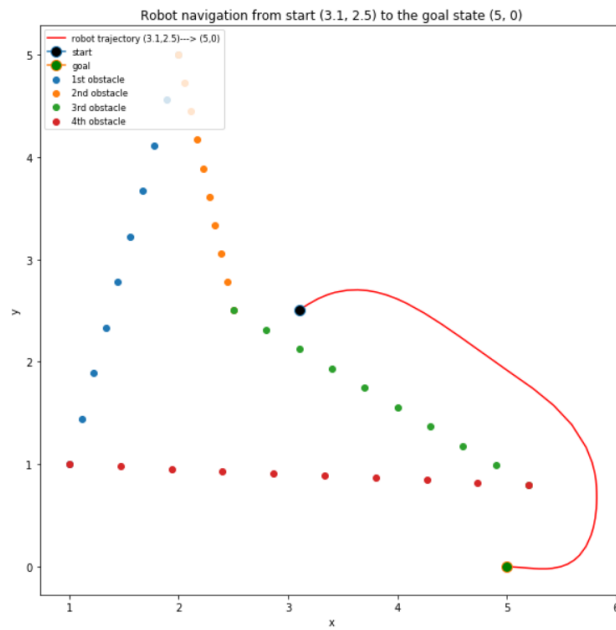
$$u(x, y) = -V_{\text{goal}}(x, y) + V_{\text{obst}}(x, y) \quad (4.8)$$

Στο παρακάτω γράφημα παρουσιάζεται η πλοήγηση από ένα επικίνδυνο σημείο δίπλα σε μία κυρτή γωνία, μέχρι το σημείο προορισμού το οποίο βρίσκεται στην άλλη πλευρά του εμποδίου.

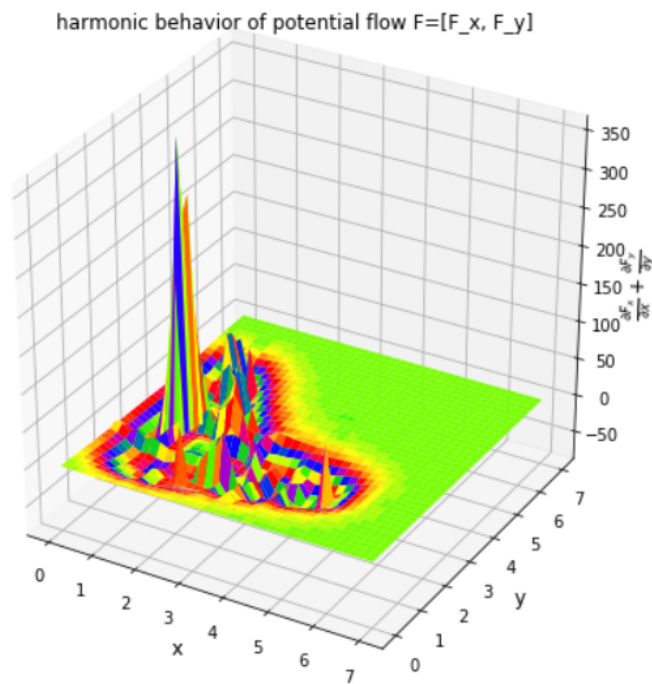


Σχήμα 4.5: Αυτόνομη πλοήγηση (3.2, 2.5) -> (0.8, 4)

Ακόμα ένα παράδειγμα με το σημείο προορισμού αυτή τη φορά να βρίσκεται πιο κοντά στην κάτω πλευρά του εμποδίου. Το αυτόνομο ρομπότ κινείται επιτυχημένα και στις δύο περιπτώσεις, καθώς παρατηρούμε ότι οι τροχιές κίνησης δεν διαπερνούν το εμπόδιο. Επιπλέον όπως και στα προηγούμενα παραδείγματα, μας ενδιαφέρει η συμπεριφορά της συνάρτησης αρμονικότητας (4.3) στον χώρο εργασίας.



Σχήμα 4.6: Αυτόνομη πλοήγηση  $(3.1, 2.5) \rightarrow (5, 0)$



Σχήμα 4.7: Βαθμίδα της συνολικής δυναμικής ροής συναρτήσει των  $x, y$  συντεταγμένων



Η συνάρτηση  $h$  παίρνει μη μηδενικές τιμές στα εσωτερικά σημεία του εμποδίου καθώς και πάνω στα τοιχώματα του. Επομένως η εμφάνιση τοπικών ελαχίστων είναι ένα φαινόμενο το οποίο μπορεί να παρουσιαστεί κυρίως στο εσωτερικό και στο σύνορο του εμποδίου. Αυτή είναι μία παραχώρηση που είμαστε διατεθειμένοι να κάνουμε, καθώς οι συνέπειες της δεν είναι καταστροφικές για τον νόμο ελέγχου, για να ικανοποιήσουμε σε έναν μεγάλο βαθμό την συνοριακή συνθήκη (3.7). Στο παράρτημα IV μπορεί να βρεθεί ο σχετικός κώδικας.

## 4.4 Λαβύρινθος

Ας προχωρήσουμε τώρα στο πρόβλημα του λαβύρινθου, μία κλασσική εφαρμογή στην αυτόνομη ρομποτική πλοήγηση. Ο νόμος ελέγχου που θα σχεδιάσουμε οφείλει να καθοδηγήσει το ρομπότ στην έξοδο του λαβυρίνθου αποφεύγοντας οποιαδήποτε σύγκρουση με τα τοιχώματα του. Σε σύγκριση με τις προηγούμενες εφαρμογές, ο βαθμός δυσκολίας της σχεδίασης κατάλληλου νόμου ελέγχου είναι υψηλότερος, καθώς η ύπαρξη πολλών εμποδίων (τοιχωμάτων) σε μικρές αποστάσεις μεταξύ τους, δημιουργεί τον κίνδυνο εμφάνισης τοπικών ελαχίστων στον χώρο εργασίας. Η μεθοδολογία που ακολουθείται σε αυτό το παράδειγμα δεν διαφέρει από αυτήν που έχουμε ήδη αναπτύξει στα προηγούμενα υποκεφάλαια, παρόλα αυτά θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί στην επιλογή διαφόρων συντελεστών. Όπως και στα προηγούμενα παραδείγματα, θα μοντελοποιήσουμε τις δυναμικές ροές του κάθε εμποδίου ξεχωριστά με τη βοήθεια των εξισώσεων (3.13), (3.14), (3.15) και μέσω της υπέρθεσης τους θα καταλήξουμε στην συνολική δυναμική ροή που θα καθορίσει τις τροχιές της κίνησης του ρομπότ. Για την συναρτήσεις κατανομής δύναμης θα θεωρήσουμε:

$$\lambda_x(x) = x - x_d \quad (4.9)$$

$$\lambda_y(y) = y - y_d \quad (4.10)$$

Ο τελικός νόμος ελέγχου, όπως και στις προηγούμενες εφαρμογές, θα είναι η υπέρθεση ενός ελκτικού δυναμικού πεδίου που σπρώχνει το ρομπότ στο σημείο προορισμού, με τα τεχνητά δυναμικά πεδία που παράγονται από τα τοιχώματα του λαβυρίνθου.

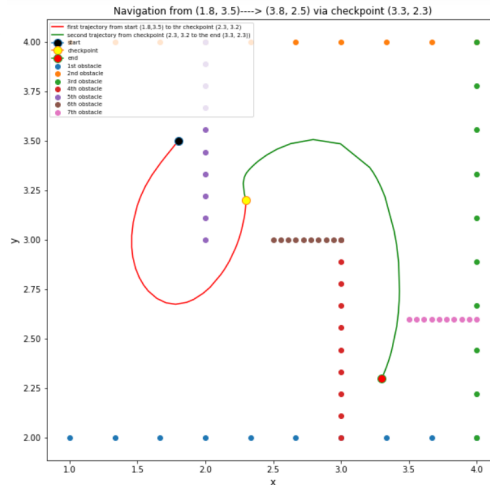
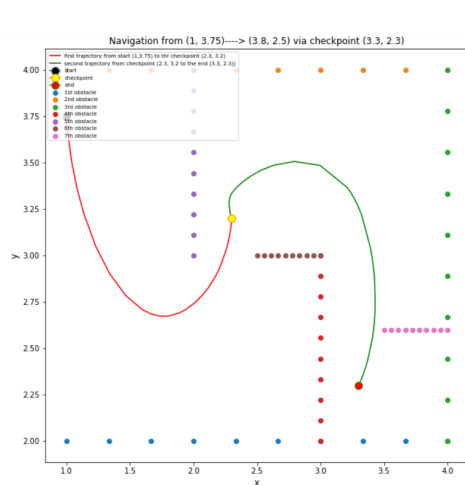
$$u(x, y) = -V_{goal} + \sum_{n=1}^K V_n \quad (4.11)$$

Για την λήψη των τροχιών κίνησης θα ενσωματώσουμε τον έλεγχο  $u(x, y)$  στο ρομποτικό μοντέλο:

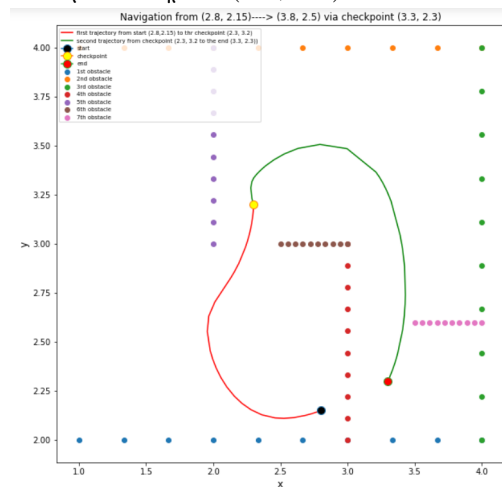
$$\dot{x} = u \quad (4.12)$$

Στα επόμενα διαγράμματα καταγράφονται τρεις διαφορετικές τροχιές από τρία σημεία εκκίνησης μέχρι το τελικό σημείο. Σημαντικό να αναφέρουμε ότι για λόγους αποδοτικότητας του αλγορίθμου έχει επιλεγεί ένα ενδιάμεσο σημείο (checkpoint) στο οποίο

το ρομπότ σταματά στιγμιαία. Επομένως η συνολική κίνηση πραγματοποιείται σε δύο στάδια.



(α') Πλοήγηση (1, 3.75) → (3.8, 2.25) με (β') Πλοήγηση (1.8, 3.5) → (3.8, 2.25) με ενδιάμεσο σημείο (3.3, 2.3)



(γ') Πλοήγηση (2.8, 2.15) → (3.8, 2.25) με ενδιάμεσο σημείο (3.3, 2.3)

Στο παράρτημα V παρουσιάζεται ο κώδικας που παράγει τα παραπάνω διαγράμματα.

## 4.5 Συμπεράσματα

Η μεθοδολογία που παρουσιάστηκε σε αυτήν την πτυχιακή, καθιστά έναν εναλλακτικό τρόπο προσέγγισης της αυτόνομης ρομποτικής πλοήγησης. Κατασκευάσαμε νόμους ελέγχου με αναλυτική μορφή, παρόμοια με τα έργα [8], [6], οι οποίοι μπορούν

να προσφέρουν επιθυμητά αποτελέσματα ανεξαρτήτως της αρχικής θέσης του ρομπότ. Επιπλέον έγινε η χρήση τεχνητών δυναμικών ροών για την προσομοίωση της επιθυμητής επιρροής των εμποδίων αλλά και του τελικού στόχου, μία προσέγγιση εμπνευσμένη από τα [4],[1]. Όμως η καινοτομία αυτού του κειμένου βρίσκεται στην μίξη της κατασκευής του νόμου ελέγχου με την μεθοδολογία του τελεστή Koorman. Στην ουσία χρησιμοποιήσαμε ένα άθροισμα των ιδιοσυναρτήσεων Koorman του δυναμικού συστήματος (3.9) για την κατασκευή μίας δυναμικής ροής η οποία θα ικανοποιούσε τις συνοριακές συνθήκες (3.6),(3.7). Συνθήκες οι οποίες είναι αναγκαίες για την εύρυθμη λειτουργία του νόμου πλοήγησης. Βέβαια υπάρχει αρκετός χώρος για περαιτέρω ανάλυση αυτής της προσέγγισης. Για παράδειγμα κάποιος θα μπορούσε να επεκταθεί στην κατάλληλη επιλογή των συντελεστών  $c_n$  της εξίσωσης (3.17), αλλά και των συναρτήσεων  $\lambda_x, \lambda_y$  των εξισώσεων (3.15), (3.16), αντιμετωπίζοντας στην ουσία αυτήν την επιλογή ως πρόβλημα βελτιστοποίησης του νόμου ελέγχου. Επιπρόσθετα θα μπορούσε να επεκταθεί η παραπάνω μεθοδολογία και σε χώρους εργασίας με πολλαπλά ρομπότ [7] ή ακόμα και σε χώρους εργασίας με μη-στατικά εμπόδια. Τέλος παρουσιάσαμε διάφορες εφαρμογές με την παραπάνω μεθοδολογία, ενώ στο επόμενο κεφάλαιο παραθέτουμε τον κώδικα που παράγει όλα τα αποτελέσματα.

# Κεφάλαιο 5

## Παραρτήματα

### 5.1 Παράρτημα I

*Εφαρμογή EDMD* : Θεωρούμε το μη γραμμικό δυναμικό σύστημα της μορφής:

$$\begin{aligned}\dot{x}_1 &= \lambda x_1 \\ \dot{x}_2 &= \mu x_2 + (2\lambda - \mu)cx_1^2\end{aligned}$$

Με την χρήση της μεθόδου EDMD που παρουσιάστηκε στην ενότητα 2.6 θα προσπαθήσουμε να εξάγουμε τις ιδιοτιμές και ιδιοσυναρτήσεις Koopman , οι οποίες θα μας επιτρέψουν να αναπαραστήσουμε την δυναμική της παραπάνω εξίσωσης με έναν εναλλακτικό και ίσως πιο βολικό τρόπο.

Αρχικά πρέπει να ορίσουμε την βιβλιοθήκη συναρτήσεων  $\Psi$ , η οποία όπως αναλύσαμε στην ενότητα 3.6 είναι καθοριστική για τον αποδοτικότητα και ακρίβεια του αλγορίθμου. Η επιλογή της θα πρέπει να προσεγγίζει όσο περισσότερο γίνεται τις ιδιοσυναρτήσεις. Όπως έχει αποδειχθεί στο [11] μία άκρως ικανοποιητική επιλογή βιβλιοθήκης για το συγκεκριμένο σύστημα είναι:

$$\Psi(x) = \begin{bmatrix} x_1 \\ x_2 - cx_1^2 \\ x_1^2 \\ 1 \end{bmatrix} \quad (5.1)$$

Στη συνέχεια με τη βοήθεια της συνάρτησης `odeint` από βιβλιοθήκη της `python` , θα παράξουμε μία βάση δεδομένων  $(x_k, y_k)$  όπου ισχύει  $S(x_k) = y_k$  ( $S$  ο τελεστής εξέλιξης του διανύσματος κατάστασης  $x$  ). Επιλέγουμε  $\lambda = 0.3, \mu = 0.2, c = -0.5$  και αρχική συνθήκη για το διάνυσμα κατάστασης  $x_0 = [0.1, 0.1]^T$  ,

```
##### solving differential equation
def diff_eq(x,t,lam,m,c):
    x1, x2= x
    x_dot=[lam*x1, m*x2+c*np.power(x1,2)*(2*lam-m)]
    return x_dot
```

```
lam,m,c=0.3,0.2,-0.5 ##### set parameters
x0=[0.1, 0.1] ##### initial conditions
print(np.shape(x0))
t=np.linspace(0,10,1000)

sol=odeint(diff_eq, x0, t, args=(lam,m,c))
X=np.transpose(sol) ##### x database

print(np.shape(X))
```

```
(2,)
(2, 1000)
```

```
##### y database
def F(X,l,m,c):
    fu=[1*X[0], m*X[1]+c*np.power(X[0],2)*(2*1-m)]
    return fu
```

Σχήμα 5.1: Αρχικές συνθήκες διανύσματος κατάστασης και παραγωγή βάσης δεδομένων για τον αλγόριθμο

Με τη βοήθεια των εξισώσεων (2.20-2.23) υπολογίζουμε την προσέγγιση της μήτρας Koopman και εξάγουμε τις ιδιοσυναρτήσεις και ιδιοτιμές της. Αυτή η διαδικασία φαίνεται στο παρακάτω κομμάτι κώδικα.

```

##### Calculate Koopman matrix
##### Equations (29-32)
psi_x=PSI(X,c)
psi_y=PSI(y,c)

M=len(psi_x[0])
G=np.matmul(psi_x, psi_x.T)/M
A=np.matmul(psi_x, psi_y.T)/M

Koop_m=pinv(G)@A ##### Koopman matrix
print(np.shape(psi_y))
print(M)

```

```

(4, 1000)
1000

```

Σχήμα 5.2: Μήτρα Koopman

```

##### Extract eigenvalues and eigenvectors
print(Koop)
lamda, w=eig(Koop)
print(lamda)
print(np.shape(w))

[[ 3.00000000e-01  3.55271368e-15  2.22044605e-16 -3.33066907e-16]
 [ 6.66133815e-16  2.00000000e-01 -5.55111512e-17 -6.24500451e-17]
 [-5.32907052e-15  1.42108547e-14  6.00000000e-01 -1.22124533e-15]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]]
[0.3 0.6 0.2 0. ]
(4, 4)

##### caculate eigenfunctions with the help of the above eigenvectors (equation 32)
phi=np.matmul(w,psi)
print(np.shape(phi))

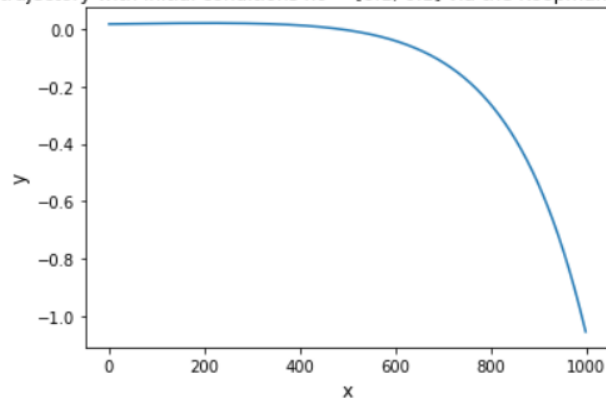
(4, 1000)

```

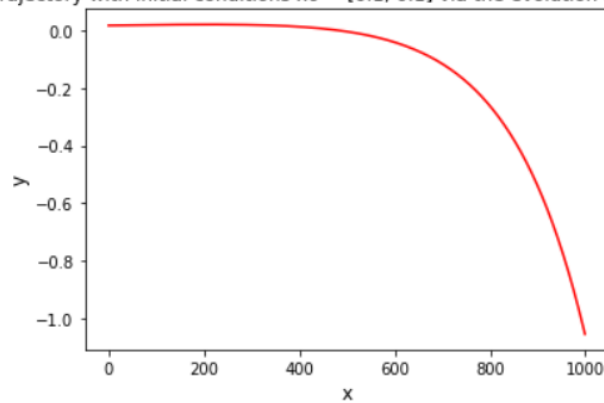
Σχήμα 5.3: Υπολογισμός των ιδιοσυναρτήσεων

Για να επαληθεύσουμε το αποτέλεσμα μας θα επανέλθουμε στην εξίσωση (2.18), η οποία υποδεικνύει ότι μπορούμε να τρέξουμε μία τροχιά μέσω, είτε του τελεστή Koopman είτε του τελεστή εξέλιξης  $S$ , και να λάβουμε το ίδιο αποτέλεσμα.

trajectory with initial conditions  $x_0 = [0.1, 0.1]^T$  via the Koopman operator



trajectory with initial conditions  $x_0 = [0.1, 0.1]^T$  via the evolution F operator



Σχήμα 5.4: Τροχιές του  $x_1$  με αρχικές συνθήκες  $x_0 = [0.1, 0.1]^T$  και με χρήση των δύο διαφορετικών τελεστών (K τελεστής στο άνω διάγραμμα και S τελεστής στο κάτω διάγραμμα.)

Όπως παρατηρούμε οι δύο τελεστές παράγουν πανομοιότυπες τροχιές όπως περιγράφει η εξίσωση 2.18. Η πολύ υψηλή ακρίβεια σε αυτό το παράδειγμα σχετίζεται με την επιλογή της βιβλιοθήκης συναρτήσεων, η οποία ουσιαστικά περιγράφει σε μεγάλο βαθμό τον χώρο των ιδιοσυναρτήσεων. Ο κώδικας που παράγει το ζευγάρι των γραφικών παρουσιάζεται παρακάτω:

```
##### Koopman modes
G=np.array([X[0],X[1]])
print(np.shape(G))
print(np.shape(psi_d))
BT=np.matmul(G, psi_d)
B=BT.T

W_star=inv(w)
V=np.matmul(B.T, W_star.T)
print(np.shape(V))
print(np.shape(G))
```

```
(2, 1000)
(1000, 4)
(2, 4)
(2, 1000)
```

```
##### trajectories with two different operators
##### equation
D=np.diag(lamda)
Koop_traj=V @ D @ phi #####
```



## 5.2 Παράρτημα II

Αρχικά ορίζουμε τον τύπο των ιδιοσυναρτήσεων Koorman :

```
: ##### Eigenfunctions

def dist(x,y,xd,yd,n):

    return np.exp(-n*(np.power(x-xd,2) + np.power(y-yd,2)))
```

Προχωράμε σε παραμετροποίηση των μεταβλητών  $x_d, y_d$  και υπολογίζουμε τα επι-καμπύλια ολοκληρώματα (3.13), (3.14) με την νέα μεταβλητή.

```
In [4]: ##### parameterization with polar coordinates
def line_x(x,y,n,c):
    ##yd=2
    ##xd=2*cot(theta)
    #(x-2*(np.cos(theta)/np.sin(theta)))
    f=lambda theta: c*dist(x,y,2*(np.cos(theta)/np.sin(theta)),2,n)*np.cos(x-2*(np.cos(theta)/np.sin(theta))
    return f

In [5]: ##### parameterization with polar coordinates
def line_y(x,y,n,c):
    ##yd=2
    ##xd=2*cot(theta)
    #(y-2)
    f=lambda theta: c*dist(x,y,2*np.cos(theta)/np.sin(theta),2,n)*np.sin(y-2)
    return f

In [6]: ##### Calculate potential flow of panel
def Int_func(x,y,M):
    f_x=lambda theta: 0
    f_y=lambda theta: 0
    c=50
    for i in range(1,M):
        f_x=lambda theta, s_x=f_x, c_x=line_x(x,y,i,c): s_x(theta) + c_x(theta)
        f_y=lambda theta, s_y=f_y, c_y=line_y(x,y,i,c): s_y(theta) + c_y(theta)
    Integ_x,error_x=quad(f_x,np.pi/4,np.pi/2)
    Integ_y,error_y=quad(f_y,np.pi/4,np.pi/2)
    return [Integ x, Integ y]
```

Αποθηκεύουμε τα αποτελέσματα της δυναμικής ροής σε μία συνάρτηση lambda έτσι ώστε να έχουμε μία αναλυτική συνάρτηση δυναμικής ροής.

```
def PHI(M):
    phi=lambda x,y: Int_func(x,y,M)

    return phi
```

Εφαρμόζουμε την συνολική δυναμική ροή στο κινηματικό ρομποτικό μοντέλο και επιλέγουμε τέσσερις διαφορές αρχικές καταστάσεις. Ως αποτέλεσμα έχουμε τέσσερις διαφορετικές τροχιές οι οποίες καταλήγουν στο σημείο προορισμού και ταυτόχρονα αποφεύγουν το εμπόδιο, όπως φαίνεται στο σχήμα 1.

```

: ##### kinematic differential equation
def robot_ode(t,h):
    #source_strength=40
    x=h[0]
    y=h[1]
    dh_dt=[(-20*(x-1)/(np.power(x-1,2)+np.power(y-3,2))+F_x(x,y), (-20*(y-3)/(np.power(x-1,2)+np.power(y-3,2))+F_y(x,y))]
    #dh_dt=[-40*sink[0](x,y)+F_x(x,y), -40*sink[1](x,y)+F_y(x,y)]
    return dh_dt

:
: #h0=[0.5,1.9]
#sol=solve_ivp(robot_ode,[0,20],h0, method='BDF')

: ##### 4 different starting points
##### All trajectories lead to the final destination
start=np.array([[0.5, 1.9], [0,0], [-0.5,1], [1,1.6]]) ### starting points
traj=[]

for i in range(0,4):
    traj.append(solve_ivp(robot_ode, [0,20], start[i], method='BDF')) ##### 4 trajectories

```

## 5.3 Παράρτημα III

Ο κώδικας σε αυτήν την εφαρμογή είναι παρόμοιος με τον κώδικα του παραρτήματος II με την μικρή διαφορά ότι σε αυτό το πρόβλημα προσθέσαμε ένα δεύτερο ευθύγραμμο εμπόδιο. Επομένως ο νόμος ελέγχου θα έχει την μορφή της εξίσωσης (4.4). Η αλλαγή αυτή φαίνεται στο παρακάτω κομμάτι κώδικα:

```
F_sx=lambda x,y: F_slope(x,y)[0]
F_sy=lambda x,y: F_slope(x,y)[1]

def robot_ode_s(t,h):
    #source_strength=50
    x=h[0]
    y=h[1]
    dh_dt=[(-50*(x-3)/(np.power(x-3,2)+np.power(y-3,2))+F_sx(x,y)+F_x(x,y), (-50*(y-3)/(np.power(x-3,2)+np.power(y-3,2))+F_sy(x,y)+F_y(x,y))]
    #dh_dt=[-40*sink[0](x,y)+F_x(x,y), -40*sink[1](x,y)+F_y(x,y)]
    return dh_dt
```

Στο σχήμα 2 φαίνονται οι τροχιές που παράγονται από τρία διαφορετικά αρχικά σημεία εκκίνησης, με την εφαρμογή του παραπάνου ελέγχου στο ρομποτικό κινηματικό μοντέλο.

## 5.4 Παράρτημα IV

Αρχικά ορίζουμε τον τύπο των ιδιοσυναρτήσεων:

```
: ##### Eigenfunctions
def dist(x,y,xd,yd,n):
    return np.exp(-n*(np.power(x-xd,2) + np.power(y-yd,2)))
```

Θέλουμε να υπολογίσουμε την ολική δυναμική ροή για κάθε τμήμα (panel) μέσω των τύπων (3.13), (3.14). Επομένως προχωράμε στην παραμετροποίηση των  $x_d, y_d$  και υπολογίζουμε το επικαμπύλιο ολοκλήρωμα συναρτήσεως της νέας μεταβλητής  $dt$ .

```
In [3]: def line_x(A,B,x,y,n,c):
        #*(x-xd)
        ##### parameterization of xd
        f_x= lambda t: c*dist(x,y,(1-t)*A[0]+t*B[0],(1-t)*A[1]+t*B[1],n)*np.sign(x-((1-t)*A[0]+t*B[0]))
        return f_x

In [4]: def line_y(A,B,x,y,n,c):
        #*(y-yd)
        ##### parameterization of yd
        f_y= lambda t: c*dist(x,y,(1-t)*A[0]+t*B[0],(1-t)*A[1]+t*B[1],n)*np.sign(y-((1-t)*A[1]+t*B[1]))
        return f_y

In [6]: #####3
        ##### Calculate the total potential flow
        def Int_func(A,B,x,y,M,c):
            f_x= lambda t: 0
            f_y= lambda t: 0

            metro=np.sqrt(np.power(B[0]-A[0],2) + np.power(B[1]-A[1],2))

            ##
            #c_s=np.linspace(10,50,M-1)

            for i in range(1,M-1):
                f_x= lambda t, s_x=f_x, c_x=line_x(A,B,x,y,i,c): s_x(t) + c_x(t)
                f_y= lambda t, s_y=f_y, c_y=line_y(A,B,x,y,i,c): s_y(t) + c_y(t)

            Integ_x,error_x=quad(f_x,0,1)
            Integ_y,error_y=quad(f_y,0,1)

            return [metro*Integ_x, metro*Integ_y]
```

Αρχικοποιούμε το μη-κυρτό εμπόδιο και υπολογίζουμε την δυναμική ροή γύρω του.

```

: ##### create non-convex obstacle
##### each (Ai,Bi) pair creates a side of the obstacle
#####
ar_A=np.array([[1,1], [2,5], [2.5,2.5], [1,1]])
ar_B=np.array([[2,5], [2.5,2.5], [5.2,0.8], [5.2,0.8]])

: ##### divide the obstacle in panels and calculate the potential flows for each segment

Flow_x=[]
Flow_y=[]

for i in range(0, len(ar_A)):
    Vx,Vy=get_field(ar_A[i], ar_B[i], 10, 60)
    Flow_x.append(Vx)
    Flow_y.append(Vy)

: Tot_Fx=lambda x,y: Flow_x[0](x,y)+ Flow_x[1](x,y)+ Flow_x[2](x,y)+ Flow_x[3](x,y)
Tot_Fy=lambda x,y: Flow_y[0](x,y)+ Flow_y[1](x,y)+ Flow_y[2](x,y)+ Flow_y[3](x,y)

```

Τέλος εισάγουμε την δυναμική ροή στο ρομποτικό κινηματικό μοντέλο (4.12) και ορίζουμε τις αρχικές συνθήκες. Ως αποτέλεσμα προκύπτουν οι επιθυμητές τροχιές οι οποίες οδηγούν το ρομπότ στο σημείο προορισμού.

```

: ##### Kinematic differential equation
##### destination point (5,0)
##### create a sink flow towards the final point and add the total potential flow of the obstacle
def track(t,h):
    x=h[0]
    y=h[1]
    dh_dt=[(-100*(x-5)/(np.power(x-5,2)+np.power(y-0,2))+Tot_Fx(x,y), (-300*(y-0)/(np.power(x-5,2)+np.power(y-0,2))+Tot_Fy(x,y)

    return dh_dt

: ##### set initial conditions
start=[3.1,2.5]
sol_t=solve_ivp(track, [0,20], start, method='BDF')

```

Τα αποτελέσματα του παραπάνω κώδικα καταγράφονται στην υποπαράγραφο 4.3.

## 5.5 Παράρτημα V

Ο κώδικας σε αυτήν την εφαρμογή έχει πολλές ομοιότητες με τους κώδικες των παραρτημάτων II, III και IV. Αρχικά ορίζουμε τις ιδιοσυναρτήσεις Koopman και στη συνέχεια υπολογίζουμε τα παραμετροποιημένα ολοκληρώματα (3.15), (3.16).

```
def dist(x,y,xd,yd,n):
    return np.exp(-n*(np.power(x-xd,2) + np.power(y-yd,2)))

def line_x(A,B,x,y,n,c):
    if A[1]-B[1]==0:
        f1=lambda theta: c*dist(x,y,A[1]*(np.cos(theta)/np.sin(theta)),A[1],n)*(x-A[1]*(np.cos(theta)/np.sin(theta)))
        return f1

    if A[0]-B[0]==0:
        f2=lambda theta: c*dist(x,y,A[0],A[0]*np.tan(theta),n)*(x-A[0])
        return f2

def line_y(A,B,x,y,n,c):
    if A[1]-B[1]==0:
        f1=lambda theta: c*dist(x,y,A[1]*np.cos(theta)/np.sin(theta),A[1],n)*(y-A[1])
        return f1

    if A[0]-B[0]==0:
        f2=lambda theta: c*dist(x,y,A[0],A[0]*np.tan(theta),n)*(y-(A[0]*np.tan(theta)))
        return f2
```

Σχήμα 5.5: Παραμετροποίηση εξισώσεων (3.13), (3.14)

```
def Int_func(A,B,x,y,M,c):
    if A[1]-B[1]==0:
        f_x=lambda theta: 0
        f_y=lambda theta: 0

    for i in range(1,M):
        f_x=lambda theta, s_x=f_x, c_x=line_x(A,B,x,y,i,c): s_x(theta) + c_x(theta)
        f_y=lambda theta, s_y=f_y, c_y=line_y(A,B,x,y,i,c): s_y(theta) + c_y(theta)

    l1=max(A[0],B[0])
    l2=min(A[0],B[0])
    Integ_x,error_x=quad(f_x,np.arctan(B[1]/l1),np.arctan(A[1]/l2))
    Integ_y,error_y=quad(f_y,np.arctan(B[1]/l1),np.arctan(A[1]/l2))
    return [Integ_x, Integ_y]

    if A[0]-B[0]==0:
        f_x=lambda theta: 0
        f_y=lambda theta: 0

    for i in range(1,M):
        f_x=lambda theta, s_x=f_x, c_x=line_x(A,B,x,y,i,c): s_x(theta) + c_x(theta)
        f_y=lambda theta, s_y=f_y, c_y=line_y(A,B,x,y,i,c): s_y(theta) + c_y(theta)

    l1=max(A[1],B[1])
    l2=min(A[1],B[1])
    Integ_x,error_x=quad(f_x,np.arctan(l2/A[0]),np.arctan(l1/B[0]))
    Integ_y,error_y=quad(f_y,np.arctan(l2/A[0]),np.arctan(l1/B[0]))
    return [Integ_x, Integ_y]
```

Σχήμα 5.6: Υπολογισμός ολοκληρωμάτων (3.15), (3.16)

Μέχρι εδώ έχουμε επαναλάβει την διαδικασία που ακολουθήσαμε και στα προηγούμενα παραρτήματα. Η ουσιαστική διαφορά βρίσκεται στην κατασκευή του λαβυρίνθου και της συνολικής δυναμικής ροής που προκύπτει από την υπέρθεση των δυναμικών ροών του κάθε τοιχώματος, όπως περιγράφεται από την εξίσωση (4.11).

```

#### ##### build labyrinth
#####
def flow F(c):
    F_lab=[]
    A_list=np.array([[1,2],[1,4],[4,2],[3,2],[2,4],[2.5,3],[4,2.6]])
    B_list=np.array([[4,2],[4,4],[4,4],[3,3],[2,3],[3,3],[3.5,2.6]])
    #c_list=[10,30,30,500,700,30]

    for i in range(0,len(A_list)):
        F_lab.append(PHI(A_list[i],B_list[i],30,c[i]))

    Flab_x=lambda x,y: F_lab[0](x,y)[0] + F_lab[1](x,y)[0] + F_lab[2](x,y)[0] +F_lab[3](x,y)[0] +F_lab[4](x,y)[0] +F_lab[5](x,y)[0]
    Flab_y=lambda x,y: F_lab[0](x,y)[1] + F_lab[1](x,y)[1] + F_lab[2](x,y)[1] +F_lab[3](x,y)[1] +F_lab[4](x,y)[1] +F_lab[5](x,y)[1]

    return Flab_x, Flab_y ##### potential flow

```

Σχήμα 5.7: Κατασκευή λαβυρίνθου και υπολογισμός της συνολικής δυναμικής ροής μέσω της υπέρθεσης των δυναμικών ροών του κάθε τοιχώματος

Φυσικά για να υπολογίσουμε τις τροχιές του ρομπότ εισάγουμε την παραπάνω δυναμική ροή ως έλεγχο στο ρομποτικό κινηματικό μοντέλο (4.12) και ορίζουμε τους συντελεστές και τις αρχικές συνθήκες.

```

def trajectory(start,checkpoint,end,c1,c2,strength1,strength2):
    sol1=solve_ivp(labyrinth, [0,20], start, args=(checkpoint[0], checkpoint[1],c1,strength1), method='BDF')
    sol2=solve_ivp(labyrinth, [0,20], checkpoint, args=(end[0], end[1],c2,strength2), method='BDF')

    return sol1, sol2

##### Coefficients
c1=[10,30,30,500,700,30,10]
c2=[500,100,500,700,10,400,20]
start=[2.8, 2.15] ##### initial conditions
checkpoint=[2.3, 3.2] ##### checkpoint
end=[3.3, 2.3] #####destination
sol1,sol2=trajectory(start, checkpoint, end,c1,c2,80,200) ##### calculate trajectory

fig,ax=plt.subplots(figsize=[10,10])
ax.plot(sol1.y[0], sol1.y[1], 'r', label='first trajectory from start (2.8,2.15) to thr checkpoint (2.3, 3.2)')
ax.plot(sol2.y[0], sol2.y[1], 'g', label='second trajectory from checkpoint (2.3, 3.2 to the end (3.3, 2.3))')
ax.scatter(x1,y1, label='1st obstacle')
ax.scatter(x2,y2, label='2nd obstacle')
ax.scatter(x3,y3, label='3rd obstacle')
ax.scatter(x4,y4, label='4th obstacle')
ax.scatter(x5,y5, label='5th obstacle')
ax.scatter(x6,y6, label='6th obstacle')
ax.scatter(x7,y7, label='7th obstacle')
ax.plot(start[0], start[1], marker='o',markersize=10, markerfacecolor='black', label='start')
ax.plot(checkpoint[0], checkpoint[1], marker='o',markersize=10, markerfacecolor='yellow', label='checkpoint')
ax.plot(end[0], end[1], marker='o',markersize=10, markerfacecolor='red', label='end')
legend=ax.legend(loc='upper left', fontsize='x-small')
plt.title('Navigation from (2.8, 2.15)----> (3.8, 2.5) via checkpoint (3.3, 2.3)')
plt.xlabel('x', fontsize='large')
plt.ylabel('y', fontsize='large')

```

Σχήμα 5.8: Υπολογισμός τροχιάς μέσω της εξίσωσης του ρομποτικού κινηματικού μοντέλου

Τα αποτελέσματα του παραπάνω κώδικα καταγράφονται στην ενότητα 4.4

# Bibliography

- [1] Sheldon Axler, Paul Bourdon, and Ramey Wade. *Harmonic function theory*, volume 137. Springer Science & Business Media, 2013.
- [2] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS one*, 11(2):e0150171, 2016.
- [3] Klaus-Jochen Engel, Rainer Nagel, and Simon Brendle. *One-parameter semi-groups for linear evolution equations*, volume 194. Springer, 2000.
- [4] Daniel E Koditschek and Elon Rimon. Robot navigation functions on manifolds with boundary. *Advances in applied mathematics*, 11(4):412–442, 1990.
- [5] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.
- [6] Savvas G Loizou. Closed form navigation functions based on harmonic potentials. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6361–6366. IEEE, 2011.
- [7] Savvas G Loizou. The multi-agent navigation transformation: tuning-free multi-robot navigation. In *Robotics: Science and Systems*, volume 6, pages 1516–1523, 2014.
- [8] Savvas G Loizou and Elon D Rimon. Correct-by-construction navigation functions with application to sensor based robot navigation. *arXiv preprint arXiv:2103.04445*, 2021.
- [9] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.
- [10] Igor Mezić and Andrzej Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1-2):101–133, 2004.
- [11] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.



- [12] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- [13] Peter J Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in fluids*, 50:1123–1130, 2011.
- [14] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.
- [15] Holger Wendland. Meshless galerkin methods using radial basis functions. *Mathematics of computation*, 68(228):1521–1531, 1999.
- [16] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.