

Ρομποτικά Συστήματα II: Εργασία 1

Ονοματεπώνυμο: Βραχωρίτη Αλεξάνδρα

Αριθμός Μητρώου: 1092793

5 Νοεμβρίου 2025

1 Μοντελοποίηση συστήματος ράβδου-μάζας

Για το σύστημα της εκφώνησης υλοποιήθηκε η συνάρτηση `dynamics()` στο αρχείο `1-modeling.ipynb`, που προσομοιώνει τη δυναμική του συστήματος ως:

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \frac{r_x f - r_y \tau}{m} \\ \frac{r_y f + r_x \tau}{m} - g \end{bmatrix} \quad (1)$$

Στη συνέχεια, για τη διακριτοποίηση του συστήματος υλοποιήθηκαν 4 διαφορετικοί integrators: Euler, Semi-Implicit Euler, Runge-Kutta 4th Order, Midpoint, βάσει των οποίων υπολογίζεται το επόμενο διάνυσμα κατάστασης του συστήματος (x_{k+1}), βάσει του προηγούμενου (x_k) και του σήματος ελέγχου ($u = [f \ \tau]^T$). Τα σήματα ελέγχου υπολογίζονται με τη βοήθεια ενός PD-ελεγκτή (συνάρτηση `pd_controller()`), που λειτουργεί ως εξής:

$$\mathbf{F} = K_p \cdot (\mathbf{x}_{desired} - \mathbf{x}) + K_d \cdot (\dot{\mathbf{x}}_{desired} - \dot{\mathbf{x}}) \quad (2)$$

όπου $(\mathbf{x}_{desired} - \mathbf{x})$ είναι το σφάλμα θέσης και $(\dot{\mathbf{x}}_{desired} - \dot{\mathbf{x}})$ είναι το σφάλμα ταχύτητας.

Το \mathbf{F} είναι ένα διάνυσμα διαστάσεων $(2, 1)$ ($F = [F_x \ F_y]^T$), αφού η δύναμη που ασκείται στο ρομπότ αναλύεται σε δύο συνιστώσες στους άξονες x και y αντίστοιχα. Εδώ θα πρέπει να δοθεί προσοχή στον κατακόρυφο άξονα y , όπου στην F_y πρέπει να ληφθεί υπόψη και ο όρος $(m \cdot g)$. Βάσει του συστήματος συντεταγμένων της εικόνας 1 στην εκφώνηση, οι δυνάμεις f και τ αναλύονται στους άξονες ως:

$$f = F_x \cdot \sin(\theta) + F_y \cdot \cos(\theta) \quad (3)$$

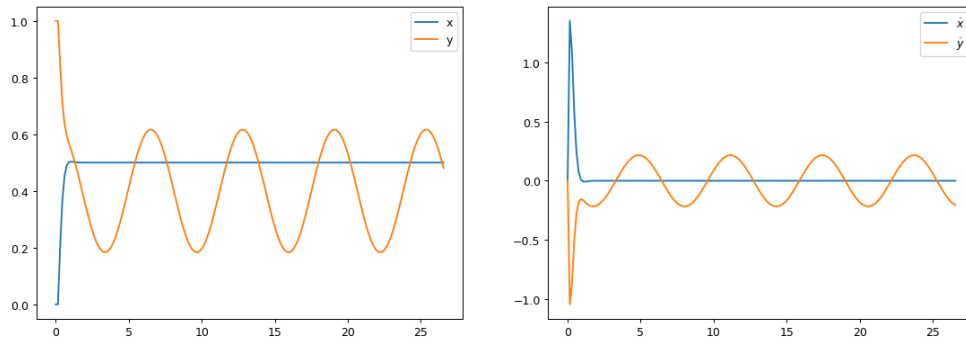
$$\tau = -F_x \cdot \cos(\theta) + F_y \cdot \sin(\theta) \quad (4)$$

όπου $\theta = \tan^{-1}(F_x, F_y)$ η γωνία μεταξύ του κατακόρυφου άξονα y και του διανύσματος της δύναμης f , που ισούται με την γωνία μεταξύ του οριζόντιου άξονα x και του διανύσματος της δύναμης τ .

1.1 Προσομοίωση συστήματος για $dt = 0.15$ και $t = 26.5$

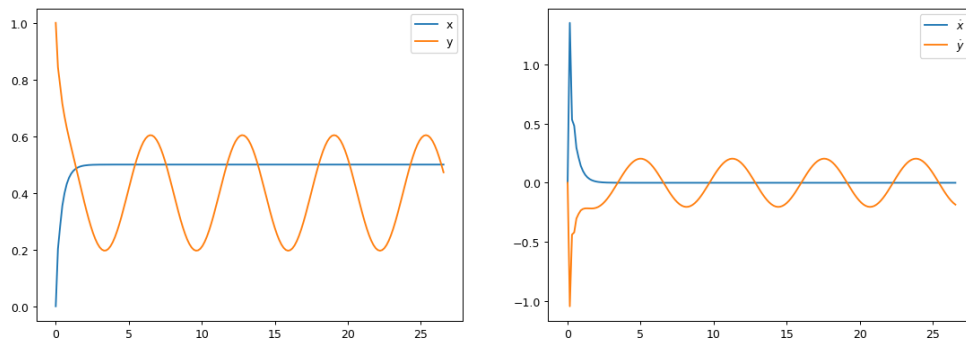
Αφού ορίστηκαν σωστά όλα τα παραπάνω θα δημιουργήσουμε μια ημιτονοειδή τροχιά και θα την εκτελέσουμε για τους διάφορους integrators, εξάγοντας τα αντίστοιχα συμπεράσματα. Η επιθυμητή τροχιά του συστήματος είναι: $\mathbf{x} = [0.5 \ 0.4 + 0.2 \cdot \sin(t) \ 0 \ 0.2 \cdot \cos(t)]^T$, με βήμα $dt = 0.15$ και συνολικό χρόνο προσομοίωσης $t = 26.5$.

1.1.1 Euler integration



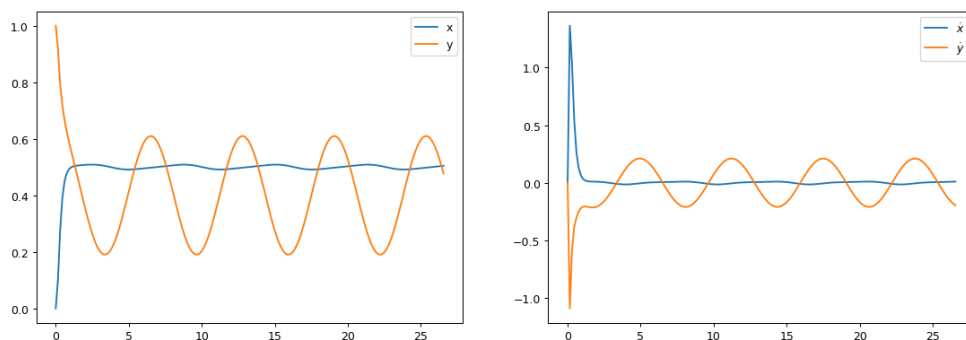
Διάγραμμα 1. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

1.1.2 Semi-Implicit Euler integration



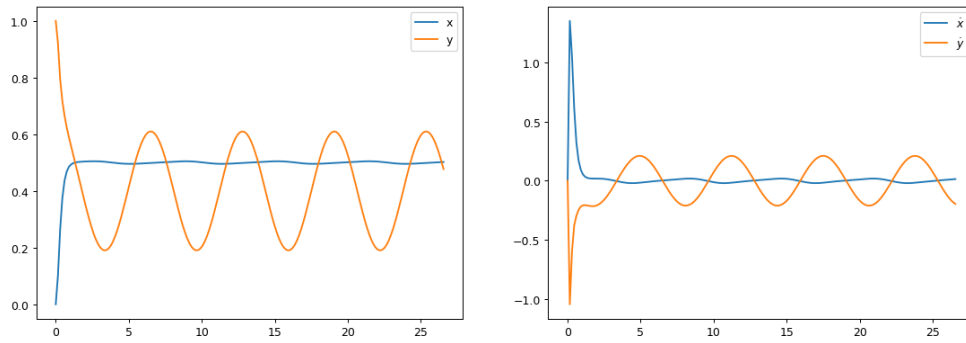
Διάγραμμα 2. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

1.1.3 Runge-Kutta 4th Order integration



Διάγραμμα 3. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

1.1.4 Midpoint integration

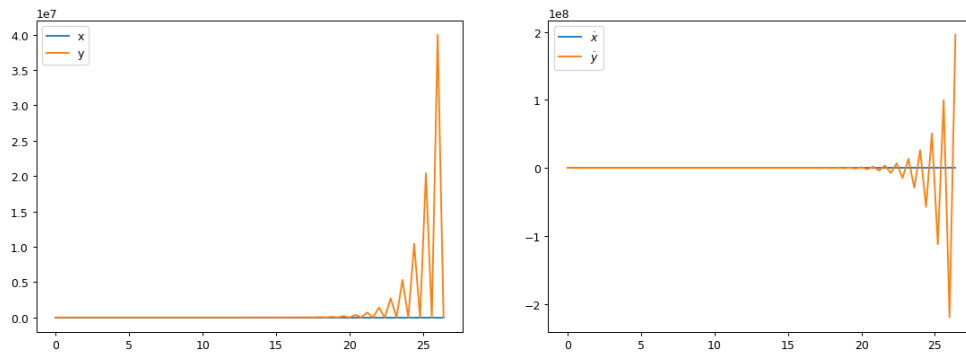


Διάγραμμα 4. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

Παρατηρώντας τα Διαγράμματα 1–4 διαπιστώνουμε πως το σύστημα έχει παρόμοια συμπεριφορά για οποιαδήποτε μέθοδο. Διαφορές εντοπίζονται μόνο στην αρχή της προσομοίωσης, όπου με Euler integration εμφανίζεται μεγαλύτερη υπερέγερση λόγω του αριθμητικού σφάλματος ολοκλήρωσης. Οι μέθοδοι Semi-Implicit Euler, Midpoint και ιδιαίτερα η Runge–Kutta 4th Order φαίνεται να έχουν μεγαλύτερη ακρίβεια στην παρακολούθηση της τροχιάς. Όλα τα παραπάνω οφείλονται στο ότι το βήμα dt είναι μικρό, μόλις 0.15. Ας δοκιμάσουμε να το αυξήσουμε.

1.2 Προσομοίωση συστήματος για $dt = 0.4$ και $t = 26.5$

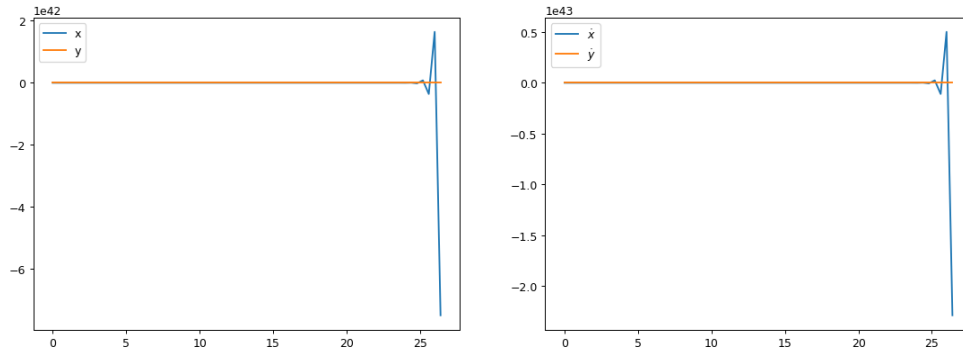
1.2.1 Euler integration



Διάγραμμα 5. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

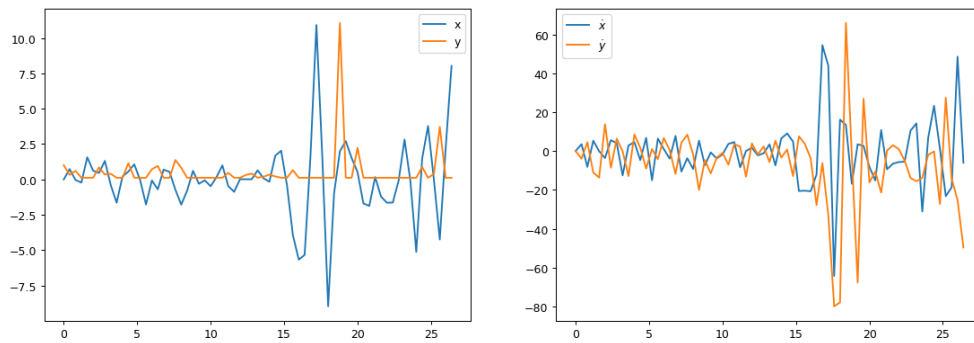
Είναι εμφανές πως ο αλγόριθμος σπάει πολύ γρήγορα με τη χρήση του Euler integration, όπως ήταν αναμενόμενο. Ας εξετάσουμε όμως και τις υπόλοιπες περιπτώσεις, ώστε να διαπιστώσουμε αν κάποιος από αυτούς μπορεί να παρακολουθήσει την τροχιά του συστήματος ικανοποιητικά.

1.2.2 Semi-Implicit Euler integration



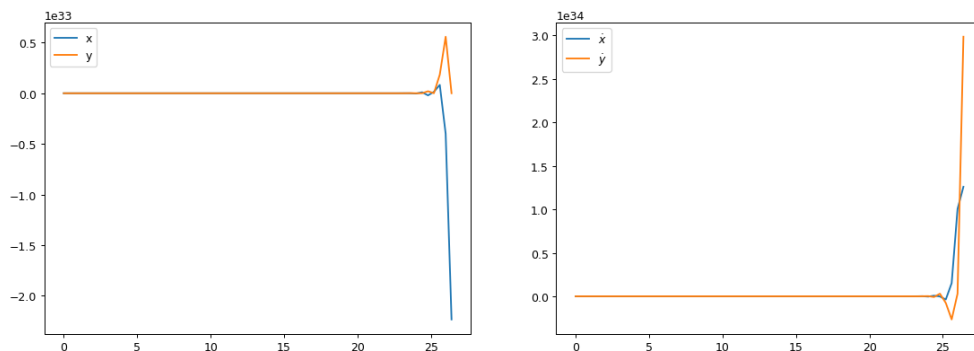
Διάγραμμα 6. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

1.2.3 Runge-Kutta 4th Order integration



Διάγραμμα 7. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

1.2.4 Midpoint integration



Διάγραμμα 8. (αριστερά) Διάγραμμα θέσεων μάζας m , (δεξιά) διάγραμμα ταχυτήτων μάζας m .

Από τα Διαγράμματα 6-8 παρατηρούμε πως μόνο με Runge-Kutta 4th Order integration μπορεί να γίνει παρακολούθηση του ημιτονοειδούς σήματος για περισσότερο χρόνο. Ο Semi-Implicit Euler και ο Midpoint επίσης σπάνε, αλλά πιο αργά σε σχέση με τον απλό Euler.

2 Quadratic Programming μέσω της μεθόδου Primal-Dual Augmented Lagrangian

Η υλοποίηση του αλγορίθμου Primal-Dual ALM βρίσκεται στο αρχείο `2-primal_dual_ALM.ipynb`. Στόχος του αλγορίθμου είναι η εύρεση του σημείου εκείνου που ελαχιστοποιεί την συνάρτηση $f(\mathbf{x})$ και ταυτόχρονα ικανοποιεί του ισοτικών και ανισοτικών περιορισμούς. Έτσι, ορίζουμε:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2} \cdot \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{q}^T \mathbf{x}, \quad (5)$$

με περιορισμούς

$$\begin{aligned} \text{s.t. } \quad & \mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0}, \\ & \mathbf{C} \mathbf{x} - \mathbf{d} \leq \mathbf{0} \end{aligned}$$

Οι ανισοτικοί περιορισμοί μπορούν να μετατραπούν σε ισοτικούς με τη βοήθεια slack μεταβλητών. Έτσι, έχουμε:

$$\mathbf{C} \mathbf{x} + \mathbf{s} - \mathbf{d} = \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0}$$

2.1 Περιγραφή του αλγορίθμου Primal-Dual ALM

Η βασική δομή του αλγορίθμου ακολουθεί τον ψευδοκώδικα της εκφώνησης για την ανανέωση των διανυσμάτων x, s, λ, μ , όπου:

- \mathbf{x} : διάνυσμα μεταβλητών βελτιστοποίησης
- \mathbf{s} : διάνυσμα slack μεταβλητών
- λ : διάνυσμα Lagrange multipliers για τους περιορισμούς ισότητας
- μ : διάνυσμα Lagrange multipliers για τους περιορισμούς ανισότητας

Μετά την ανανέωση των παραπάνω υπολογίζονται, με τη βοήθεια της άπειρης νόρμας, τα residuals των περιορισμών:

- $\|\mathbf{A} \mathbf{x}^{k+1} - \mathbf{b}\|_{\infty}$: το μέγιστο απόλυτο σφάλμα των περιορισμών ισότητας. Αν η τιμή του είναι κοντά στο μηδέν σημαίνει ότι πληρούνται αρκετά ικανοποιητικά οι περιορισμοί ισότητας. Το όριο κάτω από το οποίο είναι ικανοποιητική η προσέγγιση δίνεται από το όρισμα $e_{feas} = 10^{-6}$.
- $\|\mathbf{C} \mathbf{x}^{k+1} + \mathbf{s}^{k+1} - \mathbf{d}\|_{\infty}$: το μέγιστο απόλυτο σφάλμα των περιορισμών ανισότητας. Όμοια με πριν, $e_{feas} = 10^{-6}$.

Ένας επιπλέον έλεγχος βασίζεται στο stationarity:

- $\|\mathbf{Q} \mathbf{x}^{k+1} + \mathbf{q} + \mathbf{A}^T \lambda^{k+1} + \mathbf{C}^T \mu^{k+1}\|_{\infty}$: το μέγιστο απόλυτο σφάλμα στασιμότητας. Όσο πιο κοντά είναι στο μηδέν σημαίνει ότι για τα δεδομένα (λ, μ) η συνάρτηση f ελαχιστοποιείται. Το όριο κάτω από το οποίο είναι ικανοποιητική η προσέγγιση δίνεται από το όρισμα $e_{stat} = 10^{-6}$.

Αν ικανοποιούνται ταυτόχρονα όλοι οι παραπάνω περιορισμοί, τότε η επανάληψη σταματά και παίρνουμε ως αποτέλεσμα τα διανύσματα \mathbf{x} και \mathbf{s} , με το πρώτο να αποτελεί τη βέλτιστη λύση του προβλήματος. Ωστόσο, αυτό δεν αρκεί, καθώς ενδέχεται οι τιμές των residuals να μην βελτιώνονται ή και αν χειροτερεύουν. Επομένως, εντός της επανάληψης χρειάζεται ένας επιπλέον έλεγχος κατά τον οποίο: ορίζουμε έναν μετρητή (counter) και όσο παρατηρείται βελτίωση στα residuals των περιορισμών: counter = 0. Αν υπάρχει στασιμότητα ή χειροτέρευση των τιμών, τότε: counter += 1. Αν ο counter λάβει τιμή ≥ 2 , τότε πρέπει να τροποποιηθεί η τιμή του penalty ρ . Αυτό πρακτικά σημαίνει πως λαμβάνουμε περισσότερο υπόψη τους περιορισμούς στο πρόβλημά μας, αφού το ρ αυξάνεται, δηλαδή μας νοιάζει ακόμη περισσότερο να ικανοποιούνται. Σημαντικό ρόλο στο αποτέλεσμα παίζει η επιλογή του αρχικού ρ καθώς και το τ . Βρέθηκε με τη μέθοδο ρύθμισης (tuning) ότι καλύτερα αποτελέσματα προκύπτουν για $\rho_0 = 0.1$ και $\tau_0 = 10$. Όλα τα παραπάνω έχουν υλοποιηθεί στη συνάρτηση `pd_ALM()`.

2.2 Αποτελέσματα

Παρακάτω παρουσιάζονται τα αποτελέσματα για ένα τυχαίο παράδειγμα βελτιστοποίησης συγκριτικά με τα αποτελέσματα του επιλυτή `proxqp` της βιβλιοθήκης `proxsuite`.

2.2.1 Παράδειγμα 1ο

Με τη βοήθεια της βιβλιοθήκης `random` δημιουργούμε τυχαίους πίνακες:

$$Q = \begin{bmatrix} 4.30546128 & 1.01476915 & 0.72226152 & 0.98011399 \\ 1.01476915 & 2.18910981 & 0.61333992 & -0.78594644 \\ 0.72226152 & 0.61333992 & 2.24473143 & -0.45742741 \\ 0.98011399 & -0.78594644 & -0.45742741 & 1.02774077 \end{bmatrix}, \quad q = \begin{bmatrix} -0.0477225 \\ 0.83901143 \\ 1.06513859 \\ -1.00787443 \end{bmatrix}$$

$$A = \begin{bmatrix} -0.63890917 & 1.41098359 & -0.15862708 & -0.09947622 \end{bmatrix}, \quad b = \begin{bmatrix} -0.15331286 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.96204733 & -1.29430055 & 1.19764887 & 0.39211172 \\ 0.45405141 & -0.54300418 & -0.50573487 & -0.19866273 \end{bmatrix}, \quad d = \begin{bmatrix} -1.71228643 \\ -0.42871353 \end{bmatrix}$$

Ισοτικοί περιορισμοί: 1

Ανισοτικοί περιορισμοί: 2

Μεταβλητές βελτιστοποίησης: 4

Τα αποτελέσματα συνοψίζονται ως εξής:

”pd_ALM()” Function

f(x) = 18.691696642490555

Optimal x = [2.02937103 1.05857528 0.40032928 2.88368216]

Equality constraint : Ax - b = [6.06525176e-07]

Inequality constraint: Cx + s - d = [3.56714044e-07 8.52544095e-07]

”proxqp” Solver

f(x) = 18.691737255004096

Optimal x = [2.02937304 1.05857612 0.40033071 2.88368491]

Equality constraint: Ax - b = [2.23005744e-09]

Όπως φαίνεται από τα αριθμητικά αποτελέσματα, αλλά και από τα Διαγράμματα 9.1 και 9.2, ο αλγόριθμος προσεγγίζει με πολύ μεγάλη ακρίβεια τα αποτελέσματα του επιλυτή, άρα δουλεύει ικανοποιητικά.

3 Έλεγχος μέσω QP

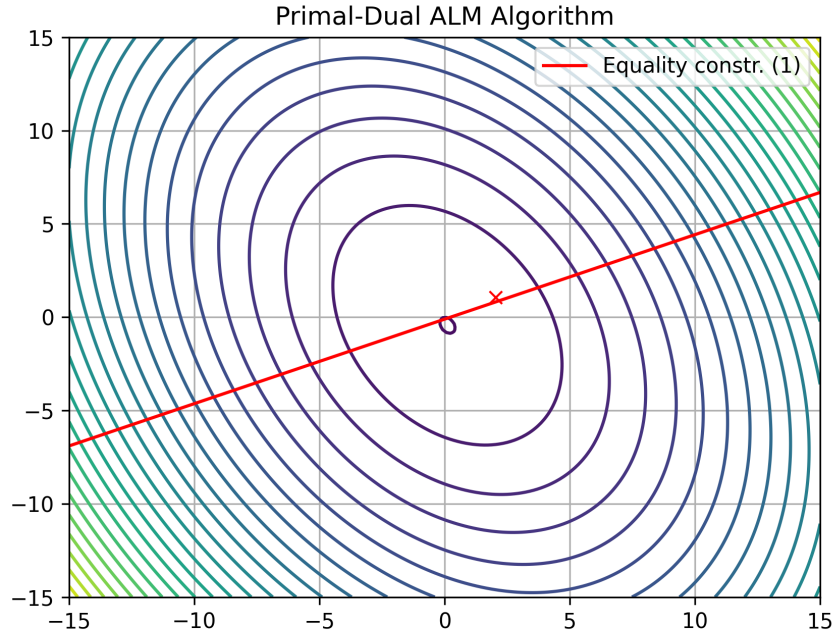
Επιστρέφοντας στο αρχικό μας σύστημα και με δεδομένο ότι οι περιορισμοί πρέπει να είναι γραμμικοί ως προς τις μεταβλητές βελτιστοποίησης, ορίζουμε:

$$\mathbf{z} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ f \\ \tau \end{bmatrix}$$

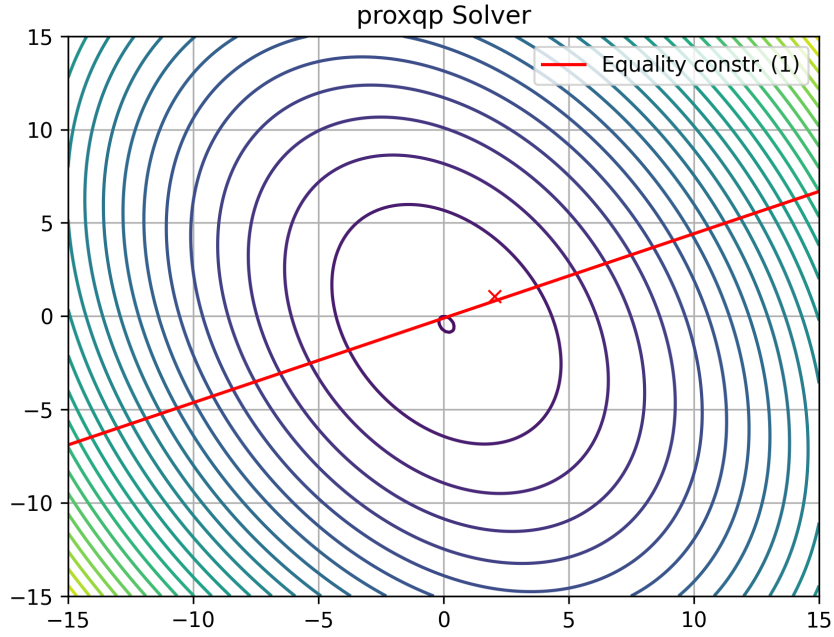
το διάνυσμα βελτιστοποίησης, το οποίο μπορούμε να χωρίσουμε σε δύο μέρη: $\mathbf{a} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}$, το διάνυσμα επιταχύνσεων και $\mathbf{u} = \begin{bmatrix} f \\ \tau \end{bmatrix}$, το διάνυσμα των δυνάμεων. Βάσει αυτών, μπορούμε να ορίσουμε την τετραγωνική συνάρτηση βελτιστοποίησης:

$$f(\mathbf{a}, \mathbf{u}) = \frac{1}{2}(\mathbf{a} - \mathbf{a}_{desired})^T \mathbf{Q}_1 (\mathbf{a} - \mathbf{a}_{desired}) + \frac{1}{2}(\mathbf{u} - \mathbf{u}_{desired})^T \mathbf{Q}_2 (\mathbf{u} - \mathbf{u}_{desired}) \quad (6)$$

και κάνοντας τις επιμεριστικές ιδιότητες έχουμε:



Διάγραμμα 9.1. Στο διάγραμμα απεικονίζεται η συνάρτηση f, ο περιορισμός ισότητας (κόκκινη γραμμή) και η βέλτιστη λύση (κόκκινο x) που προέκυψε από την εφαρμογή του αλγορίθμου Primal-Dual ALM.



Διάγραμμα 9.2. Στο διάγραμμα απεικονίζεται η συνάρτηση f, ο περιορισμός ισότητας (κόκκινη γραμμή) και η βέλτιστη λύση (κόκκινο x) που προέκυψε από τον επιλυτή proxqp.

$$\begin{aligned}
 & \frac{1}{2}(\mathbf{a} - \mathbf{a}_{desired})^T \mathbf{Q}_1 (\mathbf{a} - \mathbf{a}_{desired}) \\
 &= \frac{1}{2} \mathbf{a}^T \mathbf{Q}_1 \mathbf{a} - \frac{1}{2} \mathbf{a}^T \mathbf{Q}_1 \mathbf{a}_{desired} - \frac{1}{2} \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a} + \frac{1}{2} \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a}_{desired} \\
 &= \frac{1}{2} \mathbf{a}^T \mathbf{Q}_1 \mathbf{a} - \mathbf{a}^T \mathbf{Q}_1 \mathbf{a}_{desired} + \frac{1}{2} \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a}_{desired} \\
 &= \frac{1}{2} \mathbf{a}^T \mathbf{Q}_1 \mathbf{a} - \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a} + \frac{1}{2} \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a}_{desired}
 \end{aligned}$$

$$\begin{aligned}
& \frac{1}{2}(\mathbf{u} - \mathbf{u}_{desired})^T \mathbf{Q}_2 (\mathbf{u} - \mathbf{u}_{desired}) \\
&= \frac{1}{2} \mathbf{u}^T \mathbf{Q}_2 \mathbf{u} - \frac{1}{2} \mathbf{u}^T \mathbf{Q}_2 \mathbf{u}_{desired} - \frac{1}{2} \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u}^T + \frac{1}{2} \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u}_{desired}^T \\
&= \frac{1}{2} \mathbf{u}^T \mathbf{Q}_2 \mathbf{u} - \mathbf{u}^T \mathbf{Q}_2 \mathbf{u}_{desired} + \frac{1}{2} \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u}_{desired}^T \\
&= \frac{1}{2} \mathbf{u}^T \mathbf{Q}_2 \mathbf{u} - \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u} + \frac{1}{2} \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u}_{desired}^T
\end{aligned}$$

Οι όροι $\frac{1}{2} \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a}_{desired}^T$, $\frac{1}{2} \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u}_{desired}^T$ είναι σταθεροί, εφόσον τα $\mathbf{a}_{desired}$, $\mathbf{u}_{desired}$ είναι σταθερά και μπορούν να παραλειφθούν, καθώς δεν επηρεάζουν το αποτέλεσμα του αλγορίθμου. Έτσι, έχουμε:

$$\frac{1}{2}(\mathbf{a} - \mathbf{a}_{desired})^T \mathbf{Q}_1 (\mathbf{a} - \mathbf{a}_{desired}) = \frac{1}{2} \mathbf{a}^T \mathbf{Q}_1 \mathbf{a} - \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a}$$

$$\frac{1}{2}(\mathbf{u} - \mathbf{u}_{desired})^T \mathbf{Q}_2 (\mathbf{u} - \mathbf{u}_{desired}) = \frac{1}{2} \mathbf{u}^T \mathbf{Q}_2 \mathbf{u} - \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u}$$

και η συνάρτηση τελικά θα είναι:

$$f(\mathbf{a}, \mathbf{u}) = \frac{1}{2} \mathbf{a}^T \mathbf{Q}_1 \mathbf{a} - \mathbf{a}_{desired}^T \mathbf{Q}_1 \mathbf{a} + \frac{1}{2} \mathbf{u}^T \mathbf{Q}_2 \mathbf{u} - \mathbf{u}_{desired}^T \mathbf{Q}_2 \mathbf{u} \Rightarrow$$

$$f(\mathbf{a}, \mathbf{u}) = \frac{1}{2} \begin{bmatrix} \mathbf{a} \\ \mathbf{u} \end{bmatrix}^T \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0}_2 \\ \mathbf{0}_2 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{u} \end{bmatrix} + \begin{bmatrix} -\mathbf{Q}_1 \mathbf{a}_{desired} \\ -\mathbf{Q}_2 \mathbf{u}_{desired} \end{bmatrix}^T \begin{bmatrix} \mathbf{a} \\ \mathbf{u} \end{bmatrix} \Rightarrow$$

$$f(\mathbf{z}) = \frac{1}{2} \mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{q}^T \mathbf{z}$$

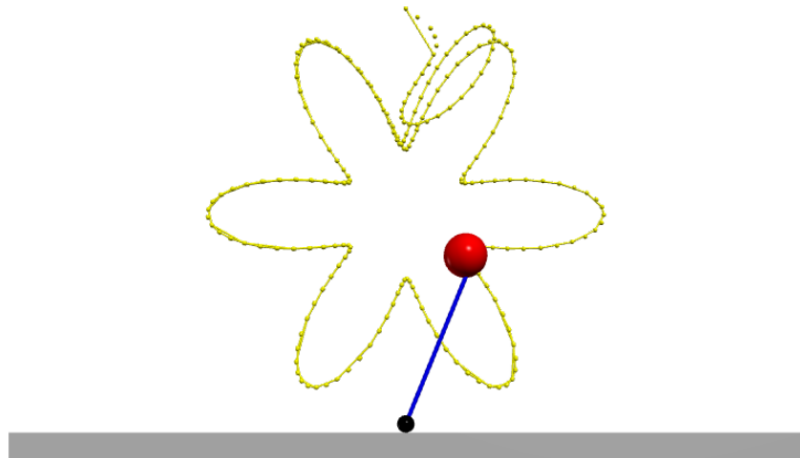
Ορίζοντας σωστά τους \mathbf{Q} και \mathbf{q} βελτιστοποιούμε το \mathbf{z} και χρησιμοποιούμε το διάνυσμα \mathbf{u} ως είσοδος ελέγχου του Runge-Kutta 4th Order integration, ώστε να υπολογίσουμε το διάνυσμα κατάστασης τη στιγμή ($k + 1$). Ας αναλύσουμε, όμως, τα διανύσματα $\mathbf{a}_{desired}$ και $\mathbf{u}_{desired}$. Στην πράξη, αυτό που ορίζουμε είναι το επιθυμητό διάνυσμα κατάστασης $\mathbf{x}_{desired}$, έπειτα παραγωγίζουμε την θέση και παίρνουμε την ταχύτητα και παραγωγίζοντας την ταχύτητα, λαμβάνουμε τη επιτάχυνση. Έτσι, προκύπτει το $\mathbf{a}_{desired}$. Εδώ, θα πρέπει να σημειώσουμε ότι αυτό από μόνο του δεν αρκεί για την σωστή παρακολούθηση της τροχιάς, καθώς είναι απλά ένας feedforward όρος. Θα πρέπει να δημιουργήσουμε και έναν feedback όρο: $K_p \cdot (\mathbf{x}_{desired} - \mathbf{x}) + K_d \cdot (\dot{\mathbf{x}}_{desired} - \dot{\mathbf{x}})$. Τέλος, το $\mathbf{u}_{desired}$ θεωρούμε πως είναι μηδενικό.

Στο αρχείο 3-control_via_QP.ipynb έχουν υλοποιηθεί τρεις τροχιές-χορευτικά, μία από τις οποίες φαίνεται στην παρακάτω εικόνα.

```

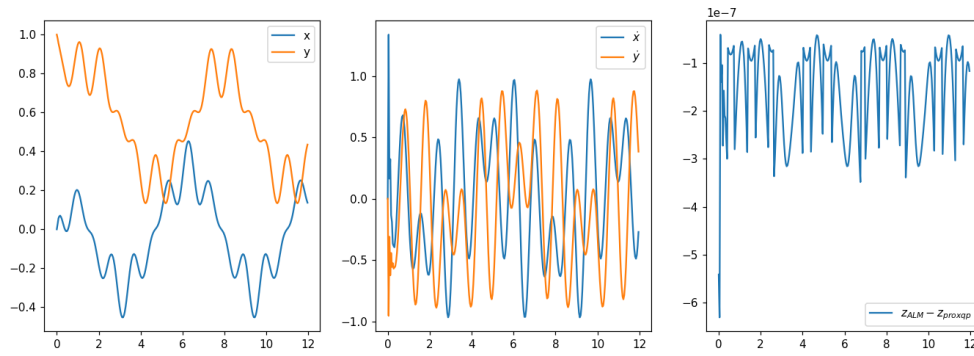
Choose one of the following trajectories:
1. simple
2. heart
3. flower
Answer (full-name): flower
Hopper Robot Vizualization

```



Εικόνα 1. Τροχιά που σχηματίζει λουλούδι.

Εκτός από τα διαγράμματα θέσεων και ταχυτήτων, τυπώνουμε και το διάγραμμα που προκύπτει από τη διαφορά $\mathbf{z}_{ALM} - \mathbf{z}_{proxp}$, ώστε να διαπιστώσουμε ότι πράγματι, οι λύσεις του επιλυτή δε διαφέρουν πρακτικά με τις λύσεις του δικού μας αλγορίθμου.



Διάγραμμα 10. (αριστερά) Διάγραμμα θέσεων μάζας m , (μέση) διάγραμμα ταχυτήτων μάζας m (δεξιά) διάγραμμα απόκλισης της βέλτιστης λύσης του αλγορίθμου ALM από αυτή του επιλυτή $proxqp$.