



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ
Ασαφής Ελεγχτής

ΕΙΣΗΓΗΤΗΣ : ΘΕΟΧΑΡΗΣ ΙΩΑΝΝΗΣ
ΚΩΣΤΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
kostakonst@ece.auth.gr
ΑΕΜ : 9209

Ιανουάριος 2022

Contents

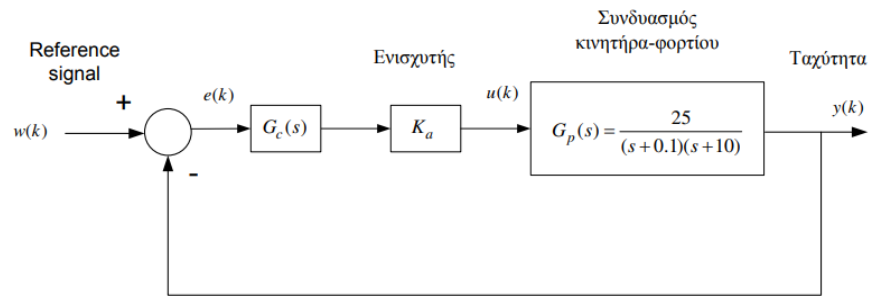
1	Εισαγωγή	3
2	Σχεδίαση γραμμικού ελεγκτή	3
3	Σχεδίαση ασαφούς ελεγκτή (FLC)	5
3.1	Σενάριο 1	11
3.2	Σενάριο 2	18

1 Εισαγωγή

Στόχος της εργασίας αποτελεί ο έλεγχος ταχύτητας ενός μηχανισμού τραπεζιού εργασίας. Ο μηχανισμός αυτός αποτελείται από ένα dc κινητήρα και ένα ταχύμετρο ακριβείας. Αρχικά, με την χρήση γραμμικού ελεγκτή PI και στην συνέχεια με ασαφή ελεγκτή FZ-PI ζητείται η διατήρηση της υψηλής ακρίβειας του ελέγχου ταχύτητας στην μόνιμη κατάσταση λειτουργίας. Η συνάρτηση μεταφοράς του συνδυασμού κινητήρα- φορτίου είναι:

$$G_p(s) = \frac{25}{(s + 0.1)(s + 10)}$$

2 Σχεδίαση γραμμικού ελεγκτή



Σχ.1

Με την χρήση του γραμμικού ελεγκτή αναλογικής-ολοκληρωτικής δράσης(PI) το σύστημα παίρνει την μορφή του σχ1. Η σχέση που περιγράφει την λειτουργία του ελεγκτή αυτού είναι:

$$G_c(s) = K_p + \frac{K_i}{s} = \frac{K_p(s + c)}{s}, c = \frac{K_i}{K_p}$$

Οι προδιαγραφές που θα πρέπει να πληρούνται είναι:

- Overshoot < 8 %
- Rise time < 0.6 SEC

Με την χρήση του `controlSystemDesigner()`, που παρέχεται απο το Matlab, μπορεί να κατασκευαστεί και να ρυθμιστεί ο γραμμικός ελεγκτής διαδραστικά. Εκεί, μέσα από μια διαδικασία δοκιμής και λάθους (trial and error) μπορούν να υπολογιστούν και οι παράμετροί του. Κατά την διαδικασία αυτή, γνωρίζοντας ότι η αύξηση του κέρδους οδηγεί σε μικρότερο "Rise time" και μεγαλύτερο "Overshoot" επιλέχτηκε $K=1.85$ και η τιμή του c για την οποία επαληθεύονται οι προδιαγραφές βρέθηκε $c=0.3$.

Με τις τιμές αυτές, η συνάρτηση μεταφοράς κλειστού βρόχου του συστήματος γίνεται:

$$Tf(s) = \frac{46.25(s + 0.3)}{(s + 0.3141)(s^2 + 9.786s + 44.18)}$$

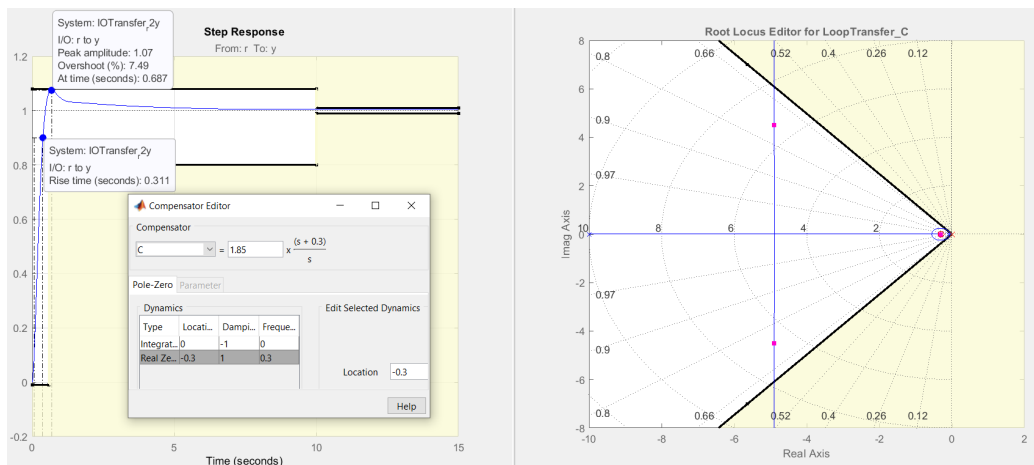
και με αντικατάσταση υπολογίζεται

$$Kp = 1.85$$

$$Ki = 0.555$$

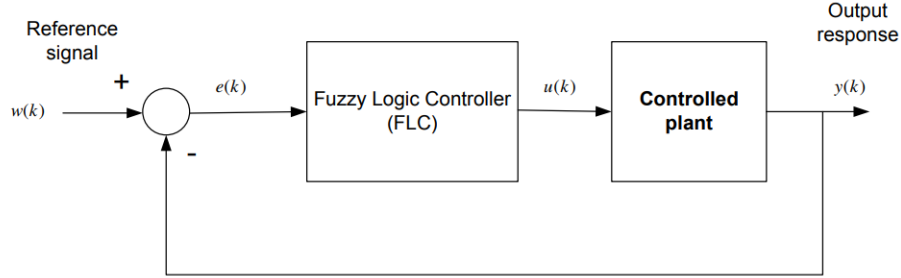
$$C = 0.3$$

Η διαδικασία αυτή ρύθμισης των παραμέτρων με τις τελικές τιμές απεικονίζεται και στα ακόλουθα διαγράμματα.



Το αρχείο `ControlSystemDesigner_PI.mat` περιέχει την ανάλυση αυτή.

3 Σχεδίαση ασαφούς ελεγκτή (FLC)



Σχ.2

Με την χρήση του ασαφούς ελεγκτή αναλογικής-ολοκληρωτικής δράσης(FZ-PI) το σύστημα παίρνει την μορφή του σχ2. Η σχέση που περιγράφει μαθηματικά την λειτουργία του ελεγκτή αυτού σε συνεχή χρόνο είναι:

$$\dot{u}^{PI}(t) = K_p \dot{e}(t) + K_i e(t),$$

όπου K_p, K_i είναι τα κέρδη του αναλογικού (proportional) και του ολοκληρωτικού (integral) όρου, αντίστοιχα.

Ψηφιακή υλοποίηση των ασαφών ελεγκτών

Για την υλοποίηση του ασαφούς ελέγχου σε έναν ψηφιακό υπολογιστή, και σε διακριτό χρόνο, είναι απαραίτητη η προσέγγιση των ολοκληρωμάτων και των παραγώγων που εμφανίζονται στον αλγόριθμο ελέγχου.

Η ολοκληρωτική πράξη που εμφανίζεται στην έξοδο ενός FZ-PI ελεγκτή, μπορεί να προσεγγιστεί ως εξής:

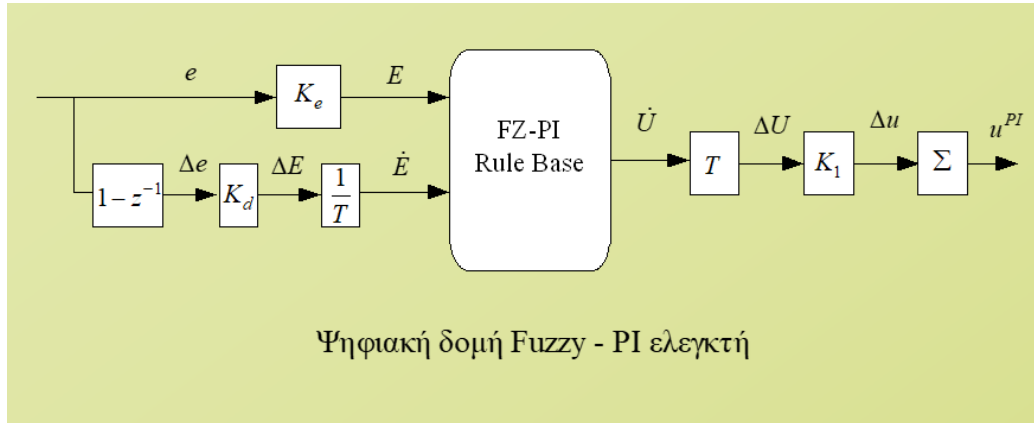
$$\dot{u}^{PI}(\kappa) = u^{PI}(\kappa - 1) + K_i \Delta U, \text{ όπου}$$

$$\Delta U = U(k) - U(k - 1) = \dot{U}(k)T$$

Η πράξη της παραγώγισης στις εισόδους των ελεγκτών, μπορεί να προσεγγιστεί ως εξής:

$$\dot{e}(k) = \frac{\Delta e(t)}{T}$$

Η δομή του σε ψηφιακή μορφή είναι:



όπου K_e , K_d κέρδη κλιμακοποίησης εισόδων, K_1 κέρδος κλιμακοποίησης εξόδου και T ο χρόνος δειγματοληψίας.

Συνεπώς, ο ασαφής ελεγκτής FZ-PI, για κάθε χρονική στιγμή, υλοποιεί μέσω της ασαφούς βάσης κανόνων μια σχέση ανάμεσα στο σφάλμα και την μεταβολή του σφάλματος από την μια μεριά και από την άλλη την μεταβολή του σήματος ελέγχου.

Κανονικοποίηση μεταβλητών

Τα σήματα εισόδου καλό είναι να βρίσκονται στο διάστημα $[-1,1]$. Με αυτόν τον τρόπο, επιτυγχάνεται ομοιομορφία στην σχεδίαση των ασαφών ελεγκτών. Επομένως,

$$e(k) = r(k) - y(k)$$

και με δεδομένο ότι,

$$\begin{aligned} r(k) &\in [0, 50], \\ y(k) &\in [0, 50] \end{aligned}$$

το σφάλμα και η μεταβολή εισόδου ορίζονται στα ακόλουθα διαστήματα:

$$e(k) \in [-50, 50]$$

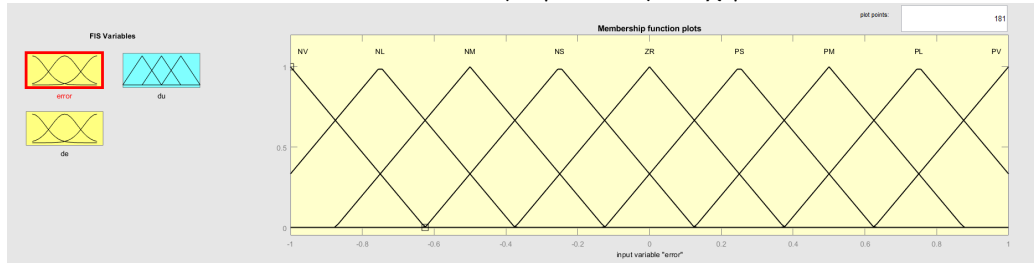
$$de(k) \in [-100, 100]$$

Συνεπώς, για την κανονικοποίηση των μεταβλητών εισόδου πολλαπλασιάζεται το $e(k)$ με $1/50$ και το $de(k)$ με $1/100$. Έτσι τελικά:

$$de(k), e(k) \in [-1, 1]$$

Στη συνέχεια, με την βοήθεια των λεγόμενων συναρτήσεων συμμετοχής (membership functions) μετατρέπονται αυτές οι τιμές σε fuzzy (fuzzification). Ο μερισμός του χώρου των εισόδων και της εξόδου γίνεται σε εννέα ασαφή σύνολα όπως παρουσιάζονται και στην συνέχεια.

Figure 1: Συναρτήσεις συμμετοχής



Ασαφής Βάση Κανόνων (Fuzzy Rule Base)

Εξάγεται από το πρότυπο των Mc Vicar και Wheelan και αποτελείται από κανόνες της μορφής.

If (error is NV) and (de is PV) then (du is ZR) (1)

Επειδή οι δύο είσοδοι ορίζονται από (9) λεκτικές τιμές η κάθε μία, η ασαφής βάση κανόνων θα αποτελείται συνολικά από $9 \times 9 = 81$ κανόνες όπως φαίνονται και στον ακόλουθο πίνακα.

error de	NV	NL	NM	NS	ZR	PS	PM	PL	PV
PV	ZR	PS	PM	PL	PV	PV	PV	PV	PV
PL	NS	ZR	PS	PM	PL	PV	PV	PV	PV
PM	NM	NS	ZR	PS	PM	PL	PV	PV	PV
PS	NL	NM	NS	ZR	PS	PM	PL	PV	PV
ZR	NV	NL	NM	NS	ZR	PS	PM	PL	PV
NS	NV	NV	NL	NM	NS	ZR	PS	PM	PL
NM	NV	NV	NV	NL	NM	NS	ZR	PS	PM
NL	NV	NV	NV	NV	NL	NM	NS	ZR	PS
NV	NV	NV	NV	NV	NV	NL	NM	NS	ZR

Τελεστής συμπερασμού και σύνθεσης

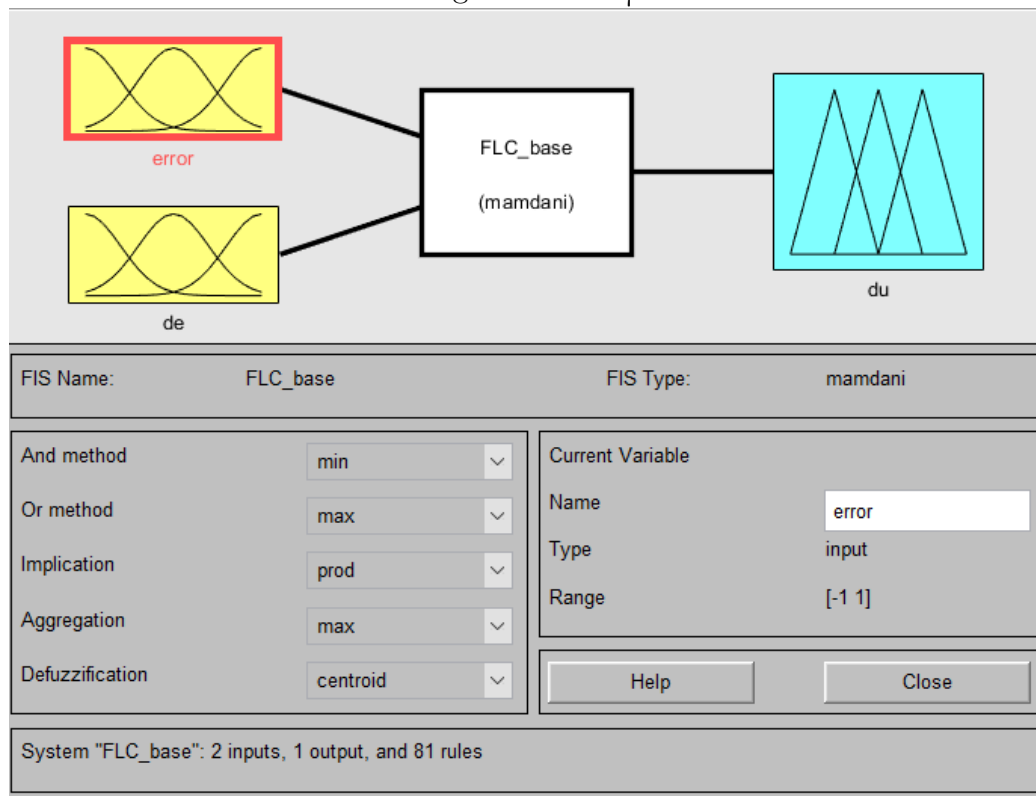
Ο τελεστής συμπερασμού που θα υλοποιήσει τους κανόνες είναι ο larsen, ενώ ο τελεστής σύνθεσης που θα χρησιμοποιηθεί είναι ο max-min. Το λογικό AND υλοποιείται με το min και το λογικό ALSO υλοποιείται με το max.

Επιλογή ασαφοποιητή/απο-ασαφοποιητή

Συγκεκριμένα, στην άσκηση αυτή ζητείται ασαφοποιητής τύπου Singleton. Με άλλα λόγια, θεωρείται ότι οι μετρήσεις των error και de που προκύπτουν από το σύστημα είναι καθαρές. Οι μετρούμενες τιμές περνάνε από τον ασαφοποιητή, δημιουργούν τα αντίστοιχα singleton και διεγείρουν κατάλληλα την βάση κανόνων. Τέλος, επιλέγεται ο τύπος του απο-ασαφοποιητή που είναι αυτός του κέντρου βάρους (centroid)

Η ρύθμιση των παραμέτρων αυτών παρουσιάζεται και στην εικόνα που ακολουθεί.

Figure 2: Βάση



Όλη η διαδικασία αυτή σχεδίασης του ασαφούς ελεγχτή (FLC) πραγματοποιήθηκε μέσω του Fuzzy editor που παρέχεται στο περιβάλλον του Matlab. Πιο συγκεκριμένα, σε αρχείο που κατασκευάστηκε στο περιβάλλον του *FuzzyLogicDesigner* με όνομα *FLC_base* υπάρχουν οι συναρτήσεις συμμετοχής, οι κανόνες της βάσης αλλά και τα χαρακτηριστικά του flc που ζητήθηκαν στην εκφώνηση της άσκησης.

Τελική δομή ελέγχου

Η εργασία πραγματοποιείται τον έλεγχο ταχύτητας μέσα από δύο ελεγκτές

διαφορετικής αρχιτεκτονικής και φιλοσοφίας που όμως ο τελικός στόχος τους παραμένει κοινός. Μέσα απο την ανάλυση που προηγήθηκε και την χρήση του *simulink* κατασκευάστηκαν κυκλώματα για την προσομοίωση και σύγκριση των ελεγχτών με απώτερο στόχο την κατασκευή ενός ασαφούς ελεγκτή με καλύτερα χαρακτηριστικά από αυτά του γραμμικού. Τα κυκλώματα που χρησιμοποιήθηκαν είναι :

Figure 3: Κύκλωμα σύγκρισης

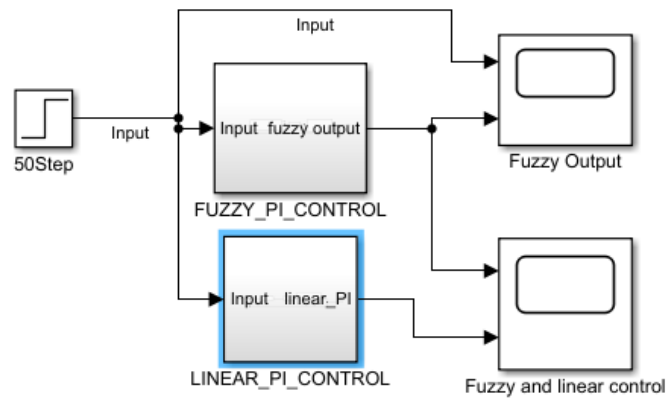


Figure 4: LINEAR_PI_CONTROL

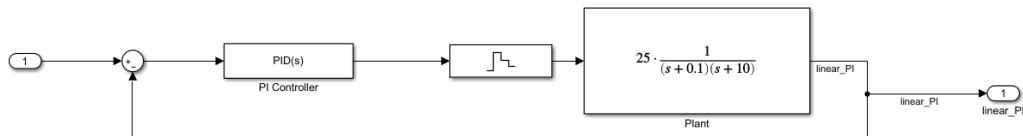


Figure 5: FUZZY_PI_CONTROL

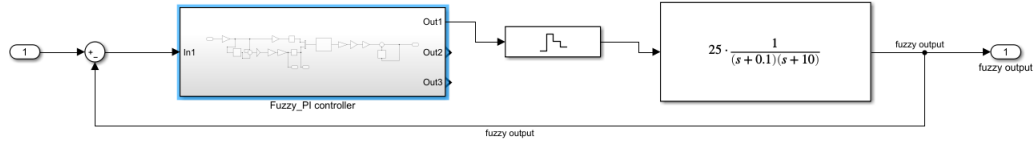
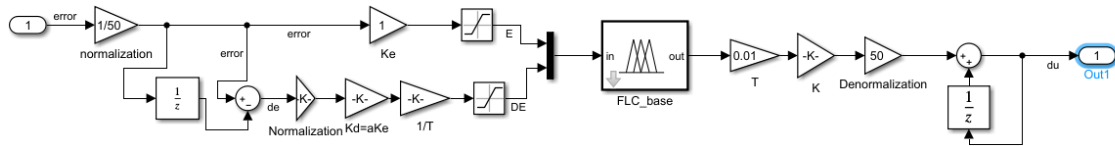


Figure 6: FUZZY_PI_CONTROLLER



3.1 Σενάριο 1

Ρυθμιση κερδών

Αρχικά, οι τιμές των κερδών του ασαφούς ελεγκτή, με δεδομένα τις τιμές του γραμμικού, ορίστηκαν ως εξής:

$$K_e = 1$$

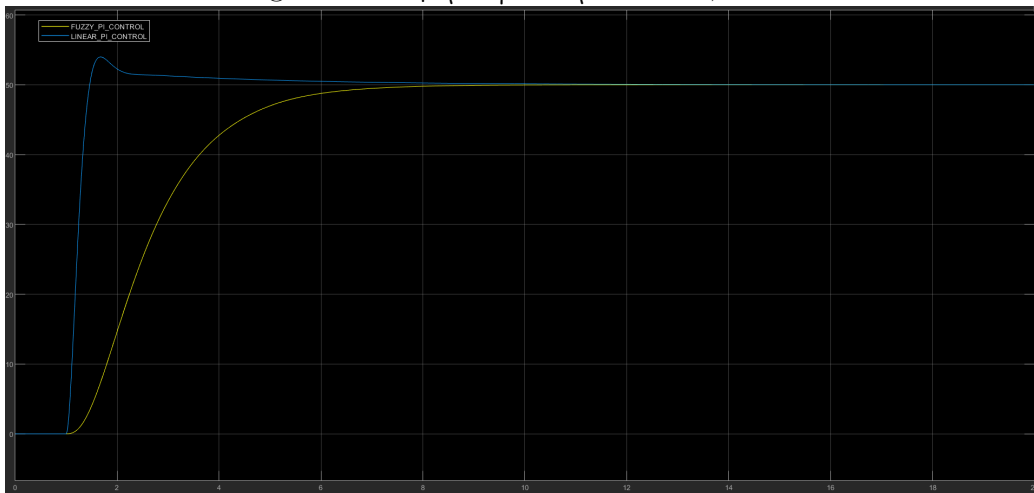
$$T_i = a = \frac{K_p}{K_i} = \frac{1.85}{0.555} = 3.333$$

$$K = \frac{K_p}{F(aK_e)} = \frac{1.85}{1} = 1.85$$

Τα αποτελέσματα που προκύπτουν με αυτά τα κέρδη, παρόλο που καθιστούν το σύστημα ευσταθές, δεν είναι ικανοποιητικά μιας και η απόκριση είναι πιο αργή σε σχέση με το αντίστοιχο γραμμικό ελεγκτή. Αυτό φαίνεται πιο καθαρά στο διάγραμμα που ακολουθεί (εικόνα 7).

Συνεπώς, κρίνεται απαραίτητη η αλλαγή των κερδών τα οποία μπορούν να καθοριστούν μέσα απο μια διαδικασία trial and error. Η διαδικασία αυτή μπορεί να επιταχυνθεί με κάποιους γενικούς κανόνες.

Figure 7: Σύγκριση αποκρίσεων Pi, FZ-Pi



Σύμφωνα, λοιπόν, με τις σημειώσεις του κ.Θεοχάρη Ι. (Fuzzy-present-FLC-09)

- Αύξηση της τιμής του κέρδους K ισοδυναμεί με αύξηση της τιμής του K_p . Η απόκριση γίνεται πιο γρήγορη και μειώνεται το σφάλμα μόνιμης κατάστασης. Μεγάλη όμως τιμή του K μπορεί να προκαλέσει ταλάντωση ή ακόμη και αστάθεια.
- Μείωση της τιμής του α σε έναν FZ-PI, είναι το ίδιο με αύξηση της τιμής του ασαφούς κέρδους K_i , η απόκριση γίνεται πιο γρήγορη και μειώνεται το σφάλμα μόνιμης κατάστασης. Πολύ μικρή τιμή όμως του α μπορεί να προκαλέσει μεγάλη ταλάντωση και τείνει να αποσταθεροποιήσει το σύστημα.

Τελικά, το σετ τιμών για τα κέρδη που επιφέρουν επιθυμητά αποτελέσματα

είναι:

$$\begin{aligned} K_e &= 1 \\ T_i &= a = 0.6 \\ K &= 20 \end{aligned}$$

Figure 8: Σήμα διέγερσης, απόκριση Pi, απόκριση FZ-Pi



Με την χρήση του *BilevelMeasurements* και με γραφικό τρόπο φαίνεται η τελική ρύθμιση του ασαφούς ελεγκτή ώστε να γίνει αποδοτικότερος του αντίστοιχου γραμμικού (εικόνα 8).

	K_p	K_i	K_e	a	K	Χρόνος Ανόδου (Sec)	Ποσοστό Υπερύψωσης (%)
Pi	1.85	0.55	-	-	-	0.307	6.9
FZ-Pi	-	-	1	0.6	20	0.29	0.505

Διατηρώντας έτσι τις 2 εκ των 3 τιμών των κερδών σταθερές και αλλάζοντας την μία μπορεί πλέον να εντοπιστεί η επίδραση του κάθε κέρδους στην απόκριση του συστήματος. Οι παραμετρικές αναλύσεις ως προς a , K , K_e εμφανίζονται στις εικόνες 9, 10, 11 αντίστοιχα.

Figure 9: παραμετρική ανάλυση ως προς a

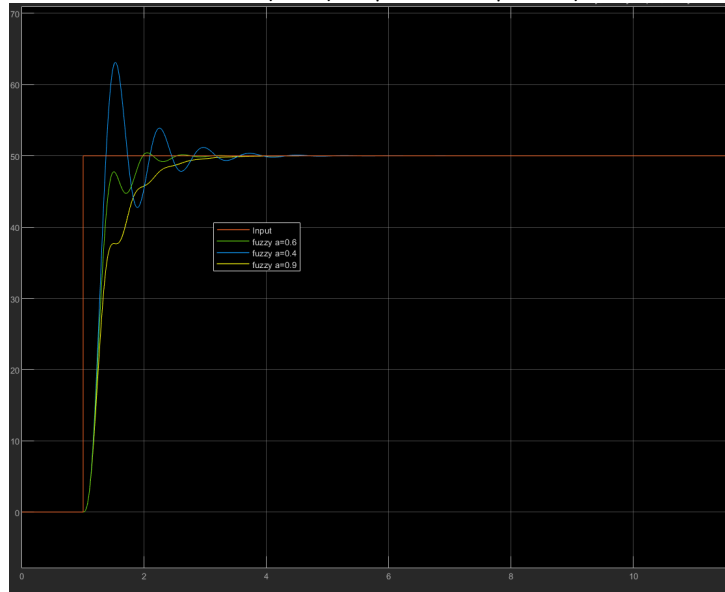


Figure 10: παραμετρική ανάλυση ως προς K

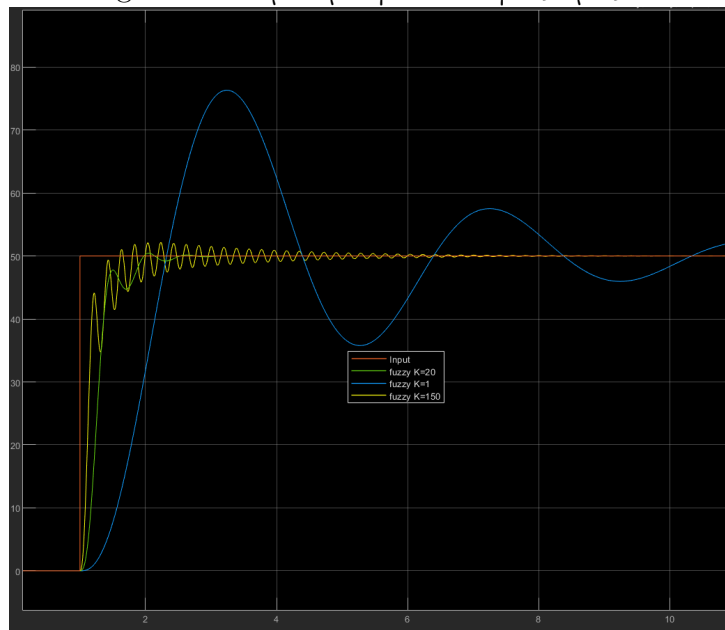
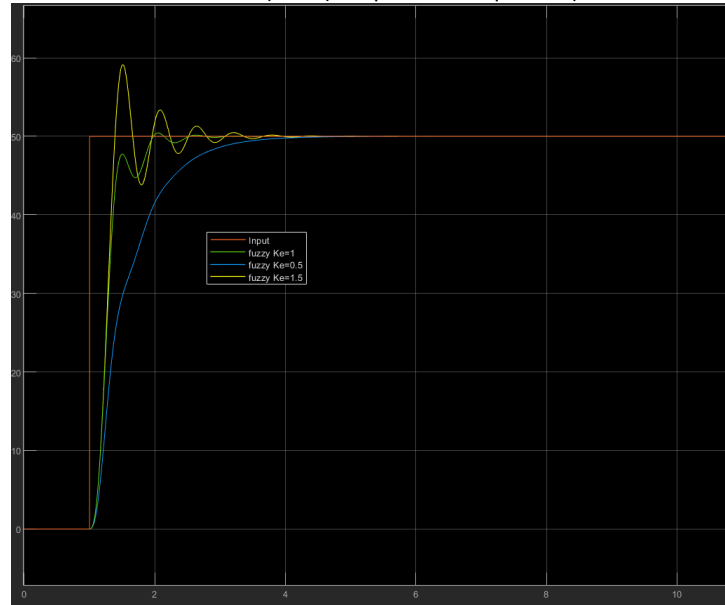


Figure 11: παραμετρική ανάλυση ως προς K_e



Λειτουργία της βάσης του ελεγκτή και συμπεράσματα

Στην συνέχεια παρουσιάζονται οι κανόνες της βάσης (εικόνες 12,13) που κατασκευάστηκαν με το περιβάλλον *FuzzyLogicDesigner* σε αρχείο με όνομα *FLC_base.fis*. Εκεί επιλέγοντας *View -> Rules* και εισάγοντας τιμές για τις δύο εισόδους e και de φαίνεται σχηματικά η λειτουργία των κανόνων συμμετοχής. Θέτοντας σαν τιμές $(e, de) = (ZR, PS) = (0, 0.25)$, η βάση λειτουργεί όπως παρουσιάζεται στις εικόνες 14, 15, 16. Στην τρίτη στήλη φαίνεται η απόκριση του κάθε κανόνα. Αυτή αποτελείται από το \min των δύο εισόδων όπως ορίζει το συνδετικό AND. Στο τελευταίο γράφημα της τρίτης στήλης βλέπουμε το μέγιστο όλων των εξόδων για την διέγερση του κάθε κανόνα σύμφωνα με το συνδετικό ALSO. Στο τελικό στάδιο η αποασαφοποίηση γίνεται με τον τελεστή centroid.

1. If (error is NV) and (de is PV) then (du is ZR) (1)
2. If (error is PV) and (de is NV) then (du is ZR) (1)
3. If (error is NL) and (de is PL) then (du is ZR) (1)
4. If (error is PL) and (de is NL) then (du is ZR) (1)
5. If (error is PM) and (de is NM) then (du is ZR) (1)
6. If (error is PS) and (de is NS) then (du is ZR) (1)
7. If (error is NS) and (de is PS) then (du is ZR) (1)
8. If (error is ZR) and (de is ZR) then (du is ZR) (1)
9. If (error is NM) and (de is PM) then (du is ZR) (1)
10. If (error is NL) and (de is PV) then (du is PS) (1)
11. If (error is NM) and (de is PV) then (du is PM) (1)
12. If (error is NS) and (de is PV) then (du is PL) (1)
13. If (error is ZR) and (de is PV) then (du is PV) (1)
14. If (error is PS) and (de is PV) then (du is PV) (1)
15. If (error is PM) and (de is PV) then (du is PV) (1)
16. If (error is PL) and (de is PV) then (du is PV) (1)
17. If (error is PV) and (de is PV) then (du is PV) (1)
18. If (error is NV) and (de is PL) then (du is NS) (1)
19. If (error is NM) and (de is PL) then (du is PS) (1)
20. If (error is NS) and (de is PL) then (du is PM) (1)
21. If (error is ZR) and (de is PL) then (du is PL) (1)
22. If (error is PS) and (de is PL) then (du is PV) (1)
23. If (error is PM) and (de is PL) then (du is PV) (1)
24. If (error is PL) and (de is PL) then (du is PV) (1)
25. If (error is PV) and (de is PL) then (du is PV) (1)
26. If (error is NV) and (de is PM) then (du is NM) (1)
27. If (error is NL) and (de is PM) then (du is NS) (1)
28. If (error is NS) and (de is PM) then (du is PS) (1)
29. If (error is ZR) and (de is PM) then (du is PM) (1)
30. If (error is PS) and (de is PM) then (du is PL) (1)
31. If (error is PM) and (de is PM) then (du is PV) (1)
32. If (error is PL) and (de is PM) then (du is PV) (1)
33. If (error is PV) and (de is PM) then (du is PV) (1)
34. If (error is NV) and (de is PS) then (du is NL) (1)
35. If (error is NL) and (de is PS) then (du is NM) (1)
36. If (error is NM) and (de is PS) then (du is NS) (1)
37. If (error is ZR) and (de is PS) then (du is PS) (1)
38. If (error is PS) and (de is PS) then (du is PM) (1)
39. If (error is PM) and (de is PS) then (du is PL) (1)
40. If (error is PL) and (de is PS) then (du is PV) (1)
41. If (error is PV) and (de is PS) then (du is PV) (1)
42. If (error is NV) and (de is ZR) then (du is NV) (1)

Figure 12: Κανόνες 1-42

43. If (error is NL) and (de is ZR) then (du is NL) (1)
44. If (error is NM) and (de is ZR) then (du is NM) (1)
45. If (error is NS) and (de is ZR) then (du is NS) (1)
46. If (error is PS) and (de is ZR) then (du is PS) (1)
47. If (error is PM) and (de is ZR) then (du is PM) (1)
48. If (error is PL) and (de is ZR) then (du is PL) (1)
49. If (error is PV) and (de is ZR) then (du is PV) (1)
50. If (error is NV) and (de is NS) then (du is NV) (1)
51. If (error is NL) and (de is NS) then (du is NV) (1)
52. If (error is NM) and (de is NS) then (du is NL) (1)
53. If (error is NS) and (de is NS) then (du is NM) (1)
54. If (error is ZR) and (de is NS) then (du is NS) (1)
55. If (error is PM) and (de is NS) then (du is PS) (1)
56. If (error is PL) and (de is NS) then (du is PM) (1)
57. If (error is PV) and (de is NS) then (du is PL) (1)
58. If (error is NV) and (de is NM) then (du is NV) (1)
59. If (error is NL) and (de is NM) then (du is NV) (1)
60. If (error is NM) and (de is NM) then (du is NV) (1)
61. If (error is NS) and (de is NM) then (du is NL) (1)
62. If (error is ZR) and (de is NM) then (du is NM) (1)
63. If (error is PS) and (de is NM) then (du is NS) (1)
64. If (error is PL) and (de is NM) then (du is NS) (1)
65. If (error is PV) and (de is NM) then (du is PM) (1)
66. If (error is NV) and (de is NL) then (du is NV) (1)
67. If (error is NL) and (de is NL) then (du is NV) (1)
68. If (error is NM) and (de is NL) then (du is NV) (1)
69. If (error is NS) and (de is NL) then (du is NV) (1)
70. If (error is ZR) and (de is NL) then (du is NL) (1)
71. If (error is PS) and (de is NL) then (du is NM) (1)
72. If (error is PM) and (de is NL) then (du is NS) (1)
73. If (error is PV) and (de is NL) then (du is PS) (1)
74. If (error is NV) and (de is NV) then (du is NV) (1)
75. If (error is NL) and (de is NV) then (du is NV) (1)
76. If (error is NM) and (de is NV) then (du is NV) (1)
77. If (error is NS) and (de is NV) then (du is NV) (1)
78. If (error is ZR) and (de is NV) then (du is NV) (1)
79. If (error is PS) and (de is NV) then (du is NL) (1)
80. If (error is PM) and (de is NV) then (du is NM) (1)
81. If (error is PL) and (de is NV) then (du is NS) (1)

Figure 13: Κανόνες 42-81

Figure 14: κανόνες 1-32 για $e=0, de=0.25$

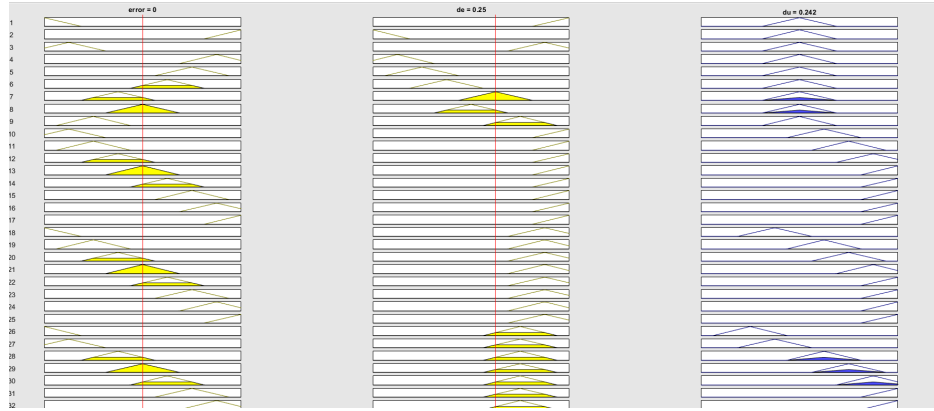


Figure 15: κανόνες 33-65 για $e=0, de=0.25$

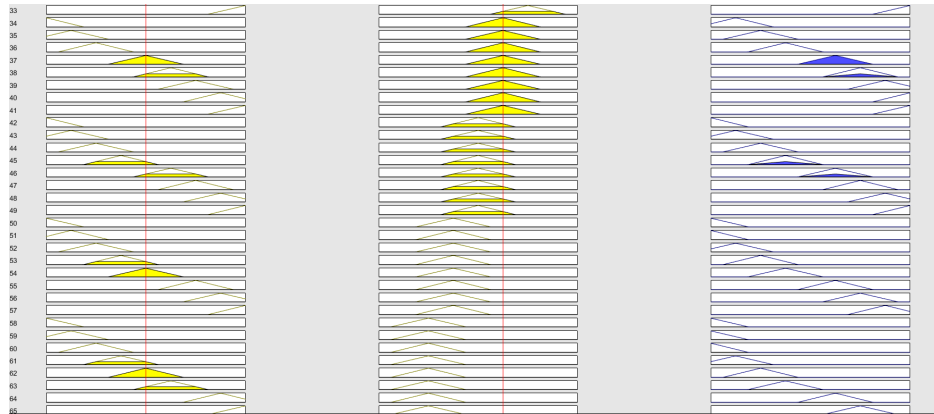
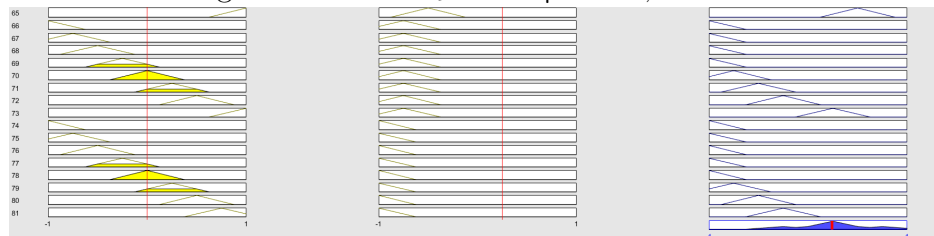


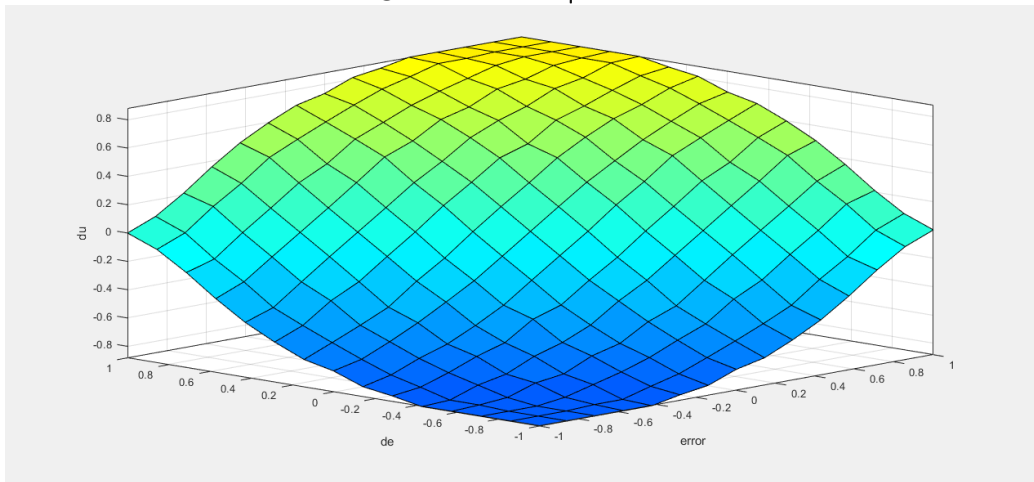
Figure 16: κανόνες 65-81 για $e=0, de=0.25$



Ερμηνεία του νόμου ελέγχου του FLC

Η παρακάτω τρισδιάστατη απεικόνιση (εικόνα 17) μας υποδεικνύει ότι και ο

Figure 17: Βάση κανόνων



πίνακας της ασαφούς βάσης κανόνων που κατασκευάστηκε σε προηγούμενο ερώτημα. Πιο συγκεκριμένα, μπορούμε να παρατηρήσουμε ότι σε χαμηλές αρνητικές τιμές σφάλματος *error* και μεταβολής σφάλματος *de*, δηλαδή σε περίπτωση που η έξοδος έχει μεγαλύτερη τιμή από την είσοδο, η έξοδος είναι εξίσου αρνητική για να μειώσει το σήμα ελέγχου και να μας επαναφέρει προς το σημείο αναφοράς. Αντίστοιχα, στις μεγάλες θετικές τιμές, η έξοδος έχει μικρότερη τιμή από την είσοδο και απομακρυνόμαστε από το σημείο αναφοράς. Έτσι, η έξοδος θα πρέπει να διορθώσει το σφάλμα και το σήμα ελέγχου να αυξηθεί. Κοντά στην κύρια διαγώνιο (γαλάζιο τμήμα), οι μεταβολές στο σήμα αναφοράς είναι μικρές (NS, PS) και απομακρυνόμενοι από την κύρια διαγώνιο οι μεταβολές γίνονται μεγαλύτερες (NL, NV) (PL, PV) με μπλε και κόκκινο χρώμα αντίστοιχα.

3.2 Σενάριο 2

Διεγείροντας το σύστημα με διαφορετικές εισόδους γίνεται αντιληπτό ότι ο ασαφής ελεγκτής επιφέρει ταχύτερη επαναφορά μηδενικού σφάλματος ενώ ταυτόχρονα το σφάλμα μόνιμης κατάστασής του είναι μικρότερο. Τα διαγράμματα και η προσομοίωση που δείχνουν τα συμπεράσματα υπάρχουν στο αρχείο *simulation_9209.slx*.

Figure 18: Τετραγωνοειδές

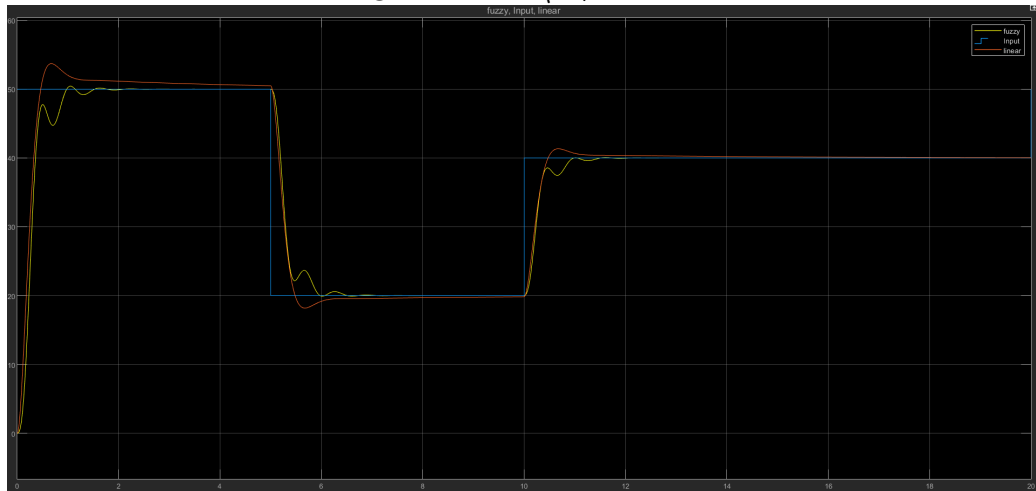


Figure 19: Ράμπα

