

Ασαφή Συστήματα

Έλεγχος αυτοκινήτου με ασαφή ελεγκτή

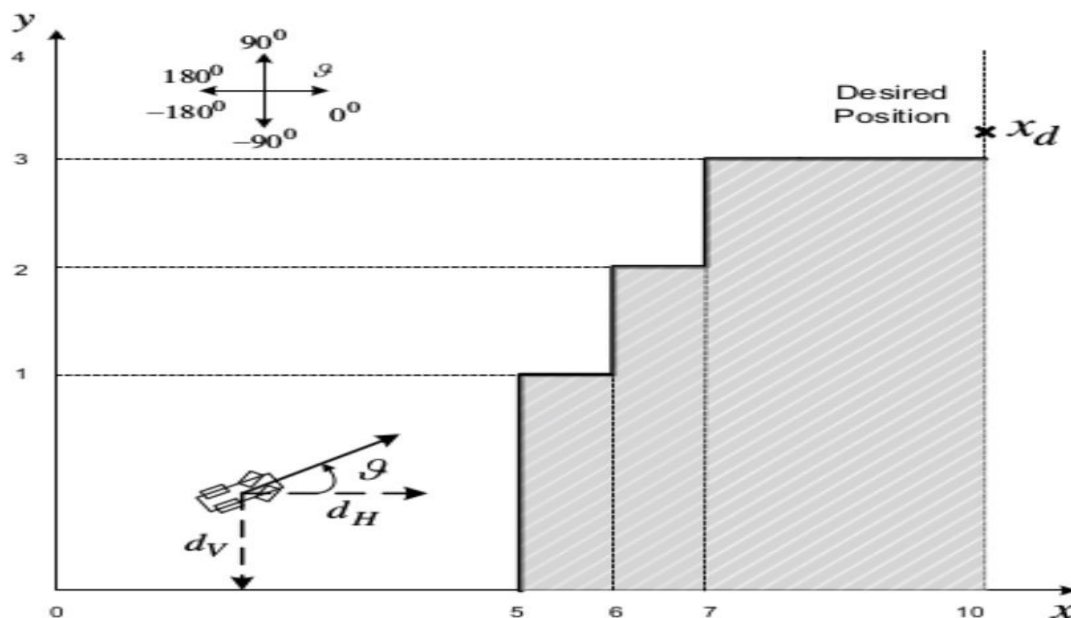
Υπατία Δάμη

AEM:8606

05/08/2019

Σκοπός εργασίας :

Σκοπός της εργασίας αυτής ήταν ο σχεδιασμός ενός ασαφούς ελεγκτή , ο οποίος είναι υπεύθυνος για τον έλεγχο ενός αυτόματου αυτοκινήτου. Συγκεκριμένα, ο ελεγκτής σχεδιάστηκε έτσι ώστε να ξεκινάει από μια συγκεκριμένη θέση $(x_0, y_0) = (4.1, 0.3)$ και να φτάνει σε μία θέση –προορισμό $(x_d, y_d) = (10, 3.2)$. ακολουθώντας μια πορεία έτσι ώστε να αποφεύγει κάποια προκαθορισμένα εμπόδια. Το όχημα κινείται με σταθερή ταχύτητα $u = 0.05 \text{ m/s}$. Διαθέτει αισθητήρες απόστασης από τα εμπόδια (d_H η οριζόντια απόσταση από το εμπόδιο και d_V η κάθετη), που δειγματοληπτούνται κάθε $T = 0.1 \text{ s}$. Για την υλοποίηση της ασαφούς λογικής του ελεγκτή χρησιμοποιήθηκε ο editor του fuzzy toolbox στο matlab, μέσω του οποίου σχεδιάστηκαν οι συναρτήσεις συμμετοχής των μεταβλητών εισόδου και εξόδου και η βάση κανόνων του συστήματος. Ως μεταβλητές εισόδου ορίζονται οι αποστάσεις d_V , d_H και η γωνία του οχήματος θ . Οι μεταβλητές εισόδου d_V και d_H παίρνουν τιμές στο κανονικοποιημένο διάστημα $[0, 1]$, το οποίο χωρίζεται με βάση την ασαφή λογική στις τιμές S (small), M (medium) και L (large). Η μεταβλητή εισόδου θ παίρνει τιμές στο διάστημα $[-180, 180]$ το οποίο χωρίζεται αντίστοιχα στις ασαφείς τιμές N (negative), ZE (zero) και P (positive). Ως μεταβλητή εξόδου ορίζεται η μεταβλητή $d\theta$ και αντιπροσωπεύει τη γωνία που χρειάζεται να στρίψει το όχημα προς αποφυγή των εμποδίων. Η μεταβλητή $d\theta$ παίρνει τιμές στο διάστημα $[-130, 130]$ και έχει ανάλογες ασαφείς τιμές με τη μεταβλητή θ .



Αρχικά σχεδιάστηκε η βάση κανόνων του ελεγκτή με βάση την εμπειρία και τη λογική. Με βάση τις μεταβλητές, τα πεδία ορισμού και τιμών τους και τη βάση κανόνων σχεδιάστηκε μέσω του fis editor ο ελεγκτής. Τέλος ακολούθησε η

συγγραφή ενός προγράμματος στο matlab, που προσομοιάζει την κίνηση του αυτοκινήτου από την αρχική του θέση μέχρι την τελική χρησιμοποιώντας τον ασαφή ελεγκτή που σχεδιάστηκε προηγουμένως.

Σχεδιασμός βάσης κανόνων:

Καθώς το σύστημά μας έχει ως εισόδους τρεις μεταβλητές με 3 πιθανές τιμές η κάθε μία, το σύνολο των κανόνων θα είναι $3^3=27$. Οι κανόνες γράφτηκαν εμπειρικά χρησιμοποιώντας την κοινή λογική. Οι τιμές επιλέχθηκαν κατάλληλα έτσι ώστε το όχημα να κινείται ευθεία ,να μην αλλάζει δηλαδή γωνία εκτός αν βρίσκεται πολύ κοντά, (σε απόσταση Small) από κάποιο εμπόδιο. Με αυτόν τον τρόπο πετυχαίνουμε τη σχεδόν εφαιπτομενική κίνηση του οχήματος δίπλα στο εμπόδιο, χωρίς όμως να το ακουμπάει καθώς όταν φτάσει σε μικρή απόσταση από αυτό εξαναγκάζεται σε στροφή περίπου 90 μοιρών. Μερικά παραδείγματα κανόνων είναι τα παρακάτω:

If dV is S AND dH is S AND theta is ZE THEN dtheta is P

If dV is L AND dH is L AND theta is P THEN dtheta is N

If dV is S AND dH is M AND theta is ZE THEN dtheta is ZE

If dV is M AND dH is M AND theta is N THEN dtheta is N

Από τη φύση των κανόνων καταλαβαίνουμε πως το όχημα θα προσπαθήσει να κρατήσει μια σταθερή ευθεία πορεία (γωνία ZE) εκτός αν έρθει πολύ κοντά σε κάποιο εμπόδιο και εξαναγκαστεί να στρίψει .

Υλοποίηση στο Matlab:

Η κίνηση του οχήματος προσομοιώθηκε στο matlab , με το script project.m . Για τον υπολογισμό διαφόρων παραμέτρων δημιουργήθηκαν κάποια scripts και functions που θα περιγραφούν παρακάτω.

Estimate_distance:

Η συνάρτηση αυτή δέχεται ως εισόδο την τρέχουσα θέση του αυτοκινήτου και υπολογίζει τις αποστάσεις dV και dH από τα εμπόδια .

Normalize_theta:

Η συνάρτηση αυτή κανονικοποιεί την εκάστοτε τιμή της γωνίας θ, από το διάστημα $[-180,180]$ στο διάστημα $[0,1]$.

Update_position:

Η συνάρτηση αυτή δέχεται ως εισόδους την τελευταία θέση του οχήματος, την ταχύτητά του , τον χρόνο δειγματοληψίας και την γωνία που χρειάζεται να στρίψει σύμφωνα με τον ασαφή ελεγκτή, και υπολογίζει τη νέα θέση του .

Car_control.fis:

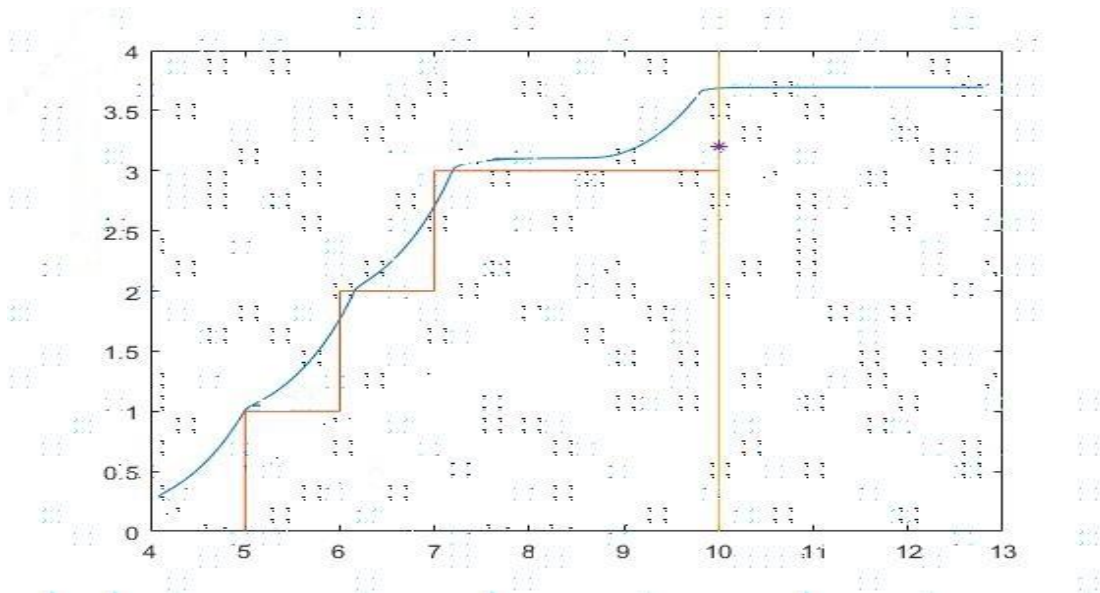
Το αρχείο αυτό είναι ο κώδικας που προκύπτει από τον fis editor και περιέχει όλες τις πληροφορίες για τον ασαφή ελεγκτή. (Εύρη τιμών, συναρτήσεις συμμετοχής, βάση κανόνων)

Project:

Αυτό είναι το βασικό script του προγράμματος .Αρχικά ορίζονται η αρχική θέση του οχήματος, η επιθυμητή θέση ,η ταχύτητα , ο χρόνος δειγματοληψίας και ο αριθμός επαναλήψεων. Έπειτα σε ένα for loop για τις επαναλήψεις που ορίστηκαν εκτιμάται σε κάθε επανάληψη η γωνία που πρέπει να στρίψει το όχημα χρησιμοποιώντας τα παραπάνω scripts και το αντικείμενο “fis” που δημιουργήσαμε στον fis editor .Τέλος προσομοιώνεται η πορεία του οχήματος με χρήση της συνάρτησης plot ,καθώς και η γραμμή που παριστά το εμπόδιο.

Σχολιασμός αποτελεσμάτων:

Για τη συμπεριφορά του ασαφούς ελεγκτή στον έλεγχο του οχήματος καταγράφηκαν αποτελέσματα από την αρχική εκτίμηση του συστήματος και έπειτα από κάποιες αλλαγές στον κώδικα του προγράμματος προέκυψαν τα τελικά αποτελέσματα που συμπίπτουν με τα επιθυμητά.Η αρχική εκτίμηση της βάσης κανόνων αποδείχθηκε σωστή, καθώς και των διαστημάτων τιμών των μεταβλητών εισόδου και εξόδου. Τα αποτελέσματα της πρώτης δοκιμής φαίνονται στην εικόνα 1.



Εικόνα 1

Το όχημα ακολουθεί την πορεία αυτή διότι οι τιμές των dV και dH που επιστρέφει η συνάρτηση `estimate_distance` κυμαίνονται στο διάστημα $[0,1]$, οπότε με την κανονικοποίηση που ακολουθεί παίρνουν τιμές της τάξης 10^{-2} , πράγμα που σημαίνει πως θα ανήκουν πάντα στο φάσμα τιμών της ασαφούς μεταβλητής $S(\text{small})$. Όπως φαίνεται και στην εικόνα το σύστημα δίνει τιμή S σε αποστάσεις μεγαλύτερες από αυτές που θέλουμε να πετύχουμε ώστε να κινείται το όχημα κοντά στο εμπόδιο. Στις τιμές dV και dH προστίθεται μια σταθερή τιμή (3,5), αλλά και πάλι δεν καλύπτεται όλο το φάσμα τιμών. Η λύση σε αυτό το πρόβλημα δόθηκε στη συνάρτηση `estimate_distance`, όπου προστέθηκε το παρακάτω κομμάτι κώδικα:

```
if (0<=dv<=0.1)
    dv=dv+1;
elseif (0.1<dv<=0.2)
    dv=dv+2;
elseif (0.2<dv<=0.3)
    dv=dv+3;
else
    dv=dv+4;
end

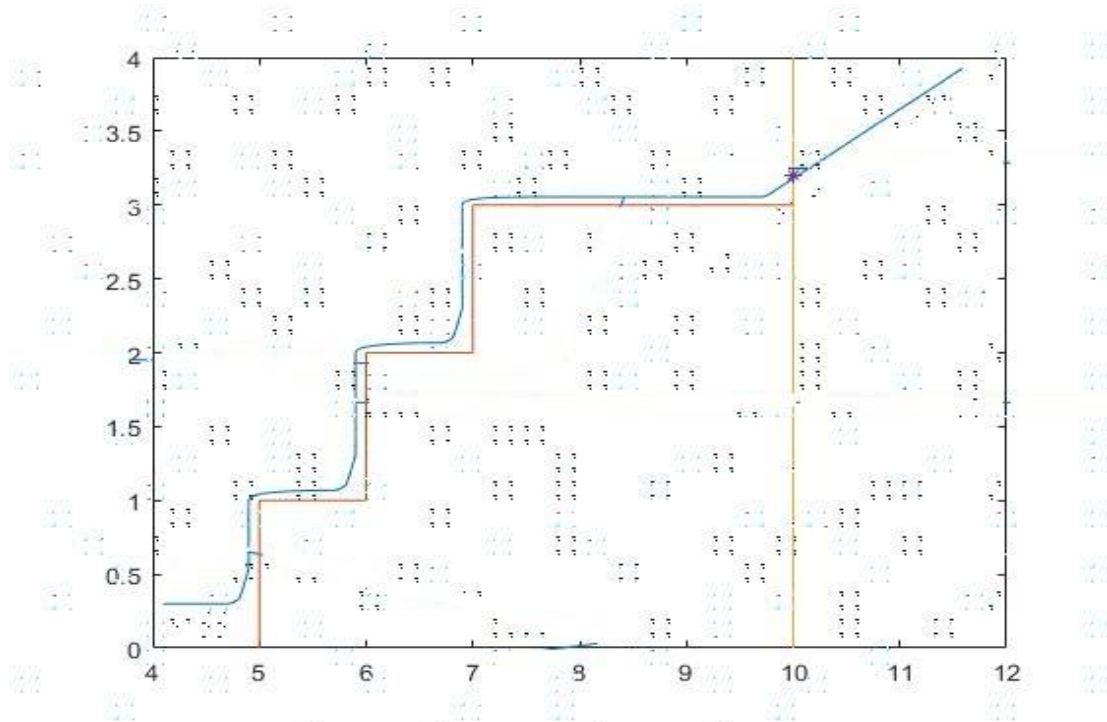
if (0<=dh<=0.1)
    dh=dh+1;
elseif (0.1<dh<=0.2)
    dh=dh+2;
elseif (0.2<dh<=0.3)
    dh=dh+3;
elseif (0.3<dh<=0.4)
    dh=dh+4;
elseif (0.4<dh<=0.5)
    dh=dh+5;
elseif (0.5<dh<=0.6)
    dh=dh+6;
elseif (0.6<dh<=0.7)
```

```

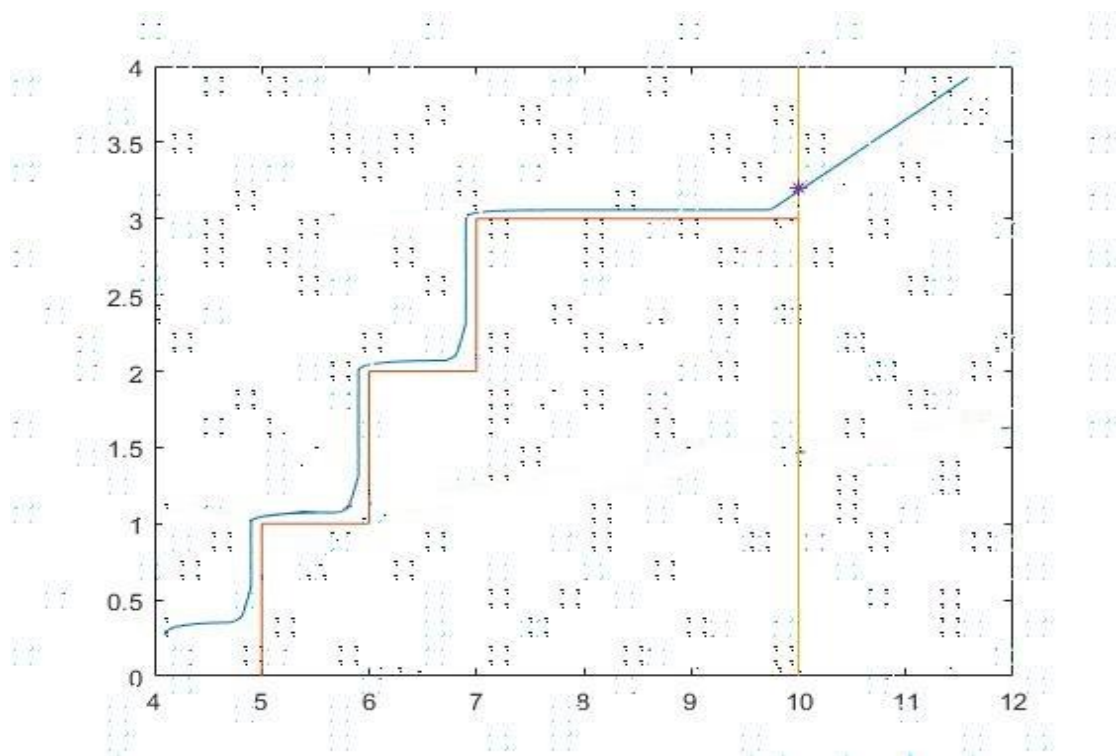
        dh=dh+7;
elseif (0.7<dh<=0.8)
        dh=dh+8;
elseif (0.8<dh<=0.9)
        dh=dh+9;
else
        dh=dh+10;
end

```

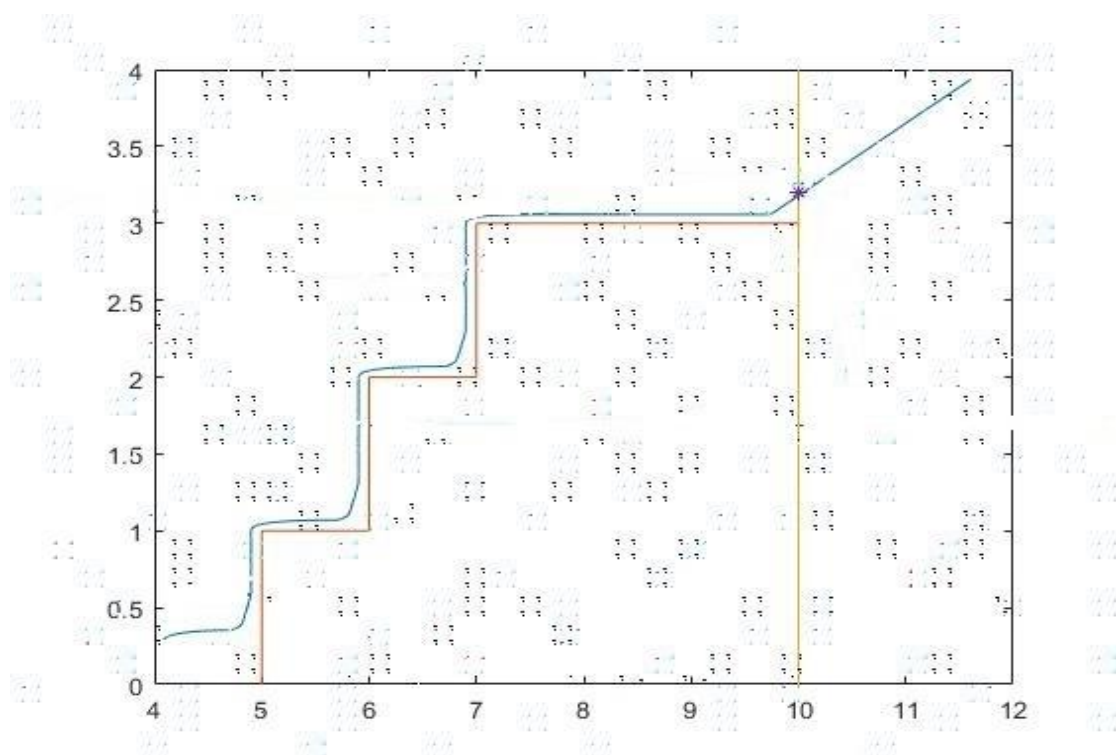
Με αυτόν τον τρόπο ρυθμίζουμε τις τιμές των μεταβλητών έτσι ώστε να ανεβούν μια τάξη μεγέθους και να καλύπτουν όλο το φάσμα τιμών των μεταβλητών εισόδου. Αντί να προσθέτουμε μια σταθερή τιμή, ελέγχουμε τις τιμές των dH και dV και αναλόγως προσθέτουμε την κατάλληλη τιμή για να γίνει σωστή κλιμακοποίηση. Έτσι παραδείγματος χάρη αν πάρουμε από την συνάρτηση `estimate_distance` την τιμή $dH=0.7$, προσθέτουμε το 7 έτσι ώστε όταν στη συνέχεια κανονικοποιηθεί (διαίρεση με το 10) να προκύψει η επιθυμητή τιμή 0.7, η οποία ανήκει στο διάστημα τιμών $[0.1]$ των μεταβλητών εισόδου και πλέον θεωρείται M και L . Τα τελικά αποτελέσματα φαίνονται στις Εικόνες 2,3 και 4 για 0,-45 και -90 μοίρες αντίστοιχα.



Εικόνα 2



Εικόνα 3



Εικόνα 4

Όπως φαίνεται στα αποτελέσματα της τελευταίας δοκιμής η πορεία του οχήματος είναι ικανοποιητική .Το όχημα πορεύεται εφαιπτομενικά με τα εμπόδια χωρίς να τα ακουμπάει και τελικά φτάνει με ακρίβεια στην επιθυμητή θέση .Η αρχική εκτίμηση των κανόνων επομένως ήταν σωστή και με την διόρθωση στον κώδικα ο ελεγκτής έχει την επιθυμητή λειτουργία.Το όχημα φτάνοντας στην επιθυμητή θέση δεν στρίβει χάρη σε παρεμβολή στον κώδικα που φαίνεται παρακάτω έτσι ώστε να τερματιστεί η διαδικασία

```
if (h>9.8)
    dh=4.3;
    dv=2;
end
```

Οι τιμές αυτές που δίνουμε εξαναγκαστικά στις μεταβλητές εισόδου οδηγούν το όχημα σε μικρή στροφή και επομένως καταλήγει στην επιθυμητή θέση.