

# Έλεγχος του ταλαντωτή Van der Pol

Παύλος Τζίτζος

58123

28/12/2023

Καθηγητής:  
Η.Κοσματόπουλος

Μπορείτε να δείτε το repository εδώ.

## Περιεχόμενα

Περιεχόμενα .....	1
Έλεγχος Συστήματος - Εξισώσεις Κατάστασης .....	3
Εφαρμογές .....	3
Μέθοδοι Ελαχιστοποίησης Κλίσης .....	4
Μέθοδος Πεπερασμένων Διαφορών (Finite Differences) .....	4
Στοχαστική Προσέγγιση Ταυτόχρονης Διαταραχής (Simultaneous Perturbation Stochastic Approximation) .....	6
Γραμμικός Τετραγωνικός Ελεγκτής (Linear Quadratic Regulator) .....	7
Μη Άμεσος Προσαρμοστικός Ελεγκτής (In direct Adaptive Controller) .....	9
Κώδικας C++ .....	12
Συναρτήσεις f, fLQR και fAC .....	12
Συναρτήσεις u2, u3, uLQR και uAC .....	12
Συνάρτηση performance και performanceLQR .....	12
Συνάρτηση value .....	13
Συναρτήσεις lqr και riccati2 .....	13
Απαριθμητής algorithms .....	13
Συνάρτηση gradientDescent .....	14
Συνάρτηση main .....	14
Βιβλιοθήκη GNUplot .....	14
Αποτελέσματα .....	15
Μέθοδος Πεπερασμένων Διαφορών (Finite Differences) .....	15
Στοχαστική Προσέγγιση Ταυτόχρονης Διαταραχής (Simultaneous Perturbation Stochastic Approximation) .....	19
Γραμμικός Τετραγωνικός Ελεγκτής (Linear Quadratic Regulator) .....	23
Προσαρμοστικός Ελεγκτής (Adaptive Controller) .....	15
Ανάλυση Ευαισθησίας (Sensitivity Analysis) .....	15
Συμπεράσματα .....	25

[Look great every time].....	16
[Heading 3].....	16
Αναφορές .....	26

## Έλεγχος Συστήματος - Εξισώσεις Κατάστασης

Διαθέτουμε το σύστημα Van der Pol που είναι ένας ταλαντωτής και περιγράφεται από την παρακάτω διαφορική εξίσωση:

$$\frac{d^2x}{dt^2} - c(1 - x^2)\frac{dx}{dt} + x = 0$$

Το  $x$  περιγράφει την θέση και είναι συνάρτηση του χρόνου. Η παράμετρος  $c$  δείχνει την μη-γραμμικότητα του συστήματος.

Στόχος μας είναι να φτιάξουμε έναν ελεγκτή ο οποίος θα παίρνει τα σημεία του  $(x_1, x_2)$  και θα τα φέρνει στο κέντρο των αξόνων  $O(0,0)$ . Ο έλεγχος θα γίνεται με ένα σήμα ελέγχου  $u$  τέτοιο ώστε :

$$m\frac{d^2x}{dt^2} - c(1 - x^2)\frac{dx}{dt} + kx = u$$

Μπορούμε να εκφράσουμε την παραπάνω εξίσωση σε μορφή εξισώσεων κατάστασης θέτωντας ως  $(x_1, x_2) = (x, \dot{x})$  :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{c}{m} \cdot (x_1^2 - 1) \cdot x_2 - \frac{k}{m} \cdot x_1 + \frac{u}{m} \end{bmatrix}$$

Παρακάτω θα δούμε πως μπορούμε να εκφράσουμε το σήμα ελέγχου  $u$  αλλά και μερικές μεθόδους για να ελέγξουμε το σύστημα με τον επιθυμητό τρόπο.

## Εφαρμογές

Ενδεικτικά θα αναφέρουμε μερικές από τις εφαρμογές που έχει ο έλεγχος αυτού του συστήματος:

1. Περιγραφή της μη-γραμμικής συμπεριφοράς στα κυκλώματα ηλεκτρονικών λυχνιών
2. Ελεγχόμενη Ταλάντωση Ηλεκτρικού Κυκλώματος
3. Μοντελοποίηση Βιολογικών Ταλαντωτών

## Μέθοδοι Ελαχιστοποίησης Κλίσης

Μας ενδιαφέρει να πάρουμε τα σημεία και να τα ελέγξουμε. Θέλουμε να ελέγξουμε το σύστημα με τέτοιο τρόπο ώστε να είναι ευσταθές. Για αυτόν τον λόγο θέλουμε μια μέθοδο που θα μας δίνει τέτοιο  $u$  ώστε να είναι το βέλτιστο για το σύστημα.

### Μέθοδος Πεπερασμένων Διαφορών (Finite Differences)

Η μέθοδος πεπερασμένων διαφορών βασίζεται στην μεταβολή που έχει μια παράμετρος ώστε να κατεθύνουμε την τιμή της στην επιθυμητή με το μέγιστο βήμα.

Αρχικά, επιλέγουμε τις παραμέτρους του συστήματος  $(m, c, k)$  και την τιμή των μεταβλητών κατάστασης  $(x_{1,init}, x_{2,init})$ . Το σήμα ελέγχου  $u$  εξαρτάται από τις μεταβλητές κατάστασης  $(x_1, x_2)$  αλλά και από μια παράμετρο  $\vec{\theta}$ . Μπορούμε να την εκφράσουμε αρχικά με διάνυσμα δύο διαστάσεων  $\vec{\theta} = (\theta_1, \theta_2)$  ως το γινόμενο τους:

$$u = u(\vec{\theta}, \vec{x}) = \vec{\theta} \cdot \vec{x} = \theta_1 \cdot x_1 + \theta_2 \cdot x_2$$

Μπορούμε να το εκφράσουμε και με διάνυσμα τριών διαστάσεων  $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$  με λίγο διαφορετικό τρόπο:

$$u = u(\vec{\theta}, \vec{x}) = \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \theta_3 \cdot x_2 \cdot (x_1^2 - 1)$$

Έπειτα υπολογίζουμε το  $\dot{\vec{x}}$ :

$$\dot{\vec{x}} = f(\vec{x}, u) = \begin{bmatrix} x_2 \\ -\frac{c}{m} \cdot (x_1^2 - 1) \cdot x_2 - \frac{k}{m} \cdot x_1 + \frac{u}{m} \end{bmatrix}$$

Και το χρησιμοποιούμε για να υπολογίσουμε την τιμή των μεταβλητών κατάστασης στο επόμενο χρονικό βήμα:

$$\vec{x}_{new} = \vec{x}_{old} + dt \cdot \dot{\vec{x}} = \vec{x}_{old} + dt \cdot f(\vec{x}_{old}, u(\vec{x}_{old}, \vec{\theta}))$$

Επαναλαμβάνουμε τα παραπάνω βήματα μέχρι το τέλος το χρόνου όπως έχουμε ορίσει από την αρχή και υπολογίζουμε την απόδοση  $P$ :

$$P = \sum_{t=0}^{t_{final}} \sqrt{x_{new,t}(1)^2 + x_{new,t}(2)^2}$$

Οι παραπάνω υπολογισμοί συνοψίζονται στα επόμενα βήματα:

Βήμα 1: Υπολογισμός του σήματος ελέγχου.

Βήμα 2: Υπολογίζουμε το  $\dot{\vec{x}}$ .

Βήμα 3: Υπολογίζουμε την νέα τιμή των μεταβλητών κατάστασης.

Βήμα 4: Υπολογίζουμε την απόδοση  $P$ .

Βήμα 5: Επαναλαμβάνουμε τα βήματα μέχρι το κριτήριο τερματισμού.

Η απόδοση αυτή υπολογίστηκε για συγκεκριμένο  $\vec{\theta}$  οπότε τώρα πρέπει να υπολογίσουμε το νέο  $\vec{\theta}$ . Για να το κάνουμε αυτό εκτελούμε τα προηγούμενα βήματα συνολικά 3 φορές. Η πρώτη φορά είναι για  $\vec{\theta} = (\theta_1, \theta_2)$ , η δεύτερη φορά για  $\vec{\theta} = (\theta_1 + \Delta\theta, \theta_2)$  και η τρίτη φορά για  $\vec{\theta} = (\theta_1, \theta_2 + \Delta\theta)$ , ώστε να μπορέσουμε να υπολογίσουμε τα  $\vec{\theta}_{new}$  χρησιμοποιώντας την μέθοδο ελαχιστοποίησης κλίσης:

$$\theta_{1,new} = \theta_{1,old} - \eta \cdot \frac{P(\vec{x}_0, \theta_{1,old} + \Delta\theta, \theta_{2,old}) - P(\vec{x}_0, \theta_{1,old}, \theta_{2,old})}{\Delta\theta}$$

και

$$\theta_{2,new} = \theta_{2,old} - \eta \cdot \frac{P(\vec{x}_0, \theta_{1,old}, \theta_{2,old} + \Delta\theta) - P(\vec{x}_0, \theta_{1,old}, \theta_{2,old})}{\Delta\theta}$$

Σε περίπτωση που χρησιμοποιηθεί το διάνυσμα τριών διαστάσεων θα πρέπει να υπολογιστεί και η απόδοση για τους παρακάτω συνδιασμούς διανυσμάτων  $\vec{\theta}$ :

$$\vec{\theta}_a = (\theta_1, \theta_2, \theta_3)$$

$$\vec{\theta}_b = (\theta_1 + \Delta\theta, \theta_2, \theta_3)$$

$$\vec{\theta}_c = (\theta_1, \theta_2 + \Delta\theta, \theta_3)$$

$$\vec{\theta}_d = (\theta_1, \theta_2, \theta_3 + \Delta\theta)$$

Και για τα καινούργια  $\theta$  θα έχουμε:

$$\theta_{1,new} = \theta_{1,old} - \eta \cdot \frac{P(\vec{x}_0, \vec{\theta}_b) - P(\vec{x}_0, \vec{\theta}_a)}{\Delta\theta}$$

$$\theta_{2,new} = \theta_{2,old} - \eta \cdot \frac{P(\vec{x}_0, \vec{\theta}_c) - P(\vec{x}_0, \vec{\theta}_a)}{\Delta\theta}$$

$$\theta_{3,new} = \theta_{3,old} - \eta \cdot \frac{P(\vec{x}_0, \vec{\theta}_d) - P(\vec{x}_0, \vec{\theta}_a)}{\Delta\theta}$$

Επαναλαμβάνουμε την παραπάνω διαδικασία μέχρι να ισχύει κάποιο κριτήριο τερματισμού. Στην υλοποίηση το κριτήριο τερματισμού είναι η τιμή της απόδοσης να γίνει αρκετά μεγάλη ή να γίνει ένα μέγιστο πλήθος επαναλήψεων.

## Στοχαστική Προσέγγιση Ταυτόχρονης Διαταραχής (Simultaneous Perturbation Stochastic Approximation)

Η μέθοδος στοχαστικής προσέγγισης ταυτόχρονης διαταραχής (ΣΠΤΔ) βασίζεται στην κατανομή bernoulli για τις παραμέτρους επιλέγοντας με τυχαίο τρόπο την τιμή της  $bernoulli(p)$  όπου  $p$  ένας τυχαίος αριθμός 0 έως 1.

Αρχικά, σύμφωνα με την αναφορά 2 χρειάζεται πρώτα να υπολογίσουμε τους συντελεστές  $c_k$ ,  $a_k$ ,  $\Delta_k$ , όπου  $k = 0, 1, 2, \dots$  και είναι οι επαναλήψεις ολόκληρης της μεθόδου. Τον πρώτο συντελεστή τον υπολογίζουμε από τον τύπο:

$$c_k = \frac{c}{k^\gamma}$$

όπου  $c$  και  $\gamma$  σταθερές

Τον δεύτερο συντελεστή από τον τύπο:

$$a_k = \frac{a}{(A + k)^\beta}$$

όπου  $a$ ,  $A$  και  $\beta$  σταθερές

Ο πίνακας  $\Delta_k$  είναι τιμές της κατανομής bernoulli οι οποίες επιλέγονται από την πιθανότητα  $p$  η οποία σε κάθε επανάληψη είναι τυχαία. Η διάσταση του πίνακα  $\Delta_k$  είναι αντίστοιχη του πλήθους  $\theta$  παραμέτρων. Οπότε για δύο  $\theta$  παραμέτρους το  $\Delta_k$  έχει δύο στοιχεία.

Η μεθοδολογία στο σημείο αυτό είναι ίδια με των πεπερασμένων διαφορών με τα κύρια σημεία να είναι:

Βήμα 1: Υπολογισμός του σήματος ελέγχου  $u$  με την ίδια συνάρτηση και τις παραμέτρους του συστήματος.

Βήμα 2: Υπολογίζουμε το  $\dot{x}$  με τον ίδιο τύπο.

Βήμα 3: Υπολογίζουμε την νέα τιμή των μεταβλητών κατάστασης.

Βήμα 4: Υπολογίζουμε την απόδοση  $P$  με τον ίδιο τύπο.

Βήμα 5: Επαναλαμβάνουμε τα βήματα μέχρι το κριτήριο τερματισμού.

Η διαφορά των δύο αυτών μεθόδων είναι ο υπολογισμός των νέων  $\theta$ :

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \cdot \hat{g}_k(\hat{\theta}_k)$$

Όπου:

$$\hat{g}_k(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k \Delta_k) - y(\hat{\theta}_k - c_k \Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k,1}^{-1} \\ \Delta_{k,2}^{-1} \\ \vdots \\ \Delta_{k,p}^{-1} \end{bmatrix}$$

Έχουμε μια συνάρτηση απωλειών την  $y = L(\cdot)$  που για εμάς είναι τα παραπάνω βήματα, οπότε για δύο  $\vec{\theta} = (\theta_1, \theta_2)$  εκτελούμε τα παραπάνω βήματα τόσες φορές όσες χρειάζεται για κάθε  $\theta$ , δηλαδή 2 φορές για το  $\theta_1$  και 2 για το  $\theta_2$ . Για 3  $\theta$  θα έχουμε 2 για κάθε ένα, συνολικά 6 φορές θα εφαρμόσουμε τα παραπάνω βήματα.

Το κριτήριο τερματισμού είναι η μεταβολή των  $\theta$  παραμέτρων να είναι μικρότερη από μια μικρή τιμή.

## Γραμμικός Τετραγωνικός Ελεγκτής (Linear Quadratic Regulator)

Η μέθοδος γραμμικού τετραγωνικού ελεγκτή (ΓΤΕ) βασίζεται στην επίλυση της αλγεβρικής εξίσωσης Riccati και το σύστημα στο οποίο υπόκειται είναι της μορφής  $\dot{\vec{x}} = \vec{A} \cdot \vec{x} + \vec{B} \cdot \vec{u}$ .

Για να χρησιμοποιήσουμε την μέθοδο πρέπει να γραμμικοποιήσουμε το σύστημά μας. Μέχρι τώρα είχαμε:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{c}{m} \cdot (x_1^2 - 1) \cdot x_2 - \frac{k}{m} \cdot x_1 + \frac{u}{m} \end{bmatrix}$$

Όμως μπορεί να ξαναγραφτεί ως:

$$\dot{\vec{x}} = \vec{A} \cdot \vec{x}$$

Για να προκύψει αυτό κάνουμε την παρακάτω ανάλυση.

Θεωρούμε ότι το σήμα ελέγχου  $u$  είναι:

$$u = m \cdot (\theta_1 \cdot (x_1^2 - 1) \cdot x_2 + \theta_2 \cdot x_1 + k_1 \cdot x_1 + k_2 \cdot x_2)$$

Θεωρούμε επίσης ότι  $\theta_1 = \frac{c}{m}$  και  $\theta_2 = \frac{k}{m}$  και  $k_1, k_2$  είναι παράμετροι.

Έτσι αντικαθιστώντας στο σύστημα προκύπτει ότι :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ k_1 & k_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \dot{\vec{x}} = A \cdot \vec{x}$$

Πρέπει να ισχύει ότι:

$$A^T P + P A = -Q = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Λύνοντας ως προς τα στοιχεία του πίνακα  $P$  :

$$P = \begin{bmatrix} p_1 & p_2 \\ p_3 & p_4 \end{bmatrix}$$

Προκύπτει ότι για  $k_1 \neq 0$  και  $k_2 \neq 0$  :

$$p_1 = \frac{k_2}{2k_1} - \frac{k_1 + 1}{2k_2}$$

$$p_3 = p_2 = -\frac{1}{2k_1}$$
$$p_4 = \frac{1 - k_1}{2k_1k_2}$$

Ξαναμετατρέπουμε το σύστημα αλλά αυτή την φορά έχουμε :

$$\bar{u} = \vec{K} \cdot \vec{x} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = k_1x_1 + k_2x_2$$

Οπότε το σύστημα είναι:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \bar{A} \cdot \vec{x} + \bar{B} \cdot \bar{u} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \bar{u}$$

Τώρα είμαστε σε θέση να εφαρμόσουμε την μέθοδο ΓΤΕ στο παραπάνω σύστημα.

Εφόσον καθορίσουμε τις αρχικές τιμές των μεταβλητών κατάστασης και καθορίσουμε τους πίνακες  $Q$  και  $R$  το πρώτο πράγμα που κάνουμε είναι να λύσουμε την εξίσωση Riccati:

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$$

Η λύση της εξίσωσης είναι ο πίνακας  $P$  και μας χρησιμεύει ώστε να υπολογίσουμε το διάνυσμα  $K$  από τον τύπο:

$$K = (R + B^T P B)^{-1} \cdot B^T P A$$

Έτσι μπορούμε να υπολογίσουμε τις τιμές των μεταβλητών κατάστασης και την απόδοση όμως κάναμε στις προηγούμενες μεθοδολογίες.

Συνοψίζοντας τα βήματα αυτού του αλγορίθμου είναι:

Βήμα 1: Λύση Εξίσωσης Riccati.

Βήμα 2: Υπολογισμός του πίνακα  $K$ .

Βήμα 3: Υπολογισμός νέας τιμής των μεταβλητών κατάστασης

Βήμα 4: Υπολογισμός Απόδοσης

Βήμα 5: Έλεγχος Κριτηρίου Τερματισμού



## Μη Άμεσος Προσαρμοστικός Ελεγκτής (In direct Adaptive Controller)

Η μέθοδος μη άμεσου προσαρμοστικού ελέγχου (ΠΕ) βασίζεται στην εξάλειψη των παραμέτρων και στην χρήση προσεγγίσεων για τις παραμέτρους του συστήματος και όχι τις πραγματικές τιμές. Οπότε πρέπει να ορίσουμε το σφάλμα αυτών των προσεγγίσεων.

Θεωρούμε ότι  $\theta_1 = \frac{c}{m}$  και  $\theta_2 = \frac{k}{m}$ .

Ορίζουμε τα σφάλματα των παραμέτρων  $\theta$ :

$$\tilde{\theta}_1 = \theta_1 - \hat{\theta}_1$$

$$\tilde{\theta}_2 = \theta_2 - \hat{\theta}_2$$

Πρέπει να εκφράσουμε το σύστημα μας με τα σφάλματα. Οπότε όπως και στην προηγούμενη μέθοδο ΓΤΕ μετατρέπουμε το σύστημα θεωρώντας ότι το σήμα ελέγχου είναι:

$$u(\vec{x}, \vec{\hat{\theta}}) = \hat{\theta}_1 \cdot x_2 \cdot (x_1 - 1)^2 + \hat{\theta}_2 \cdot x_1 + k_1 \cdot x_1 + k_2 \cdot x_2$$

Το σύστημα μέχρι τώρα ήταν:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{c}{m} \cdot (x_1^2 - 1) \cdot x_2 - \frac{k}{m} \cdot x_1 + \frac{u}{m} \end{bmatrix}$$

Οπότε αντικαθιστώντας καταλήγουμε στο:

$$\dot{\vec{x}} = A \cdot \vec{x} + \tilde{A} = \begin{bmatrix} 0 & 1 \\ k_1 & k_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{\theta}_1 x_2 (x_1^2 - 1) + \tilde{\theta}_2 x_1 \end{bmatrix}$$

Παίρνοντας την συνάρτηση Lyapunov ως:

$$V = x^T P x$$

Την παραγωγίζουμε και παίρνουμε:

$$\dot{V} = (x^T P x)' = \dot{x}^T P x + x^T P \dot{x} = (x^T A^T + \tilde{A}^T) P x + x^T P (A x + \tilde{A}) = x^T A^T P x + \tilde{A}^T P x + x^T P A x + x^T P \tilde{A} = x^T (-Q) x + \tilde{A}^T P x + x^T P \tilde{A}$$

Το οποίο μετά από αντικατάσταση καταλήγει σε:

$$\dot{V} = -x_1^2 - x_2^2 + \tilde{\theta}_1 \cdot x_2 (x_1^2 - 1) \cdot \left( x_2 \frac{1 - k_1}{k_1 k_2} - \frac{x_1}{k_1} \right) + \tilde{\theta}_2 \cdot x_1 \cdot \left( x_2 \frac{1 - k_1}{k_1 k_2} - \frac{x_1}{k_1} \right)$$

Θεωρώντας ότι:

$$A_1 = x_2 \cdot (x_1^2 - 1) \cdot \left( x_2 \frac{1 - k_1}{k_1 k_2} - \frac{x_1}{k_1} \right)$$

$$A_2 = x_1 \cdot \left( x_2 \frac{1 - k_1}{k_1 k_2} - \frac{x_1}{k_1} \right)$$

Η  $\dot{V}$  γίνεται:

$$\dot{V} = -x_1^2 - x_2^2 + A_1\tilde{\theta}_1 + A_2\tilde{\theta}_2$$

Θεωρούμε τώρα την συνάρτηση  $W$  τέτοια ώστε όταν την παραγωγίσουμε να δείξουμε ότι είναι συνάρτηση Lyapunov.

Έστω ότι είναι :

$$W = V + \frac{1}{2\gamma}\tilde{\theta}_1^2 + \frac{1}{2\gamma}\tilde{\theta}_2^2$$

Τότε η παράγωγος θα είναι:

$$\dot{W} = \dot{V} + \frac{1}{\gamma}\tilde{\theta}_1\dot{\tilde{\theta}}_1 + \frac{1}{\gamma}\tilde{\theta}_2\dot{\tilde{\theta}}_2$$

Θέλουμε όμως να ισχύει ότι  $\dot{W} \leq 0$ . Οπότε για κάθε όρο έχουμε:

$$-x_1^2 - x_2^2 = 0 \Rightarrow x_1 = x_2 = 0$$

$$\tilde{\theta}_1(A_1 + \frac{1}{\gamma}\dot{\tilde{\theta}}_1) = 0 \Rightarrow \dot{\tilde{\theta}}_1 = -\gamma A_1 \Rightarrow \dot{\hat{\theta}}_1 = \gamma A_1$$

$$\tilde{\theta}_2(A_2 + \frac{1}{\gamma}\dot{\tilde{\theta}}_2) = 0 \Rightarrow \dot{\tilde{\theta}}_2 = -\gamma A_2 \Rightarrow \dot{\hat{\theta}}_2 = \gamma A_2$$

Επειδή όμως γνωρίζουμε τις προσεγγίσεις  $\hat{\theta}$  μας διευκολύνουν οι παραπάνω σχέσεις γιατί μπορούμε να τις μεταβάλουμε με την μέθοδο gradient descent. Οι τύποι παρουσιάζονται παρακάτω:

$$\hat{\theta}_{1,new} = \hat{\theta}_{1,old} - d\theta \cdot \dot{\hat{\theta}}_1 = \hat{\theta}_{1,old} - d\theta \cdot \gamma \cdot A_1$$

$$\hat{\theta}_{2,new} = \hat{\theta}_{2,old} - d\theta \cdot \dot{\hat{\theta}}_2 = \hat{\theta}_{2,old} - d\theta \cdot \gamma \cdot A_2$$

Αν πάμε να κατασκευάσουμε έναν αλγόριθμο αυτής της μεθόδου θα έχουμε τα παρακάτω βήματα:

Βήμα 1: Υπολόγισε τα σφάλματα των παραμέτρων  $\theta$ .

Βήμα 2: Υπολόγισε τα  $A_1$ ,  $A_2$ .

Βήμα 3: Υπολόγισε την  $\dot{W}$ .

Βήμα 4: Υπολόγισε την  $\dot{\hat{\theta}}$ .

Βήμα 5: Υπολόγισε τις νέες τιμές των μεταβλητών κατάστασης χρησιμοποιώντας το νέο σύστημα  $\dot{\hat{x}} = A \cdot \hat{x} + \tilde{A}$ .

Βήμα 6: Επανέλαβε τα βήματα 2 έως 5 μέχρι να γίνει  $\dot{W} = 0$ .

Βήμα 7: Υπολόγισε να νέα  $\hat{\theta}$ .

Βήμα 8: Έλεγχε το κριτήριο τερματισμού. Αν ισχύει σταμάτα, αλλιώς πήγαινε στο Βήμα 1.



## Κώδικας C++

Στην παράγραφο αυτή θα γίνει μια συνοπτική εξήγηση των συναρτήσεων που έχουν υλοποιηθεί.

### Συναρτήσεις $f$ , $fLQR$ και $fAC$

Οι συναρτήσεις αυτές περιλαμβάνουν τα συστήματα όπως έχουν προκύψει κατά την ανάλυση στην προηγούμενη παράγραφο.

Η συνάρτηση  $f$  είναι το αρχικό σύστημα υλοποιημένο με τις παραμέτρους του συστήματος  $(m, c, k)$  και χρησιμοποιείται από την μέθοδο Πεπερασμένων Διαφορών και την SPSA.

Η συνάρτηση  $fLQR$  αφορά αποκλειστικά την μέθοδο LQR και δεν περιέχει τις παραμέτρους  $(m, c, k)$ .

Η συνάρτηση  $fAC$  αφορά αποκλειστικά την μέθοδο Προσαρμοστικού Ελέγχου και δεν περιέχει τις παραμέτρους του συστήματος  $(m, c, k)$ . Περιλαμβάνει τις εκτιμήσεις των παραμέτρων  $\theta, \hat{\theta}_1$  και  $\hat{\theta}_2$ .

### Συναρτήσεις $u2$ , $u3$ , $uLQR$ και $uAC$

Οι συναρτήσεις αυτές υλοποιούν το σήμα ελέγχου του συστήματος όπως επιλέχθηκαν στην προηγούμενη παράγραφο.

Η συνάρτηση  $u2$  και η  $u3$  αφορούν τις μεθόδους Πεπερασμένων Διαφορών και SPSA και η διαφορά τους είναι μόνο το πλήθος των παραμέτρων  $\theta$ . Αντίστοιχα δύο και τρεις παράμετροι.

Η συνάρτηση  $uLQR$  αφορά μόνο την μέθοδο LQR και είναι το γινόμενο των των διανυσμάτων  $K$  και  $x$ .

Η συνάρτηση  $uAC$  αφορά την μέθοδο Προσαρμοστικού Ελέγχου και περιλαμβάνει τις εκτιμήσεις των παραμέτρων  $\theta$ , τις σταθερές  $k_1, k_2$ , τις τιμές των μεταβλητών κατάστασης  $\vec{x}$  και την παράμετρο του συστήματος  $m$ .

### Συνάρτηση *performance* και *performanceLQR*

Όταν στην μέθοδο πεπερασμένων διαφορών εισάχθηκε η απόδοση ως μετρική, στην υλοποίηση ήταν μια συνάρτηση που παίρνει ως είσοδο τα παρακάτω:

1. Αρχικές τιμές των μεταβλητών κατάστασης.
2. Παράμετροι  $\theta$ .
3. Επιλογή πλήθους  $\theta$  παραμέτρων.
4. Παράμετροι Συστήματος.
5. Όνομα αρχείου για αποθήκευση των δεδομένων.

Η έξοδος της συνάρτησης αυτής είναι η τιμή της απόδοσης που υπολογίζεται από τους τύπους που έχουν αναφερθεί στην προηγούμενη παράγραφο.

Η συνάρτηση *performance* καλείται από την μέθοδο Πεπερασμένων Διαφορών και την SPSA ενώ χρησιμοποιεί τις συναρτήσεις  $f$ ,  $u2$  και  $u3$ .

Η συνάρτηση *performanceLQR* καλείται από την μέθοδο LQR, εκτελεί την ίδια δουλειά με την μόνη διαφορά ότι οι παράμετροι  $\theta$  αντικαταστήθηκαν από τις  $K$  και ότι δεν χρειάζονται οι παράμετροι του συστήματος  $(m, c, k)$ . Επίσης αντί για να καλεί τις συναρτήσεις  $f$ ,  $u2$ ,  $u3$  καλεί τις αντίστοιχες  $fLQR$  και  $uLQR$ .

## Συνάρτηση *value*

Αντίστοιχη υλοποίηση της *performance* για την μέθοδο Προσαρμοστικού Ελέγχου είναι η συνάρτηση *value*. Η συνάρτηση αυτή δέχεται ως είσοδο:

1. Αρχικές τιμές των μεταβλητών κατάστασης.
2. Τις εκτιμήσεις των παραμέτρων  $\theta$ ,  $\hat{\theta}$ .
3. Τις πραγματικές τιμές των παραμέτρων  $\theta$ .
4. Την παράγωγο του σφάλματος των παραμέτρων  $\theta$ ,  $\dot{\hat{\theta}}$ .
5. Την παράμετρο του συστήματος  $m$ .

Η συνάρτηση αυτή καλεί τις αντίστοιχες συναρτήσεις  $fAC$  και  $uAC$  και επιστρέφει τις τιμές των συντελεστών  $A_1$  και  $A_2$ .

## Συναρτήσεις *lqr* και *riccati2*

Για την επίλυση της Αλγεβρικής Εξίσωσης Riccati είναι υπεύθυνη η συνάρτηση *riccati2* και η υλοποίησή της αφορά πίνακες  $2 \times 2$ . Η συνάρτηση αυτή παίρνει ως είσοδο τους πίνακες  $A$ ,  $B$ ,  $Q$ ,  $Q_f$ ,  $R$ . Υπολογίζει τον νέο πίνακα  $P$  και τον επιστρέφει.

Η συνάρτηση *lqr* χρησιμοποιεί την *riccati2* για να υλοποιήσει το βήμα αυτό του αλγορίθμου αναδρομικά. Έχει ως είσοδο το αρχικό διάνυσμα των μεταβλητών κατάστασης και τα στοιχεία  $q$ ,  $r$ . Επιστρέφει τις παραμέτρους  $k_1$  και  $k_2$  του διανύσματος  $K$ .

## Απαριθμητής *algorithms*

Ο απαριθμητής *algorithms* χρησιμοποιείται για την επιλογή αλγορίθμου κατά την εκτέλεση. Αυτή την στιγμή είναι διαθέσιμες και οι 4 μέθοδοι που έχουν αναφερθεί στην προηγούμενη παράγραφο:

1. FD2: Πεπερασμένες Διαφορές με πλήθος παραμέτρων 2.
2. FD3: Πεπερασμένες Διαφορές με πλήθος παραμέτρων 3.
3. SPSA2: SPSA με πλήθος παραμέτρων 2.
4. SPSA3: SPSA με πλήθος παραμέτρων 3.
5. LQR: Γραμμικός Τετραγωνικός Ελεγκτής.

6. AC: Μη άμεσος Προσαρμοστικός Έλεγχος.

### Συνάρτηση *gradientDescent*

Η συνάρτηση αυτή περιλαμβάνει μια δομή switch για να επιλέξει την αντίστοιχη μέθοδο από τις διαθέσιμες που υπάρχουν στον απεριθμητή *algorithms*.

Παίρνει ως είσοδο το αρχικό διάνυσμα κατάστασης, τις παραμέτρους  $\theta$ , τις παραμέτρους του συστήματος  $(m, c, k)$  και το ποιον αλγόριθμο θα εκτελέσει.

Επιστρέφει ένα διάνυσμα  $2 \times N$  όπου  $N$  το μέγιστο πλήθος επαναλήψεων.

Κατά την εκτέλεση των μεθόδων υπάρχει επιπλέον έλεγχος για τις περιπτώσεις που τα δεδομένα φτάσουν να είναι Άπειρα ή NaN λόγω υπερχείλησης ή άλλου σφάλματος.

Συνήθως βέβαια, έπειτα από δοκιμές, τα κομμάτια αυτά του κώδικα θα εκτελεστούν είτε όταν υπάρχει υπερχείληση είτε όταν οι αρχικές συνθήκες δεν είναι “καλές”.

### Συνάρτηση *main*

Στην *main* εκτελούνται σε 10 βήματα όλοι οι αλγόριθμοι και με όλες τις περιπτώσεις των παραμέτρων του συστήματος, όπως παρουσιάζονται στην επόμενη παράγραφο. Επίσης τυπώνονται και τα διαγράμματα.

### Βιβλιοθήκη *GNUplot*

Η βιβλιοθήκη αυτή χρησιμοποιήθηκε για την παραγωγή διαγραμμάτων των αποτελεσμάτων χωρίς να χρησιμοποιηθεί άλλη γλώσσα όπως η *python* ή αλλά εργαλεία όπως το *excel* ή το *matlab*.

## Αποτελέσματα

Στην παράγραφο αυτή παρουσιάζονται τα αποτελέσματα των εκτελέσεων. Μας ενδιαφέρει να δούμε την απόδοση συναρτήσεως του χρόνου αλλά και την ευστάθεια, η οποία θα φανεί από το διάγραμμα των μεταβλητών κατάστασης. Οι εκτελέσεις έχουν γίνει για 2 και 3 παραμέτρους  $\theta$  αλλά και για ένα πλήθος συνδυασμών των παραμέτρων του συστήματος  $(m, c, k)$  με τιμές -1 ή +1, όπως φαίνεται στον παρακάτω πίνακα:

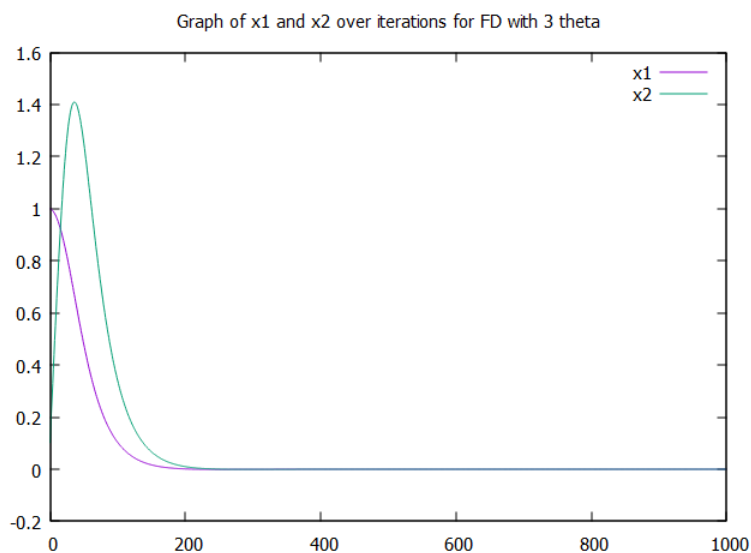
Αριθμός	$k$	$m$	$c$
1	+1	+1	+1
2	+1	+1	-1
3	+1	-1	+1
4	+1	-1	-1
5	-1	+1	+1
6	-1	+1	-1
7	-1	-1	+1
8	-1	-1	-1

Πίνακας 1: Συνδυασμοί των Παραμέτρων του Συστήματος  $(m, c, k)$

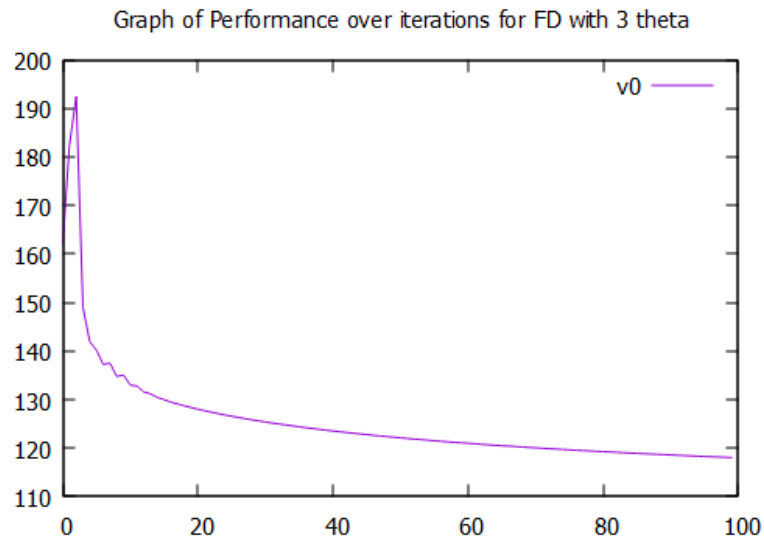
## Μέθοδος Πεπερασμένων Διαφορών (Finite Differences)

1. Εκτέλεση για  $m = k = c = 1$ :

A. Για  $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$ :

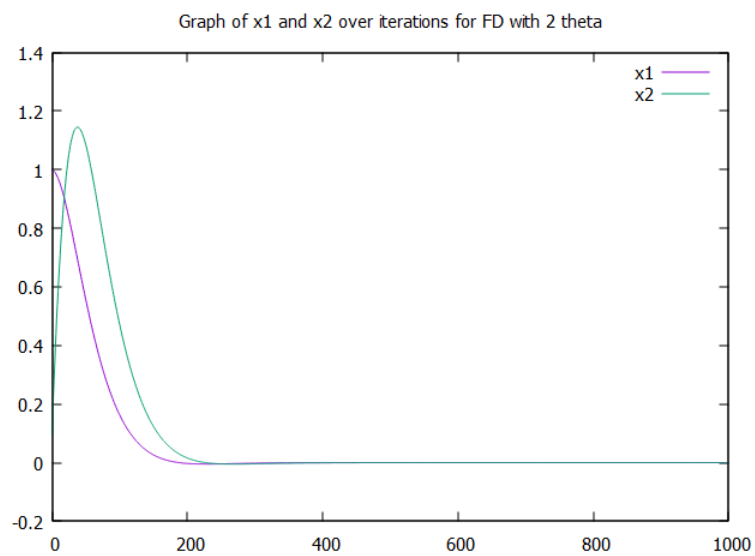


Εικόνα 1: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$



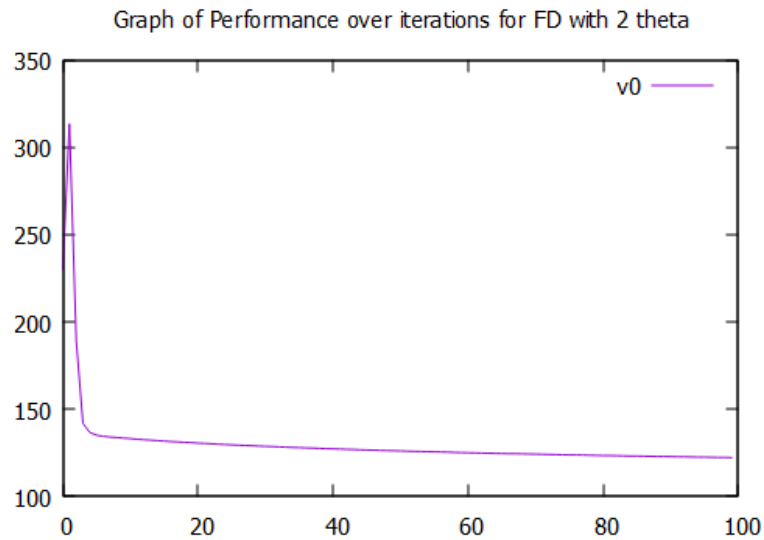
Εικόνα 2: Διάγραμμα  $P$  vs επαναλήψεις

Β. Για  $\vec{\theta} = (\theta_1, \theta_2)$ :



Εικόνα 3: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$

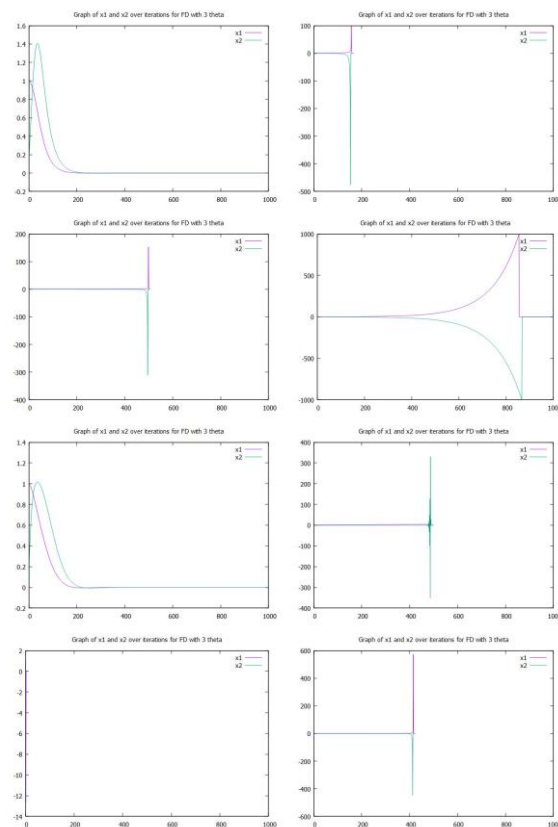




Εικόνα 4: Διάγραμμα  $P$  vs επαναλήψεις

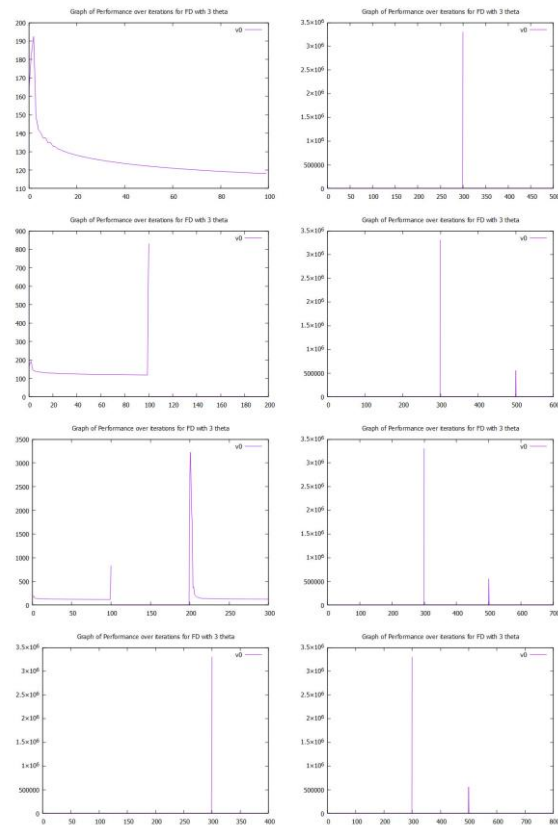
2. Εκτέλεση για όλους τους συνδιασμούς των  $(m, c, k)$  με τιμές -1 ή +1:

A. Για  $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$ :



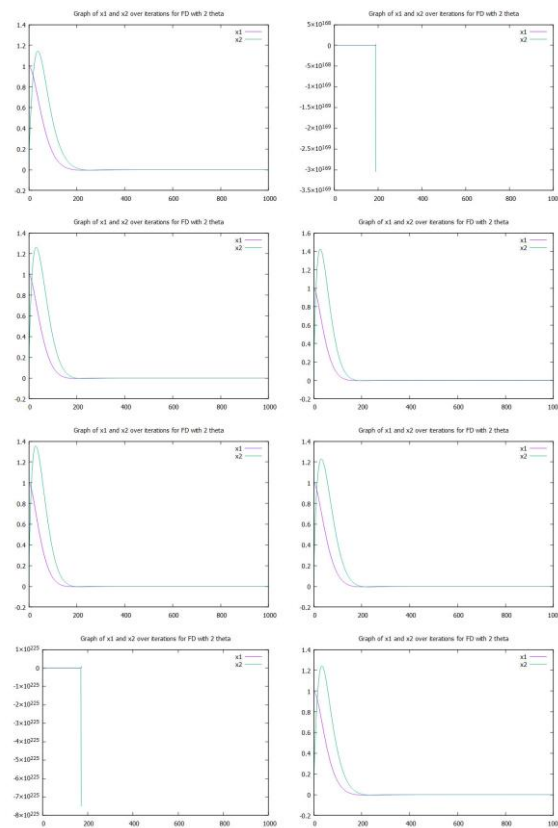
Εικόνα 5: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$

## Έλεγχος Ταλαντωτή Van der Pol - Ειδικά Θέματα Συστημάτων Αυτομάτου Ελέγχου

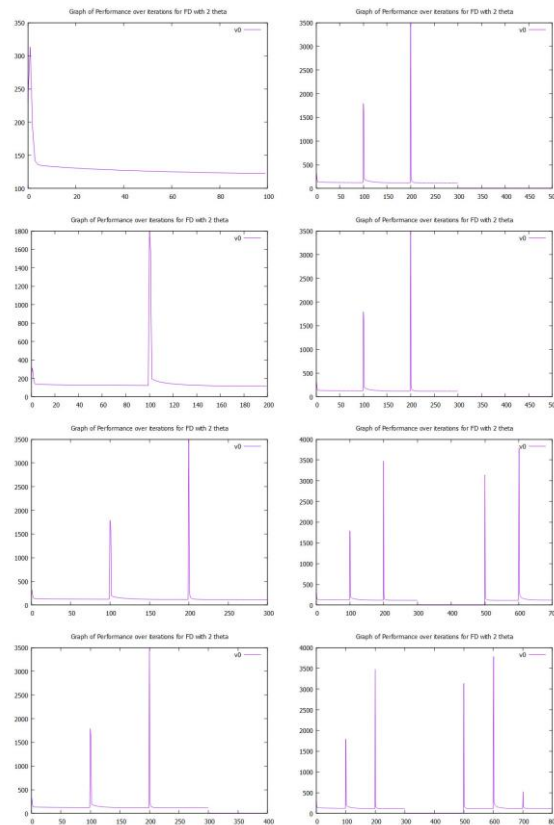


Εικόνα 6: Διάγραμμα  $P$  vs  $t$

Β. Για  $\vec{\theta} = (\theta_1, \theta_2)$ :



Εικόνα 7: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$  (αριστερή στήλη 1~4, δεξιά στήλη 5~8)

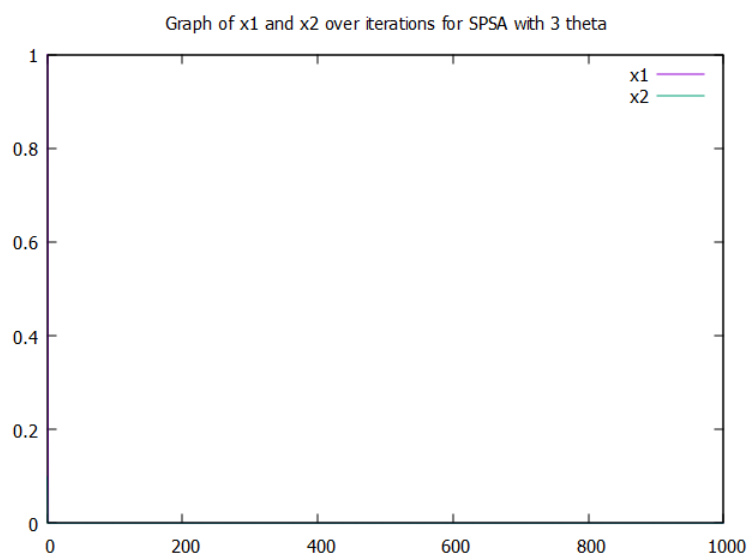


Εικόνα 8: Διάγραμμα  $P$  vs  $t$  (αριστερή στήλη 1~4, δεξιά στήλη 5~8)

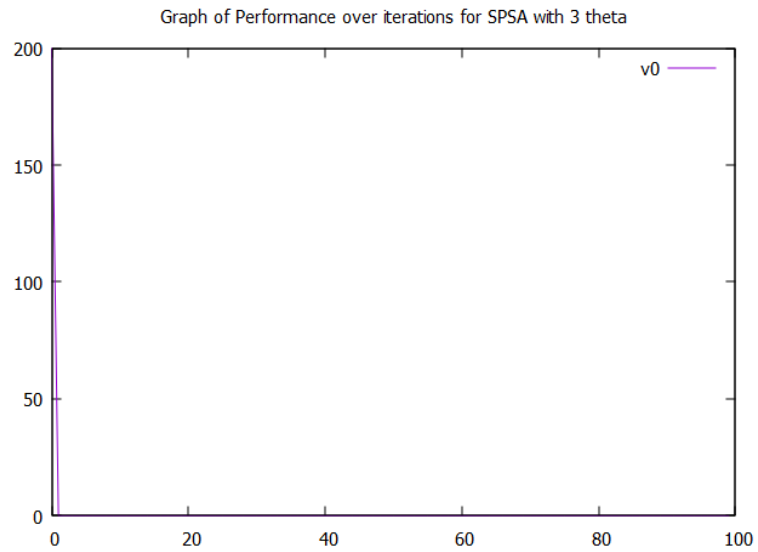
## Στοχαστική Προσέγγιση Ταυτόχρονης Διαταραχής (Simultaneous Perturbation Stochastic Approximation)

1. Εκτέλεση για  $m = k = c = 1$ :

A. Για  $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$ :

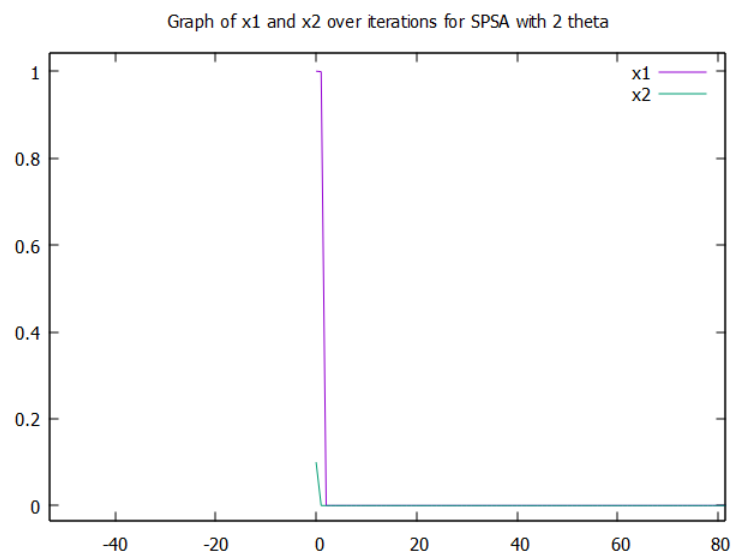


Εικόνα 9: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$

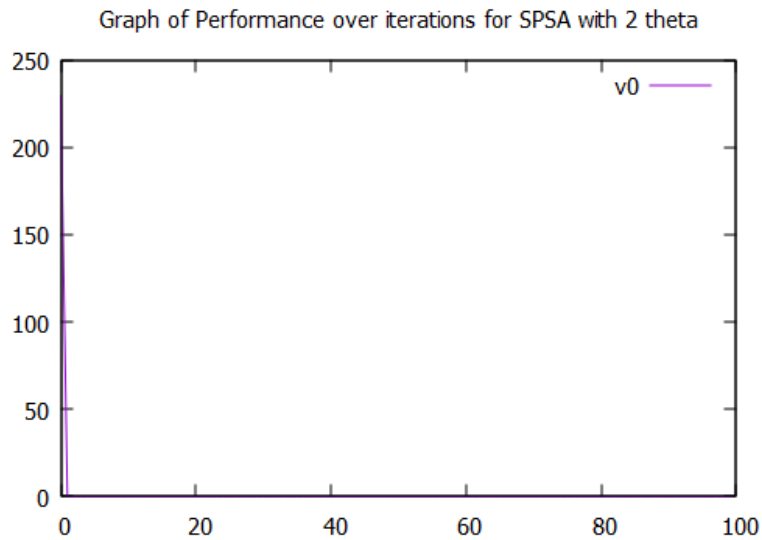


Εικόνα 10: Διάγραμμα  $P$  vs επαναλήψεις

Β. Για  $\vec{\theta} = (\theta_1, \theta_2)$ :



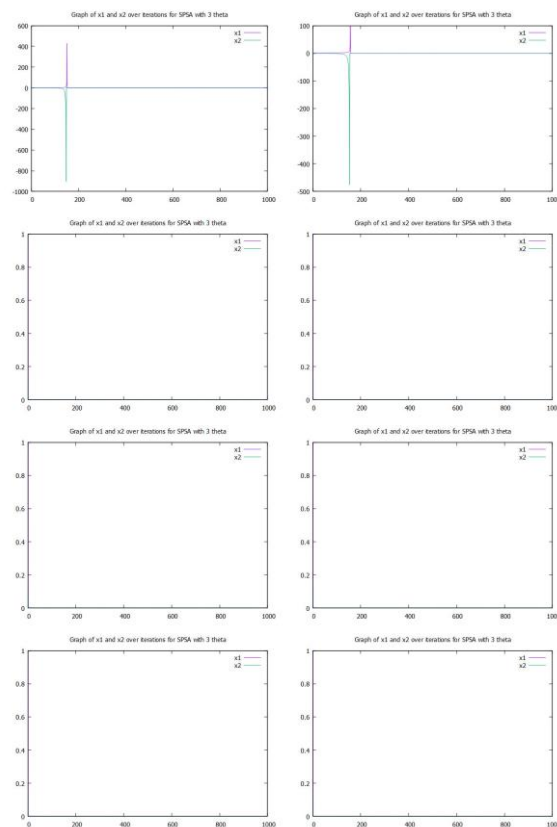
Εικόνα 11: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$



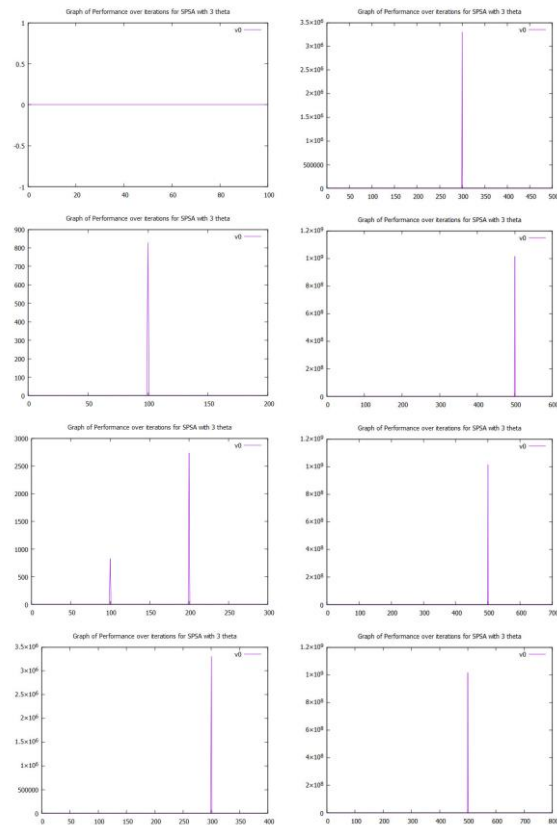
Εικόνα 12: Διάγραμμα  $P$  vs επαναλήψεις

2. Εκτέλεση για όλους τους συνδιασμούς των  $(m, k, c)$  με τιμές -1 ή +1

A. Για  $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$ :

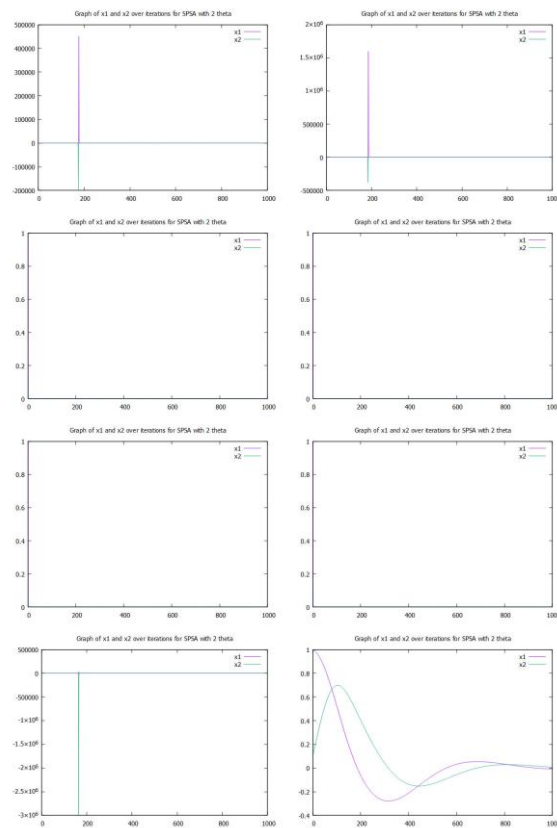


Εικόνα 13: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$

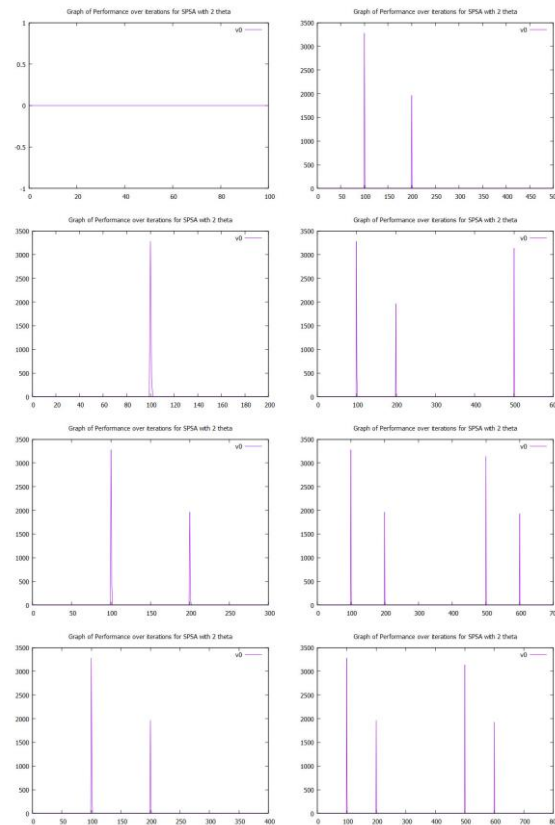


Εικόνα 14: Διάγραμμα  $P$  vs  $t$

Β. Για  $\vec{\theta} = (\theta_1, \theta_2)$ :



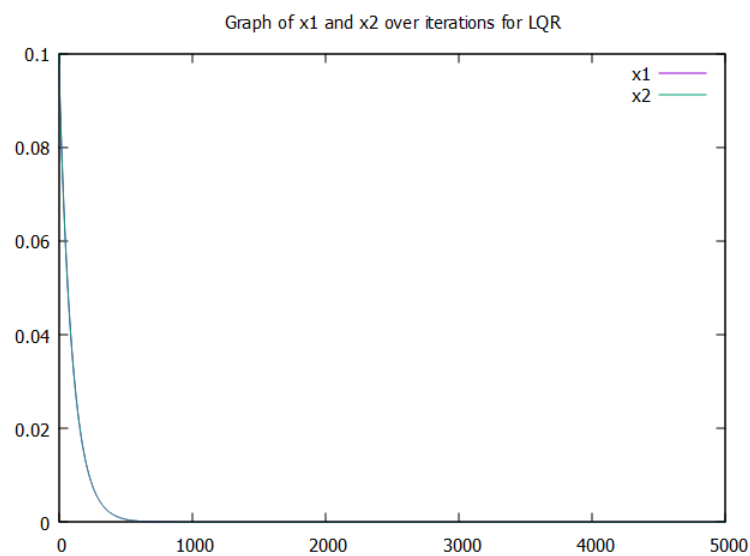
Εικόνα 15: Διάγραμμα  $x_1$  vs  $t$  και  $x_2$  vs  $t$



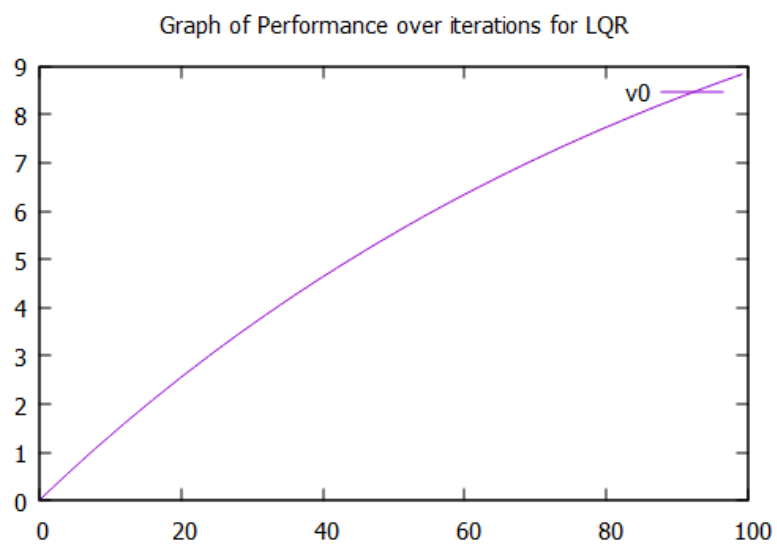
Εικόνα 16: Διάγραμμα  $P$  vs  $t$

## Γραμμικός Τετραγωνικός Ελεγκτής (Linear Quadratic Regulator)

1. Εκτέλεση για  $K = (k_1, k_2)$ :



Εικόνα 17: Διάγραμμα  $x_1$  vs  $x_2$



Εικόνα 18: Διάγραμμα  $P$  vs  $t$



## Συμπεράσματα

Ο αλγόριθμος που για τις περισσότερες τιμές έδινε έναν καλό έλεγχο (καλή απόδοση και τα  $x$  προσεγγίζουν το 0) ήταν η μέθοδος πεπερασμένων διαφορών(FD). Η μέθοδος SPSSA επειδή βασιζόταν σε τυχαία νούμερα εξαρτάται από την build-in συνάρτηση random\_device και χρειάζεται πολλές εκτελέσεις ολόκληρου του αλγορίθμου μέχρι να πετύχει καλή τυχαία τιμή. Ανάμεσα σε αυτούς τους δύο θα ήταν προτιμότερη η μέθοδος FD αν έχουμε μια ανοχή στην απόδοση, ενώ την SPSSA μπορούμε να προτιμήσουμε αν μπορούμε να την εκτελέσουμε περισσότερες φορές από την FD ώστε να πετύχουμε καλές τιμές.

Η μέθοδος LQR μπορεί σε πολύ λιγότερα βήματα να δώσει αποτέλεσμα σε αντίθεση με την FD και την SPSSA. Θα ήταν χρησιμη σε περιπτώσεις που ο χρόνος υπολογισμού είναι πιο κρίσιμος από την ακρίβεια των πράξεων και το μέγεθος των δεκαδικών. Αν δεν γνωρίζουμε τις παραμέτρους του συστήματος αλλά θέλουμε να το οδηγήσουμε η μέθοδος αυτή ενδεικνυται μιας και έχουμε απαλήψει τις παραμέτρους του συστήματος.

Στα διαγράμματα και όπως ο κώδικας έχει υλοποιηθεί, όπου υπάρχει απότομη αλλαγή από μια τιμή στο μηδέν, αυτό είναι για να γίνει διαχείριση των απείρων και των NaN που προκύπτουν από τους υπολογισμούς. Έτσι, για παράδειγμα η απόδοση μπορεί να αυξάνεται στην εικόνα 5 αλλά με το πρώτο άπειρο γίνεται έξοδος και σταματάει ο αλγόριθμος. Αυτό φυσικά είναι ένδειξη ότι χρειάστηκε πάρα πολύ προσπάθεια για να γίνει ο έλεγχος, πράγμα που δεν το επιθυμούμε. Στον SPSA στα διαγράμματα έχουμε ακραίες τιμές πάρα πολύ γρήγορα και αυτό οφείλεται στους σταθερούς συντελεστές αλλά και στην χαοτική συμπεριφορά που έχει η μέθοδος εξαιτίας της κατανομής  $bernoulli(p)$ . Αυτό είναι χρήσιμο γιατί αρχικά και σύμφωνα με την βιβλιογραφία η μέθοδος αυτή ανήκει στην κατηγορία εκείνων των μεθόδων που στοχεύουν στον έλεγχο με ένα βήμα. Στην εικόνα 9 και 10 φαίνεται αυτό ξεκάθαρα αφού για τα  $x$  δεν υπάρχει κριτήριο τερματισμού παραμόνο να γίνουν αρκετές εκτελέσεις. Επομένως οι μηδενικές τιμές δείχνουν ότι ήδη από πολύ νωρίς έχει φτάσει στην επιθυμητή κατάσταση. Να σημειωθεί επίσης ότι η απόδοση ή καλύτερα το κόστος για να ελεγχθεί σε ένα μόνο βήμα είναι σύμφωνα με την εικόνα 10 γύρω στο 200 το οποίο είναι καλύτερο από το 340 της FD με δύο  $\theta$  αλλά λίγο χειρότερο από την FD με τρία  $\theta$ .

## Αναφορές

Nonlinear and Optimal Control Systems by Thomas L. Vincent and Walter J. Grantham, John Wiley & Sons, Inc., Ed. 1, 2021

Underactuated Robotics - Algorithms for Walking, Running, Swimming, Flying and Manipulation, Course Notes for MIT 6.832 by Tedrake Russ, 2023, [Link](#)

Fuzzy System Identification and Adaptive Control - Adaptive Control: A Tutorial Introduction, by Ruiyun Qi, Gang Tao, Bin Jiang, 2019, [Link](#)

Biological Applications of the “Filtered” Van der Pol Oscillator, Journal of the Franklin Institute by B.Z. Kaplan, I. Gabay, G. Sarafian, D. Sarafian, 2008, [Link](#)

Introduction to Stochastic Search and Optimization - Simultaneous Perturbation Stochastic Approximation, Johns Hopkins University by James C. Spall, 2001, [Link](#)

Volume 55 - Van der Pol Oscillator. Technical Applications, by V. Ph. Zhuravlev, 2020, [Link](#)