ΤΕΧΝΙΚΕΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ PROJECT 2020-2021

Νικόλαος Γιακουμόγλου 9043

January 5, 2021

1 Περιγραφή του προβλήματος

1.1 Το σύστημα

Εστω ϕυσικό σύστημα δύο εισόδων - δύο εξόδων που περιγράφεται από την εξίσωση

$$\frac{dx}{dt} = f(x, u)$$

όπου $f(x, u) : \mathbb{R}^4 \times \mathbb{R}^2 \to \mathbb{R}^2$ άγνωστη με

$$x = [x_1, x_2, x_3, x_4]^T \in \mathbb{R}^4$$

$$u = [u_1, u_2]^T \in \mathbb{R}^2 \ [Volt]$$

$$y = [x_1, x_3] \in \mathbb{R}^2 \ [rad]$$

Το σύστημα μπορεί να ελεχθεί με έναν PID ελεγκτή όπου

$$u(t) = -K_P \cdot e(t) - K_I \int_0^t e(\tau)d\tau - K_D \cdot e'(t)$$

$$e(t) = [x_1(t) - y_{d,1}(t), x_3(t) - y_{d,2}(t)]^T$$

με

$$K_p = \begin{bmatrix} K_{P,1} & 0 \\ 0 & K_{P,2} \end{bmatrix} \in [0, 100]$$

$$K_I = \begin{bmatrix} K_{I,1} & 0 \\ 0 & K_{I,2} \end{bmatrix} \in [0, 10]$$

$$K_D = \begin{bmatrix} K_{D,1} & 0 \\ 0 & K_{D,2} \end{bmatrix} \in [0, 100]$$

Το σύστημα θεωρείται αρχικά στη θέση

$$x = [x_1(0), x_2(0), x_3(0), x_4(0)]^T = [\frac{110\pi}{180}, 0, \frac{100\pi}{180}, 0]^T$$

Ο στόχος ελέγχου είναι η έξοδος του συστήματος να παρακολουθεί τις τροχιές

$$y_{d,1}(t) = \frac{90\pi}{180} + \frac{30\pi}{180} \cdot \cos(t)$$

$$y_{d,2}(t) = \frac{90\pi}{180} + \frac{30\pi}{180} \cdot \sin(t)$$

1.2 Οι προδιαγραφές

1. Σϕάλματα παρακολούθησης εξόδου στη μόνιμη κατάσταση το πολύ $\frac{\pi}{180}$ [rad]
2. Υπερύψωση στα σϕάλματα παρακολούθησης εξόδου το πολύ $\frac{\pi}{180}$ [rad]
3. Χρόνος αποκατάστασης των σϕαλμάτων παρακολούθησης εξόδου στη ζώνη $[-\frac{\pi}{180}, \frac{\pi}{180}]$ το πολύ 1 [s]
4. Είσοδο ελέγχου που η στιγμιαία της τιμή είναι κατά το δυνατό μικρότερη και σε κάθε περίπτωση όχι μεγαλύτερη από 18 [V]
5. Είσοδο ελέγχου που η στιγμιαία της μεταβολή είναι κατά το δυνατό μικρότερη και σε κάθε περίπτωση όχι μεγαλύτερη από 160 [V/s]
6. Είσοδο ελέγχου που η συνολική της μεταβολή είναι κατά το δυνατό μικρότερη.

2 Δείκτες και Fitness Function

2.1 Δείκτες στον χρόνο t

1. $J_1 = \max_{t\to\infty}(|y_i(t) - y_{d,i}(t)|)$, $i = 1, 2$
2. $J_2 = \max_{t>0}(|y_i(t) - y_{d,i}(t)|)$, $i = 1, 2$
3. $J_3 = \underset{t}{\operatorname{argmax}}_{t>0}(|y_i(t) - y_{d,i}(t)|)$, $i = 1, 2$
4. $J_4 = \max_{t\geq 0}|u_i(t)|$, $i = 1, 2$
5. $J_5 = \max_{t\geq 0}|u_i'(t)|$, $i = 1, 2$
6. $J_6 = \max_{t\geq 0}\{u_i(t)\} - \min_{t\geq 0}\{u_i(t)\}$, $i = 1, 2$ (θα μπορούσε και $J_6 = \lim_{t\to\infty} u_i(t) - u_i(0)$, $i = 1, 2$;)

όπου $y_1(t) = x_1(t)$ και $y_2(t) = x_3(t)$

2.2 Δείκτες στα δείγματα n

1. $J_1 = \max_{n\to\infty}(|y_i[n] - y_{d,i}[n]|)$, $i = 1, 2$
2. $J_2 = \max_{n>0}(|y_i[n] - y_{d,i}[n]|)$, $i = 1, 2$
3. $J_3 = \underset{n}{\operatorname{argmax}}_{n>0}(|y_i[n] - y_{d,i}[n]|)$, $i = 1, 2$
4. $J_4 = \max_{n\geq 0}|u_i[n]|$, $i = 1, 2$
5. $J_5 = \max_{n\geq 0}|\frac{u_\iota[n]-u_\iota[n-1]}{n-n+1}|$, $i = 1, 2$
6. $J_6 = \max_{n\geq 0}\{u_i[n])\} - \min_{n\geq 0}\{u_i[n]\}$, $i = 1, 2$

όπου $y_1[n] = x_1[n]$ και $y_2[n] = x_3[n]$

2.3 Fitness Function

Ορίζουμε ως fitness function την

$$fitness\_function = \frac{1}{6}\sum_{i=1}^{6} f(J_i; b_i)$$

Εδώ εισάγουμε μια ποινή κάθως φορά που παραβιάζεται ένας περιορισμός. Η ποινή είναι ανάλογη του πόσο παραβιάζεται ο περιορσιμός με μέγιστη τιμή 100. Γι'αυτό χρησιμοποιούμε μια συνάρτηση

$$f(x; b) = -100 \cdot e^{-b\cdot x} + 100$$

με κατάλληλη επιλογή του $b > 0$ που ισχύει $\lim_{x\to\infty} f(x;b) = 100$ και $f(0) = 0$ και $f(x)$ γνησίως αύξουσα. Έτσι αν παραβιάζεται ο περιορισμός εισάγεται μια ποινή που είναι τόσο μεγάλη, όσο είναι και η απόκλιση από τον περιορισμό. Μάλιστα, επειδη θέλουμε μερικά σήματα όσο τον δυνατόν μικρότερα, εισάγουμε την ποινή χωρίς συνθήκη.

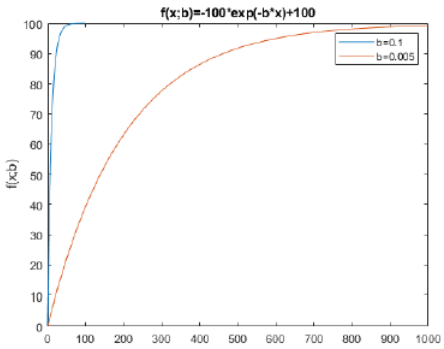

Figure 1: $f(x) = -100 \cdot e^{-b \cdot x} + 100$

Συγκεκριμένα για τους δείκτες $1-3$, αν παραβιάζονται επιλέγουμε $b = 0.1$ ενώ για τους $4-6$ εισάγουμε την ποινή με $b = 0.005$ είτε παραβιάζεται ο περιορισμός είτε όχι. Η διαφορετική επιλογή του b οφείλεται στο διαφορετικό εύρος τιμών των περιορισμών κατόπιν παρατήρησης. Ηδη γνωρίζουμε ότι η μέγιστη τιμή της ποινής είναι 100. Για να έχει και η fitness function μέγιστη τιμή 100, προσθέτουμε το $\frac{1}{6}$ της ποινής στην fitness function (αφού όλοι οι δείκτες είναι 6). Ετσι η fitness function έχει άνωθεν κατώφλι 100 και κάτωθεν το 0 με το 0 να δηλώνει perfect fit (δεν παραβιάζεται κανένας περιορισμός).

3 Επίλυση με την συνάρτηση *ga* του *MATLAB*

Το script βρίσκεται στον φάκελο *Genetic Algorithm MATLAB*. Τρέχουμε τον γενετικό αλγόριθμο για αρχικό πληθυσμό μεγέθους 200 και 50 generations. Η παράμετροι της fitness function είναι

$$x = [K_{P,1}, K_{P,2}, K_{I,1}, K_{I,2}, K_{D,1}, K_{D,2}]$$

που είναι θετικοί και μικρότεροι του 100 εκτός των $K_{I,1}, K_{I,2}$ που είναι μικρότεροι του 10. Με την εντολή *ga* τρέχουμε τον γενετικό αλγόριθμο στο *MATLAB* και βρίσκουμε ως βέλτιστες τιμές τις ακόλουθες

$$\begin{bmatrix} K_{P,1} \\ K_{P,2} \\ K_{I,1} \\ K_{I,2} \\ K_{D,1} \\ K_{D,2} \end{bmatrix} = \begin{bmatrix} 70.9380 \\ 48.1870 \\ 1.4791 \\ 9.4318 \\ 99.8956 \\ 8.8191 \end{bmatrix}$$

Εποπτικά οι παράμετροι φαίνεται να τηρούν τις προδιαγραφές.

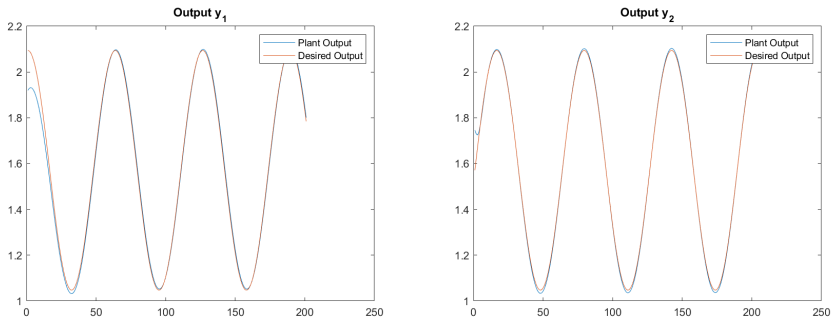

Figure 2: Παρακολούθηση της επιθυμητής εξόδου

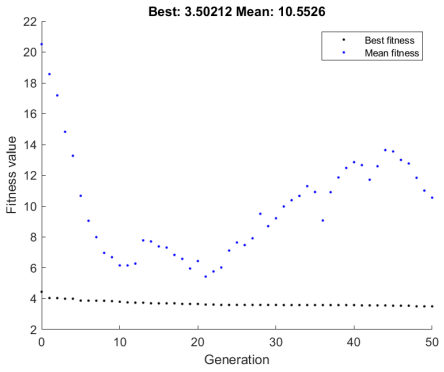

Figure 3: Γενετικός Αλγόριθμος: Fitness values σε κάθε generation

4 Επίλυση με Genetic Algorithm From Scratch

4.1 Ο Γενετικός Αλγόριθμος

Ο αλγόριθμος που υλοποιήθηκε αποτελείται από τις εξής διαδικασίες:

- Selection
- Crossover
- Mutation
- Elitism

Στο στάδιο της επιλογής (selection), διαλέγουμε τα πιο "καλά" χρωμοσώματα, δηλαδή αυτά με το μικρότερο fitness value. Όπως και στη φύση, δεν επιβιώνουν πάντοτε τα ικανότερα χρωμοσώματα, ωστόσο έχουν μεγαλύτερη πιθανότητα. Μετά ακολουθεί το στάδιο της διασταύρωσης (crossover) που "ζευγαρώνουν" δύο χρωμοσώματα ώστε να δημιουργούν νέα. Για τον λόγο αυτό λαμβάνονται τυχαία ζεύγη γονέων χρωμοσωμάτων. Στο επόμενο στάδιο ακολουθεί η μετάλλαξη γονιδίων (mutation). Επιλέγονται τυχαία από τον εκάστοτε πληθυσμό χρωμοσώματα και μεταλλάσσονται με τυχαίο τρόπο κάποια από τα γονίδια αυτών. Το τελευταίο στάδιο είναι του ελιτισμού (elitism) κατά το οποίο τα "καλύτερα" χρωμοσώματα επιβιώνουν στις επόμενες γενεές. Τα 4 αυτά στάδια υλοποιούνται σε ξεχωριστές συναρτήσεις στο *MATLAB* (*Crossover.m*, *Elitism.m*, *Mutation.m*, *Selection.m*). Ο γενετικός αλγόριθμος υπάρχει στην συνάρτηση *GeneticAlgorithmFromScratch.m* και δεν είναι τίποτα άλλο απο εφαρμογή των 4 σταδίων σε κάποιο αρχικό αριθμό χρωμοσωμάτων για προκαθορισμένο αριθμό γενεών. Η fitness function είναι αυτή του Κεφαλαίου 2.3. Ο custom made genetic algorithm βρίσκεται στον φάκελο *Genetic Algorithm From Scratch*.

4.2 Αποτελέσματα

Τρέχουμε τον γενετικό αλγόριθμο που φτιάξαμε για αρχικό population μεγέθους 200, 50 generations και εκ κατασκευής του προβλήματος 6 genes (όπως και στον *ga* που τρέξαμε στο Κεφάλαιο 3). Θέτουμε την πιθανότητα crossover ίση με 0.8, την πιθανότητα mutation και αυτή ίση με 0.8 και την πιθανότητα του ελιτισμού ίση με 0.01. Βρίσκουμε ως βέλτιστες τιμές τις ακόλουθες

$$\begin{bmatrix} K_{P,1} \\ K_{P,2} \\ K_{I,1} \\ K_{I,2} \\ K_{D,1} \\ K_{D,2} \end{bmatrix} = \begin{bmatrix} 80.0280 \\ 84.4309 \\ 1.2712 \\ 8.6394 \\ 95.6936 \\ 42.0429 \end{bmatrix}$$

Εποπτικά οι παράμετροι φαίνεται να τηρούν τις προδιαγραφές.

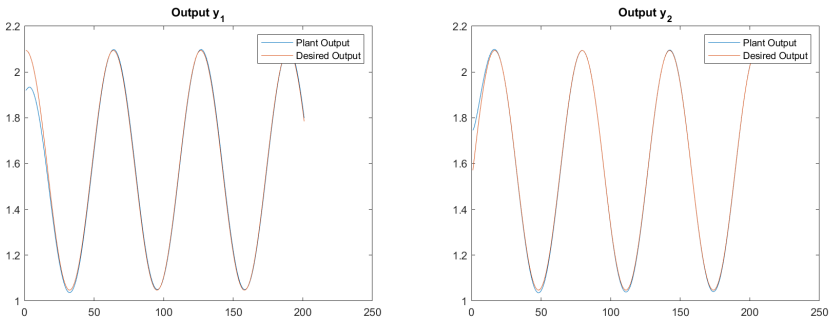

Figure 4: Παρακολούθηση της επιθυμητής εξόδου

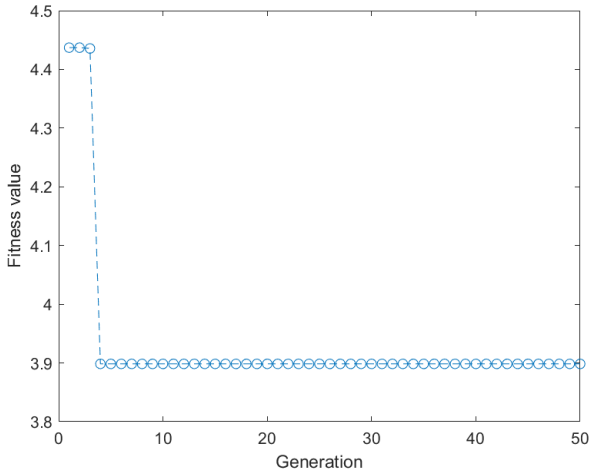

Figure 5: Γενετικός Αλγόριθμος: Fitness values σε κάθε generation