

ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ II

ΕΡΓΑΣΙΑ I

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΧΟΥΛΙΑΡΑΣ ΒΑΣΙΛΕΙΟΣ

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ ΦΟΙΤΗΤΗ: 1083938

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

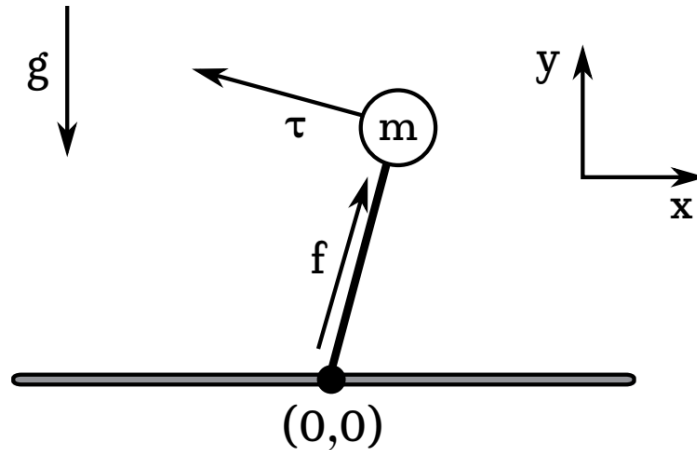
Πίνακας περιεχομένων

1. MODELING.....	4
1.1 ΣΥΣΤΗΜΑ ΥΛΟΠΟΙΗΣΗ	4
1.2 ΕΞΙΣΩΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ	5
1.3 ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΟΣ	6
2. QP – ADMM.....	10
2.1 Τετραγωνικός προγραμματισμός.....	10
2.2 Quadratic Programming via Alternative Directions Method of Multiplier:	11
2.3 Υλοποίηση Αλγόριθμου QP _ADMM:	12
3 Control via QP	13

1. MODELING

1.1 ΣΥΣΤΗΜΑ ΥΛΟΠΟΙΗΣΗ

Το σύστημα που υλοποιείται φαίνεται στην Εικόνα 1.1, είναι μία απλοποιημένη μορφή ενός hover robot. Πιο συγκεκριμένα, η μάζα μπορεί να κινηθεί σε οποιοδήποτε σημείο του δισδιάστατου χώρου, καθώς η δοκός που ασκεί την δύναμη \mathbf{f} είναι αμελητέα δηλαδή δεν εμποδίζει την μάζα να κινηθεί προς την διεύθυνσή της. Η δύναμη τ είναι πάντοτε κάθετη στην δύναμη \mathbf{f} , και έχει πάντα κατεύθυνση 90° αριστερά από την δύναμη \mathbf{f} . Στην μάζα ασκείται επιπλέον μία ακόμη δύναμη, το βάρος και έχει κατεύθυνσή πάντα προς στον αρνητικό ημιάξονα oy' .



Εικόνα 1.1: Το ρομποτικό σύστημα. Πηγή: Homework1 Robotics Systems II

1.2 ΕΞΙΣΩΣΕΙΣ ΣΥΣΤΗΜΑΤΟΣ

Στο διπλανό σχήμα φαίνεται το ρομποτικό σύστημα που θα υλοποιήσουμε σε μία τυχαία θέση. Με χρήση του διπλανού συστήματος θα υπολογίσουμε τις εξισώσεις του μοντέλου.

Θεωρούμε ότι η δύναμη f σχηματίζει γωνία θ με τον κατακόρυφο άξονα y όπως φαίνεται στην Εικόνα 2.

Ακόμη, γνωρίζουμε ότι:

$$\sin(\text{Arctan}(a)) = \frac{a}{\sqrt{1+a^2}}$$

$$\cos(\text{Arctan}(a)) = \frac{1}{\sqrt{1+a^2}}$$

και

$$\theta = \text{Arctan}(x/y)$$

Οπότε:

$$\sin(\theta) = \sin(\text{Arctan}(x/y)) = \frac{\frac{x}{y}}{\sqrt{1+\frac{x^2}{y^2}}} = \frac{x}{\sqrt{x^2+y^2}} = rx$$

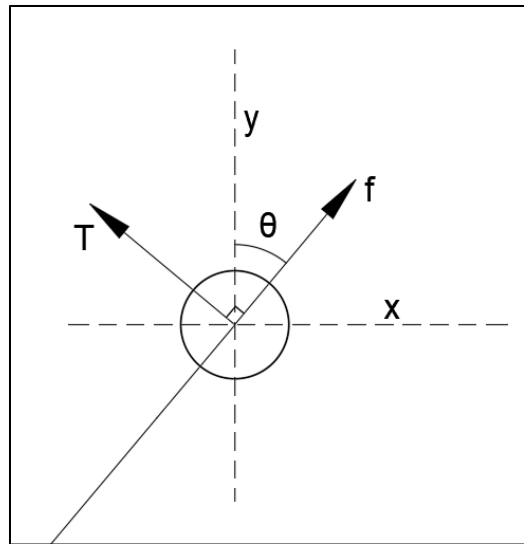
και

$$\cos(\theta) = \cos(\text{Arctan}(x/y)) = \frac{1}{\sqrt{1+\frac{x^2}{y^2}}} = \frac{y}{\sqrt{x^2+y^2}} = ry$$

Επομένως τώρα μπορούμε να υπολογίσουμε τις προβολές των δυνάμεων f και T στους αντίστοιχους άξονες.

Πιο συγκεκριμένα οι δυνάμεις που ασκούνται στον άξονα x που φαίνεται στην Εικόνα 2

$$fx = f \sin \theta \quad , \quad Tx = T \sin(90 - \theta) = T \cos(\theta)$$



Εικόνα 1.2 Ρομποτικό Σύστημα σε μία τυχαία Θέση

Οι δυνάμεις που ασκούνται στον άξονα y που φαίνεται στην Εικόνα 2

$$f_y = f \cos(\theta) \quad , \quad T_y = T \cos(90 - \theta) = T \sin(\theta)$$

Από τον Δεύτερο Νόμο του Νεύτωνα για τον x άξονα προκύπτει:

$$\Sigma f_x = m a \xrightarrow{a=\ddot{x}} f_x + T_x = m \ddot{x} \xrightarrow{\text{Μέτρα}} f \sin(\theta) - T \cos(\theta) = m \ddot{x}$$

$$\xrightarrow{\sin(\theta)=rx, \cos(\theta)=ry} f r_x - T r_y = m \ddot{x} \xrightarrow{\text{ως προς } \ddot{x}} \ddot{x} = \frac{r_x f - r_y T}{m} \quad (1)$$

Από τον Δεύτερο Νόμο του Νεύτωνα για τον y άξονα προκύπτει:

$$\Sigma f_y = m a \xrightarrow{a=\ddot{y}} f_y + T_y + m g = m \ddot{y} \xrightarrow{\text{Μέτρα}} f \cos(\theta) + T \sin(\theta) - m g = m \ddot{y}$$

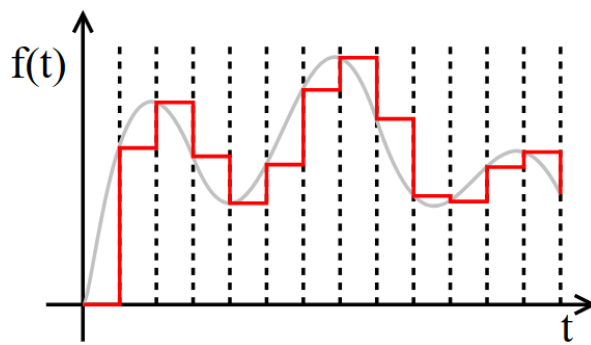
$$\xrightarrow{\sin(\theta)=rx, \cos(\theta)=ry} f r_y + T r_x - m g = m \ddot{y} \xrightarrow{\text{ως προς } \ddot{y}} \ddot{y} = \frac{r_y f + r_x T}{m} - g \quad (2)$$

Οπότε από (1) και (2) υπό μορφή πινάκων και διανυσμάτων οι εξισώσεις του ρομποτικού συστήματος είναι :

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} r_x & -r_y \\ r_y & r_x \end{pmatrix} \begin{pmatrix} f \\ T \end{pmatrix} - \begin{pmatrix} 0 \\ g \end{pmatrix} \quad \text{όπου } r_x = \sin(\theta) = \frac{x}{\sqrt{x^2+y^2}} \quad , \quad r_y = \cos(\theta) = \frac{y}{\sqrt{x^2+y^2}}$$

1.3 ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΟΣ

Διακριτοποίηση συστήματος: Είναι η διαδικασία με την οποία ένα σύστημα συνεχούς χρόνου μεταφράζεται σε σύστημα διακριτού χρόνου. Ανάλογα το πως γίνεται η κβάντωση του χρόνου προκύπτει ή αντίστοιχη διακριτοποίηση του συστήματος. Στην μοντελοποίηση του φυσικού συστήματος θα χρησιμοποιηθεί η Zero on the Hold κβάντωση του χρόνου (ZOH). Η ZOH διακριτοποιεί το σύστημα με ένα σταθερό βήμα χρόνου σε διαδοχικές τιμές της συνεχούς χρόνου συνάρτησης $f(t)$ [Γραφική 1].



Γραφική 1. Πηγή: Wikipedia

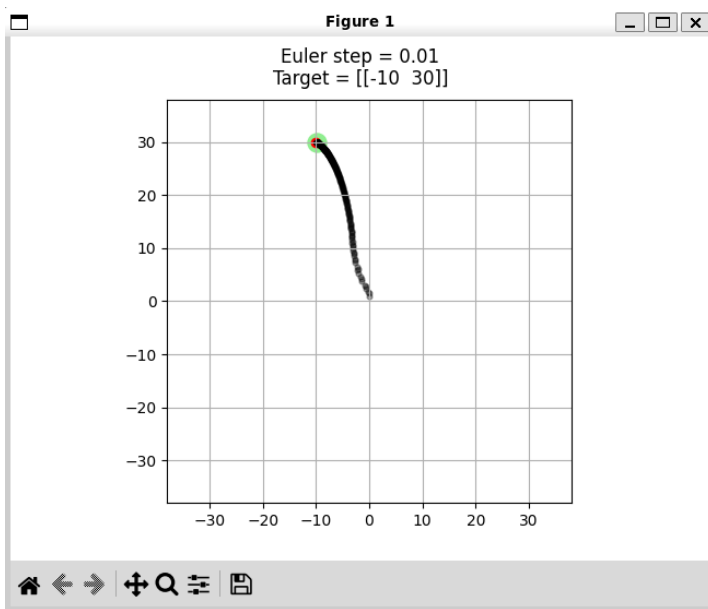
Μέθοδοι Διακριτοποίησης - Ολοκληρωτές: Οι βασικότερες μέθοδοι διακριτοποίησης είναι ο Forward Euler, Runge & Kutta 4th order, Semi-Implicit Euler και η Midpoint. Οι παραπάνω μέθοδοι χρησιμοποιούνται για να υπολογίσουν την επόμενη κατάσταση του συστήματος. Με λίγα λόγια δρουν ως ολοκληρωτές καθώς σε περίπτωση μη-διακριτοποίησης ενός γραμμικού συστήματος θα χρειαζόταν να υπολογίσουμε την χρονική παράγωγο του συστήματος \dot{x} από την γραμμική σχέση $\dot{x} = Ax + Bu$ και έπειτα να ολοκληρώσουμε ώστε να υπολογίσουμε την επόμενη κατάσταση του συστήματος x .

Στόχος Υλοποίησης: Στο πρώτο μέρος της εργασίας γίνεται υλοποίηση και η διακριτοποίηση του συστήματος, η δημιουργία γραφικής απεικόνισης του και η προσέγγιση ενός στόχου με χρήση PD controller.

Περιγραφή Κώδικα Μοντελοποίησης: Το σύστημα έχει υλοποιηθεί με την γλώσσα προγραμματισμού python και συγκεκριμένα την έκδοση 3.10.12. Η υλοποίηση του συστήματος βρίσκεται στο αρχείο με όνομα " Modeling_Part1.py". Στο συγκεκριμένο αρχείο υλοποιούνται 3 κλάσεις με ονόματα HoperRobot, PD_Controller , Visualization. Η HoperRobot υλοποιεί την δυναμική συνάρτηση του συστήματος (dynamics()) καθώς και 4 μεθόδους για την διακριτοποίηση του συστήματος, μια από αυτές τις μεθόδους είναι η euler_discrete(). Η PD_Controller υλοποιεί ένα pd ελεγκτή ο οποίος συγκεκριμένα υλοποιείται στην pd(). Τέλος η Visualization δημιουργεί ένα γραφικό περιβάλλον με την χρήση της βιβλιοθήκης matplotlib.pyplot και χρησιμοποιώντας την συνάρτηση plt.ion() βλέπουμε online την πορεία του συστήματος στον διδιάστατο χώρο. Αναλυτικότερα, στον κατασκευαστή της κλάσης Visualization δημιουργείται το παράθυρο γραφικών αλλά και οι αρχικές συνθήκες του συστήματος και έπειτα με την μέθοδο update() γίνεται ενημέρωση των γραφικών αλλαγών του συστήματος της Εικόνας 1 που προσομοιώνεται.

Κέρδη PD Ελεγκτή: Για την υλοποίηση του PD ελεγκτή, χρησιμοποιήθηκαν μητρώα με διάστασης 2×4 καθώς προσδιορίζουμε διαφορετικό βάρος σε κάθε μεταβολή συνεισφοράς. Πιο συγκεκριμένα ο ελεγκτής που υλοποιείτε για το σύστημα, έχει κέρδη τέτοια ώστε να προσεγγίζει τα σημεία που ικανοποιούν τις σχέσεις: $x < y, x < 0, y > 0$ (I). Δηλαδή προσεγγίζει σημεία τα οποία βρίσκονται στο δεύτερο τεταρτημόριο και πάνω από την ευθεία $y = -x$. Η πρώτη γραμμή των κερδών K_p και K_d αναφέρεται σε μεταβολές που προκαλεί η δύναμη f στην θέση x, y και την ταχύτητα \dot{x}, \dot{y} αντίστοιχα των στηλών τους. Όμοια η δεύτερη γραμμή των κερδών K_p και K_d αναφέρεται σε μεταβολές που προκαλεί η δύναμη T στην θέση x, y και την ταχύτητα \dot{x}, \dot{y} αντίστοιχα των στηλών τους. Επομένως για την προσέγγιση των σημείων της σχέσης (I) είναι λογικό να μας ενδιαφέρουν πιο πολύ οι μεταβολές της δύναμής f αντί της δύναμης T καθώς σύμφωνα με την Εικόνα 2 η γωνία θ θα γίνει αρνητική αρά η συνιστώσα της δύναμης T στον αρνητικό ημιάξονα y θα μεγαλώνει το μέτρο της στις μεταβολές του σφάλματός με αποτέλεσμα να επιβαρύνει την κίνηση προς τον τελικό στόχο. Ωστόσο, η T έχει σχετικά μεγάλα κέρδη για τις y συνιστώσες καθώς λειτουργεί ως αντίβαρο σε περίπτωση απότομης αύξησης του σφάλματος. Τέλος τα κέρδη για την T βοηθούν και στην ομαλή σύγκλιση των θέσεων του συστήματος με τον ίδιο ρυθμό για το χώρο που ορίζουν οι σχέσεις (I) ελαττώνοντας έτσι το σφάλμα κοντά στον τελικό στόχο αποφεύγοντας προβλήματα overshoot.

Παραδείγματα Προσομοιώσεων με αποδεκτό σφάλμα θέσης 0.1



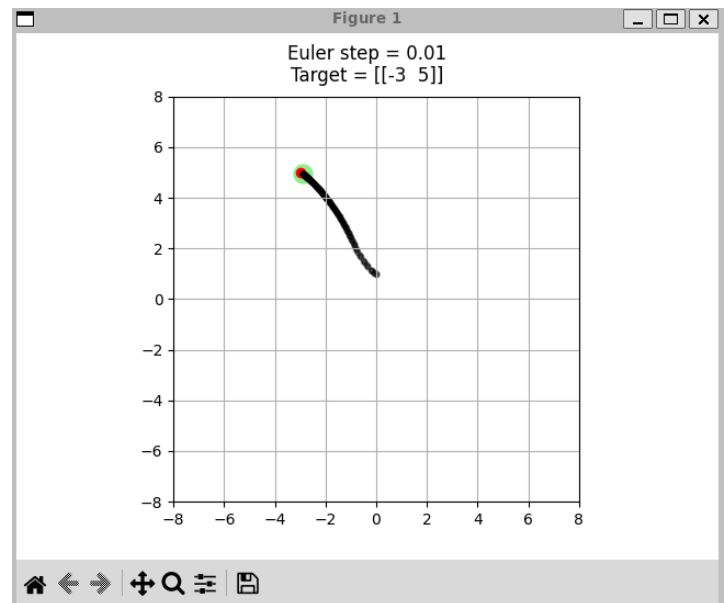
Στην Εικόνα 1.3 για την προσομοίωση της χρειάστηκαν 310 επαναλήψεις τις μεθόδου Euler, με τελική κατάσταση

$[-9.907, 29.976, -0.633, 0.402].T$ και ο στόχος της τελικής κατάστασης

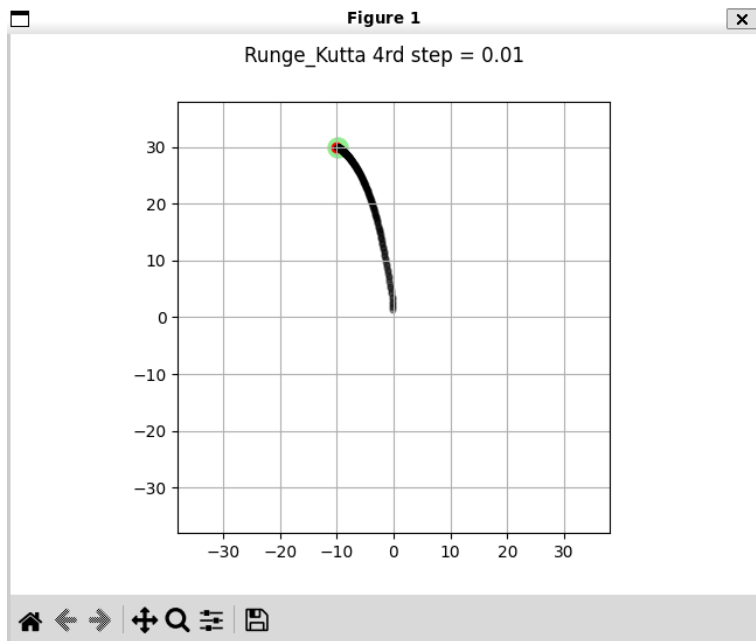
$[-10, 30, 0, 0].T$.

Εικόνα 1.3: Μέθοδος Euler με βήμα διακριτοποίησης 0.01 και στόχος τελικής θέσης (-10,30)

Στην εικόνα 1.4 για την προσομοίωση της χρειάστηκαν 171 επαναλήψεις τις μεθόδου Euler, με τελική κατάσταση $[-2.916, 4.961, -0.702, 0.545].T$ και ο στόχος της τελικής κατάστασης $[-3, 5, 0, 0].T$.



Εικόνα 2.4: Μέθοδος Euler με βήμα διακριτοποίησης 0.01 και στόχος τελικής θέσης (-3,5)



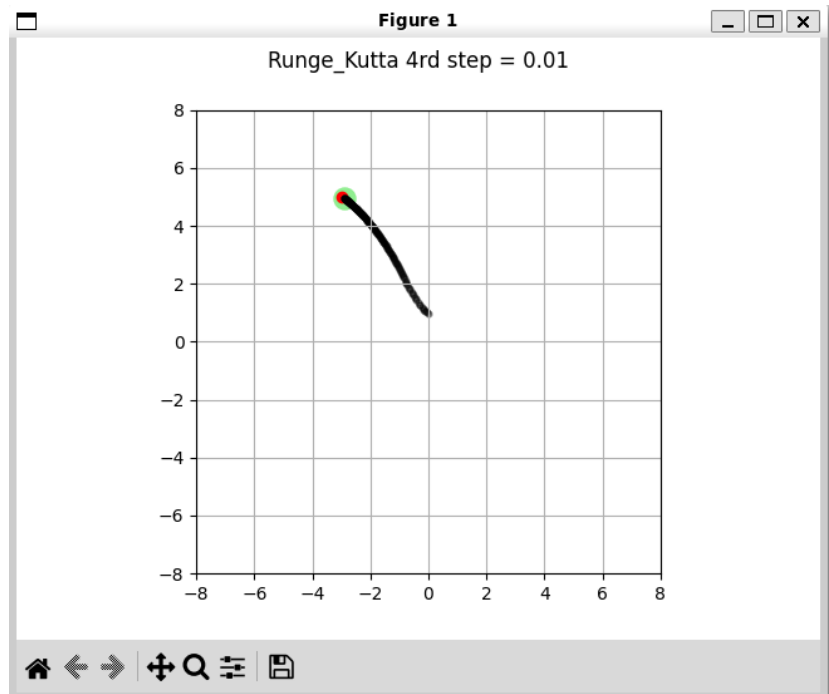
Εικόνα 1.5: Μέθοδος Runge Kutta 4rd με βήμα διακριτοποίησης 0.01 και στόχος τελικής θέσης (-10,30)

Στην εικόνα 1.5 για την προσομοίωση της χρειάστηκαν 327 επαναλήψεις τις μεθόδου Euler, με τελική κατάσταση $[-9.902, 29.990, -0.642, 0.377].T$ και ο στόχος της τελικής κατάστασης $[-10,30,0,0].T$.

Στην εικόνα 1.5 για την προσομοίωση της χρειάστηκαν 327 επαναλήψεις τις μεθόδου Euler, με τελική κατάσταση

$[-2.914, 4.963, -0.706, 0.542].T$ και ο στόχος της τελικής κατάστασης

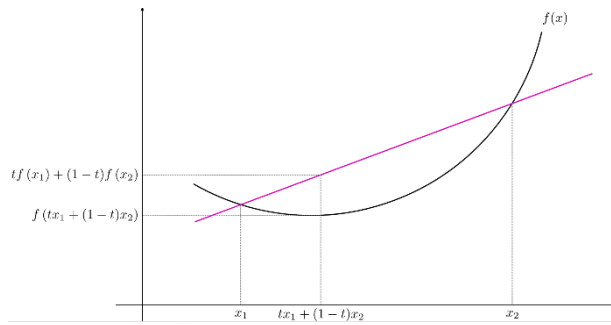
$[-3,5,0,0].T$.



Εικόνα 1.6: Μέθοδος Runge Kutta 4rd με βήμα διακριτοποίησης 0.01 και στόχος τελικής θέσης (-3,5)

2. QP – ADMM

2.1 Τετραγωνικός προγραμματισμός



Γραφική 2. Πηγή: Wikipedia

Μία μαθηματική συνάρτηση με πραγματικές τιμές είναι κυρτή αν κάθε γραμμή μεταξύ δύο οποιοδήποτε διαφορετικών σημείων της συνάρτησης βρίσκεται πάνω από την συνάρτηση για το διάστημα που ορίζουν τα δύο αυτά σημεία.

Ο τετραγωνικός προγραμματισμός είναι μια ειδική περίπτωση κυρτής βελτιστοποίησης, και το τετραγωνικό πρόβλημα ορίζεται ως εξής:

$$f(x) = x^T Q x + q^T x$$

$$Ax - b = 0$$

$$Cx - d \leq 0$$

Το διάνυσμα x και q είναι διάστασης $N \times 1$, ο πίνακας Q είναι διάστασης $N \times N$ και περιγράφει τον τετραγωνικό όρο της αντικειμενικής συνάρτησης. Ο πίνακας A είναι διάστασης $M \times N$ και περιγράφει του ισότικους περιορισμούς του προβλήματος και ο πίνακας C είναι διάστασης $K \times N$ και περιγράφει τους ανισοτικούς περιορισμούς. Αντίστοιχα τα διανύσματα b και d είναι διάστασης $M \times 1$ και $N \times 1$.

Μέσω του υπολογισμού Lagrangian που έχει την παρακάτω μορφή, μπορούμε και υπολογίζουμε τα ελάχιστα της αντικειμενικής συνάρτησης f που ικανοποιούν και τους παραπάνω περιορισμούς.

$$L(x, \lambda, \mu) = f(x) + \lambda^T (Ax - b) + \mu^T (Cx - d)$$

Και οι συνθήκες KKT που πρέπει να ικανοποιούνται είναι:

$$\nabla f + A^T \lambda + C^T \mu = 0 \quad (1)$$

$$Ax - b = 0 \quad (2)$$

$$Cx - d \leq 0 \quad (3)$$

$$\mu \geq 0 \quad (4)$$

$$\mu^T (Cx - d) = 0 \quad (5)$$

Όπου λ και μ είναι οι ισοτικοί και οι ανισοτικοί συντελεστές διάστασης $M \times 1$ και $K \times 1$ αντίστοιχα.

2.2 Quadratic Programming via Alternative Directions Method of Multiplier:

Θεωρούμε ότι το πρόβλημα έχει την μορφή:

$$\frac{1}{2} x^T P x + q^T x$$

με ισοτικούς περιορισμούς:

$$Ax = z$$

$$z \in C$$

Όπου $C = [l, u]$ κυρτό σύνολο.

Αναδιατυπώνοντας QP πρόβλημα χρησιμοποιώντας Penalty όρους το οποίο λαμβάνει την μορφή:

$$\min \quad \frac{1}{2} x^T P x + q^T x + \frac{\sigma}{2} \|x - x^k\|_2^2 + \frac{\rho}{2} \|z - z^k + \rho^{-1} y^k\|_2^2$$

$$Ax = z$$

Όπου σ είναι η penalty παράμετρος για το x , την μεταβλητή του προβλήματος που θέλουμε να ελαχιστοποιήσουμε. Το ρ είναι η penalty παράμετρος για το διάνυσμα του ισοτικού περιορισμού, και y είναι οι συντελεστές της Lagrangian που σχετίζονται με τον ισοτικό περιορισμό, επίσης το $y \in N_C(z)$, που είναι ο κανονικός κώνος του κυρτού συνόλου C .

Αρά υλοποιώντας την (Augmented) Lagrangian όπως ορίζεται στην 2.1 παράγραφο και απαιτώντας να ισχύουν οι KKT συνθήκες, προκύπτει για τις συνθήκες αυτές ότι:

$$\text{Από (1):} \quad Px^{k+1} + q + \sigma (x^{k+1} - x^k) + A^T v^{k+1} = 0$$

που στην ουσία σημαίνει ότι θέλω η πρώτη παράγωγος ως προς x της Lagrangian και ο penalty όρος για το x να είναι μηδέν. Επισημαίνεται ότι: v^{k+1} είναι οι συντελεστές της Lagrangian που σχετίζονται με τον ισοτικό περιορισμό και παράγονται κάθε φορά που λύνεται το σύστημα, ενώ οι y είναι οι παράμετροι που ορίζονται στην αρχή ώστε να έχω καλύτερη σύγκλιση στην λύση του προβλήματος.

Από (2): $Ax^{k+1} - z^{k+1} = 0$

Επιπλέον θα πρέπει οι να τηρούνται τα όρια του ισοτικού περιορισμού καθώς και οι lagragian συντελεστές. Το οποίο μαθηματικά μεταφράζεται:

$$\rho(z^{k+1} - z^k) + (y^k - v^{k+1}) = 0$$

--

Επομένως, παραλείψουμε τον όρο z^{k+1} από την τελευταία σχέση προκύπτει το παρακάτω σύστημα:

$$\begin{pmatrix} P + \sigma I & A^T \\ A & -\rho^{-1}I \end{pmatrix} \begin{pmatrix} x^{k+1} \\ v^{k+1} \end{pmatrix} = \begin{pmatrix} \sigma x^k - q \\ z^k - \rho^{-1}y^k \end{pmatrix}$$

Και

$$z^{k+1} = z^k - \rho^{-1}y^k$$

2.3 Υλοποίηση Αλγόριθμου QP ADMM:

Ο αλγόριθμος που υλοποιεί το πρόβλημα στην παράγραφο 2.2 φαίνεται στην εικόνα 2.1

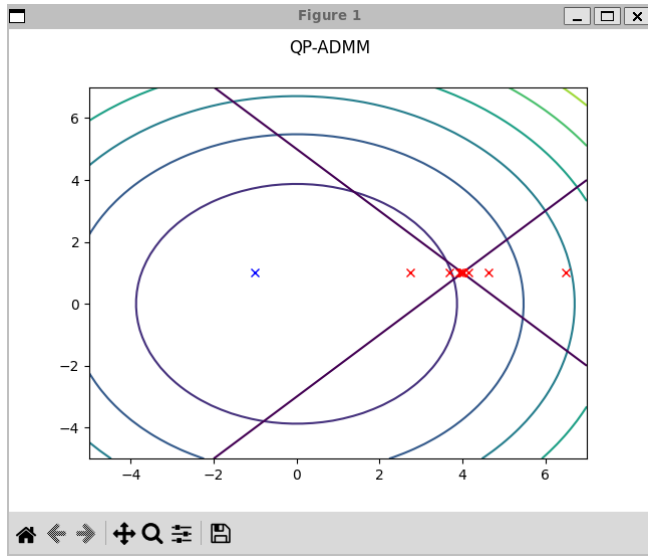
Algorithm 1

- 1: **given** initial values x^0, z^0, y^0 and parameters $\rho > 0, \sigma > 0, \alpha \in (0, 2)$
 - 2: **repeat**
 - 3: $(\tilde{x}^{k+1}, \nu^{k+1}) \leftarrow$ solve linear system $\begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1}I \end{bmatrix} \begin{bmatrix} \tilde{x}^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1}y^k \end{bmatrix}$
 - 4: $\tilde{z}^{k+1} \leftarrow z^k + \rho^{-1}(\nu^{k+1} - y^k)$
 - 5: $x^{k+1} \leftarrow \alpha \tilde{x}^{k+1} + (1 - \alpha)x^k$
 - 6: $z^{k+1} \leftarrow \Pi(\alpha \tilde{z}^{k+1} + (1 - \alpha)z^k + \rho^{-1}y^k)$
 - 7: $y^{k+1} \leftarrow y^k + \rho(\alpha \tilde{z}^{k+1} + (1 - \alpha)z^k - z^{k+1})$
 - 8: **until** termination criterion is satisfied
-

Εικόνα 2.1 : Equality Constrain Quadratic Programming via Alternative Direction Method of Multiplier

Ο Αλγόριθμος 1 υλοποιείτε στο αρχείο “QP_ADMM_Part2.py”. Το αρχείο αυτό αποτελείται από δύο κλάσης την QP_ADMM_LC που υλοποιεί τον Αλγόριθμό 1 στην μέθοδο converge() και η Visualization που χρησιμοποιείται για την γραφική απεικόνιση προβλημάτων με δυο μεταβλητές.

Παραδείγματα Υλοποίησης



```
Solution Found !
=====
xk:
[[3.99844345]
 [0.99900051]]

yk:
[[0.]
 [0.]]

zk:
[[5.]
 [3.]]

vk+1:
[[-4.995005]
 [-2.997003]]

iterations:
10

=====
Now Solving with Proxsuite below
Proxsuite Solution :
[[3.999999 ]
 [0.99999975]]

Comparing the solutions
||xk - x_prox|| 0.0018488381613287112

Should be zero !

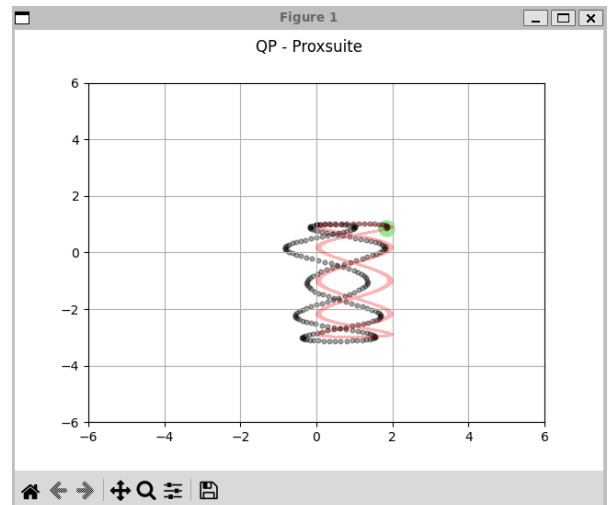
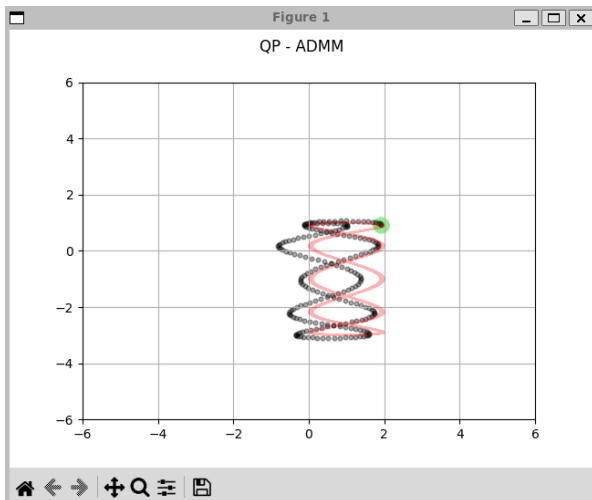
Main has ended with 0
```

3 Control via QP

Παρακάτω παρουσιάζεται η υλοποίηση ελεγκτή με χρήση QP.

Στα αριστερά χρησιμοποιείτε το QP που υλοποιήθηκε στο αρχείο "QP_ADMM_Part2.py" και στα δεξιά χρησιμοποιείτε το QP της βιβλιοθήκης proxsuite

Οι Τροχιά που πρέπει να ακολουθήσει το ρομποτικό σύστημα είναι η κόκκινη γραφική.



Παρατηρήσεις: Ο Έλεγχος του συστήματος γίνεται με τις επιταχύνσεις τους συστήματος καθώς έτσι διασφαλίζεται ότι έχουμε γραμμικούς ισοτικούς περιορισμούς. Η υλοποίηση με την χρήση της βιβλιοθήκης proxsuite είναι αισθητικά πιο γρήγορη. Η προσέγγιση της τροχιάς στην αρχή δεν είναι καλή καθώς αυτό είναι πιθανό να οφείλεται στα μεγάλα κέρδη του ελεγκτή.