

## MOVING TO A POSE

[Ψηφιακός έλεγχος  
αυτοκινήτου και  
μετακίνηση του σε  
επιθυμητή θέση.]

Αναστασόπουλος  
Νικόλαος  
Λιτσάκης  
Νικόλαος  
[EE883]

# ΨΗΦΙΑΚΟΣ ΕΛΕΓΧΟΣ

Ακαδημαϊκό Έτος 2016/17

Πανεπιστήμιο Πατρών

Τμήμα Ηλεκτρολόγων Μηχανικών και  
Τεχνολογίας Υπολογιστών





# Ψηφιακός Έλεγχος

## Πρότζεκτ

### Μέλη ομάδος :

- Αναστασόπουλος Νικόλαος 228268
- Λιτσάκης Νικόλαος 228390

### Περιεχόμενα :

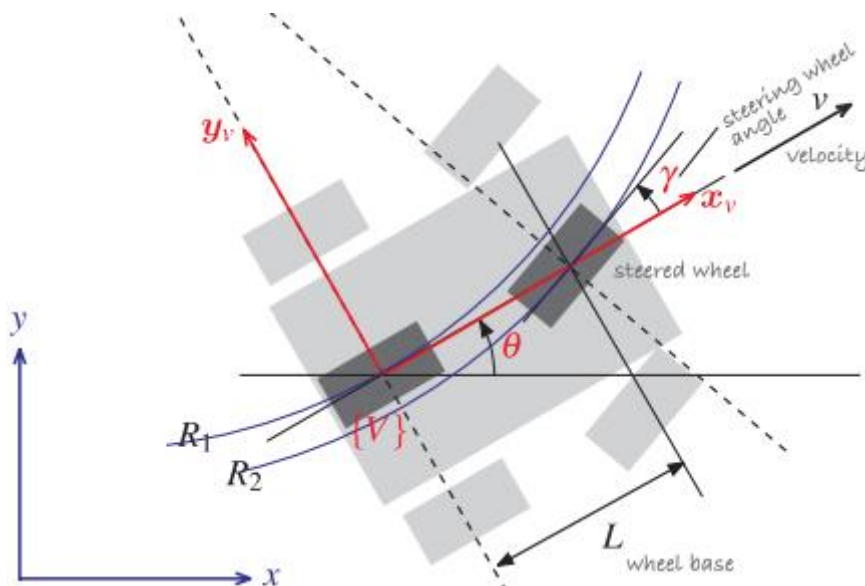
1) Συνοπτική εισαγωγή στο περιεχόμενο του πρότζεκτ . . . . .	3
2) Περιγραφή του συστήματος . . . . .	3
3) Οργάνωση του τρόπου επίλυσης . . . . .	4
4) Υπολογισμός περιόδου δειγματοληψίας . . . . .	5
5) PID έλεγχος DC Motor . . . . .	6
6) Αλγόριθμος παρακολούθησης . . . . .	9
7) Προσομοίωση ολικού συστήματος . . . . .	13
8) Συμπεράσματα . . . . .	15
9) Παράρτημα Α, ανάλυση κώδικα MATLAB . . . . .	16
10) Παράρτημα Β, θεωρητικό υπόβαθρο . . . . .	19
11) Βιβλιογραφία . . . . .	21

## 1) Περιεχόμενο του πρότζεκτ

Σκοπός αυτού του πρότζεκτ είναι ο ψηφιακός έλεγχος της θέσης ενός αυτοκινήτου. Πιο αναλυτικά το αυτοκίνητο θα πρέπει να έχει μία αρχική κατάσταση  $(X_0, Y_0, \theta_0, V_0)$  όπου  $X_0, Y_0$  είναι οι συντεταγμένες του αυτοκινήτου στο καρτεσιανό επίπεδο συντεταγμένων,  $\theta_0$  η γωνία του οχήματος μετρώντας αριστερόστοφα από τον Οχ άξονα και  $V_0 = 0$  η αρχική ταχύτητα του οχήματος. Το υπό έλεγχο όχημα θεωρείται ότι έχει ένα DC Motor. Τελικός στόχος είναι το αυτοκίνητο να μπορεί να συγκλίνει ασυμπτωτικά στην κατάσταση  $(X_f, Y_f, \theta_f, V_f)$  όπου  $X_f, Y_f, \theta_f$  είναι η τελική θέση του οχήματος στο καρτεσιανό επίπεδο και  $V_f = 0$  είναι η ταχύτητα του οχήματος στην τελική θέση (μηδενική ταχύτητα για ασυμπτωτική σύγκλιση).

## 2) Περιγραφή του συστήματος

Το αυτοκίνητο που θέλουμε να ελέγξουμε περιγράφεται από τις εξής καταστατικές εξισώσεις :  $\dot{Z} = A * Z + B * U$  όπου το διάνυσμα  $Z = [x \ y \ \theta]^T$ , ο πίνακας  $A$  είναι μηδενικός, ο πίνακας των εισόδων  $B = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix}$  και οι είσοδοι θεωρούνται  $U = \begin{bmatrix} \bar{u} \\ \gamma \end{bmatrix}$ . Στο τελευταίο διάνυσμα το  $\bar{u}$  είναι η διανυσματική ταχύτητα του οχήματος και  $\gamma$  είναι η γωνία των τροχών σε σχέση με τον άξονα του αυτοκινήτου όπως φαίνεται και στο σχήμα παρακάτω :



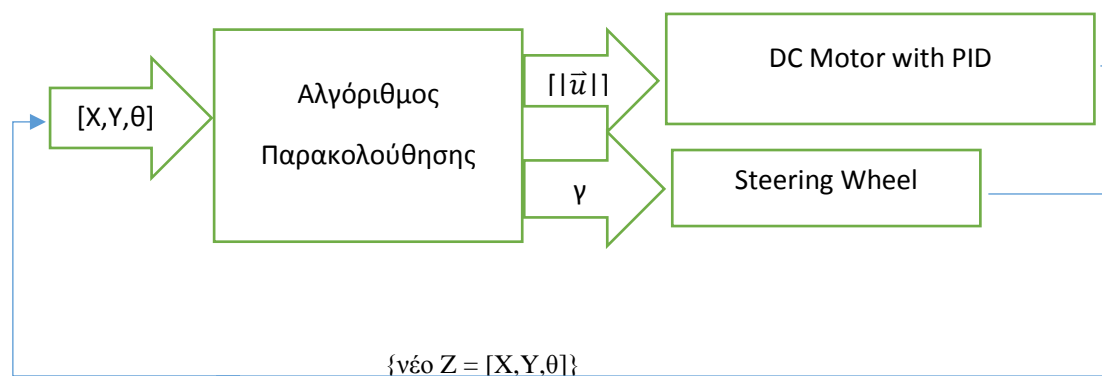
Η γωνία  $\gamma$  ελέγχεται άμεσα μιας και θεωρείται αμελητέα η καθυστέρηση να στρίψουν οι τροχοί, ενώ όσον αφορά το μέτρο του διανύσματος της ταχύτητας ο έλεγχος είναι έμμεσος μέσω του ελέγχου των στροφών του κινητήρα το οποίο περιγράφεται στη αντίστοιχη παράγραφο. Σημείο αναφοράς στην θέση του οχήματος είναι η το μέσο του άξονα των μπροστινών τροχών. Επίσης, οι παραπάνω καταστατικές εξισώσεις αφορούν την επιτρεπτή περιοχή λειτουργίας. Πιο αναλυτικά, επειδή δεν

ασχολούμαστε με ταχύτητες η μέγιστη ταχύτητα του οχήματος είναι 30 km/h και η γωνία  $\gamma$  δεν μπορεί να πάρει απόλυτη τιμή μεγαλύτερη των  $40^\circ$  μιας και στα αυτοκίνητα υπάρχουν όρια στην γωνία στριψίματος.

Για τον DC Motor ισχύει ότι μπορεί να προσεγγιστεί με ένα δευτεροβάθμιο σύστημα με δύο πόλους και κανένα μηδενικό όσον αφορά την σχέση εισόδου εξόδου όταν ελέγχουμε την τάση της εισόδου. Άρα μπορούμε να προσεγγίσουμε την συνάρτηση μεταφοράς  $G(s) = \frac{b_1}{a_1*s^2+a_2*s+a_3}$ , όπου το  $b_1 = K_t$  η σταθερά ροπής (torque constant) του DC Motor. Το  $a_1 = R * B_m * T_m * T_e$ , όπου  $R$  είναι η αντίσταση των τυλιγμάτων του DC Motor,  $B_m$  είναι η τριβή στον κινητήρα και  $T_m$ ,  $T_e$  είναι μηχανική και ηλεκτρική σταθερά του κινητήρα. Το  $a_2 = R * B_m * (T_m + T_e)$  και το  $a_3 = (R * B_m) + (T_m * T_e)$ . Έτσι, επιλέγοντας τυπικές τιμές  $K_t = 0.01 \frac{N*m}{A}$ ,  $R = 5 \Omega$ ,  $B_m = 0.1 \frac{N*m*s}{rad}$ ,  $T_m = 0.1 s$  και  $T_e = 0.5 s$  καταλήγουμε στην εξής συνάρτηση μεταφοράς:  $\frac{\theta(s)}{V(s)} = \frac{0.02}{0.025*s^2+0.3*s+0.5}$  (Αν δεν φαίνεται επαρκώς, στην συνάρτηση ορίζεται η παραγωγός του  $\theta$ ).

### 3) Οργάνωση του τρόπου επίλυσης

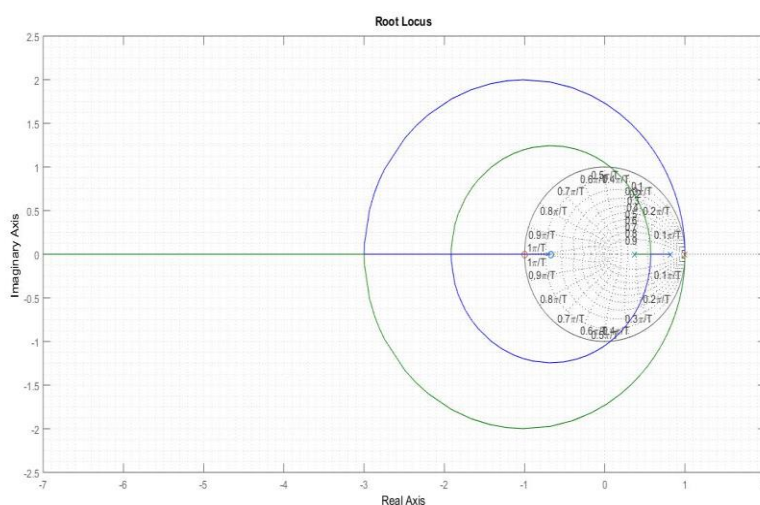
Αρχικά, για το πλήρες σύστημα, εφαρμόζουμε έναν αλγόριθμο παρακολούθησης, ο οποίος επιστρέφει το μέτρο της ταχύτητας  $|\vec{u}|$  του αυτοκινήτου την δεδομένη στιγμή που δειγματοληπτούμε και την γωνία  $\gamma$ . Η τιμή της γωνίας εφαρμόζεται στιγμιαία (υπόθεση) ενώ η ταχύτητα στέλνεται σε μία πηγή τάσης όπου γεννά το σήμα  $r(t)$  (Volts) για τον υπό έλεγχο κινητήρα. Αυτή η οργάνωση επιλέχθηκε σύμφωνα με την αρχή του διαχωρισμού, όπου αν κάθε υποσύστημα είναι ευσταθές, παρατηρήσιμο και ελέγξιμο (καταστατικές εξισώσεις και συνάρτηση μεταφοράς της παραγράφου (2)), τότε και το πλήρες σύστημα έχει αυτές τις ιδιότητες. Η απόδειξη της ελεγχιμότητας των καταστατικών εξισώσεων βρίσκονται στην αντίστοιχη παράγραφο του αλγορίθμου παρακολούθησης. Τελικά, το δομικό διάγραμμα του ελεγχόμενου συστήματος είναι το εξής:



Το νέο διάνυσμα κατάστασης υπολογίζεται σύμφωνα με την περίοδο δειγματοληψίας και τα υποσυστήματα είναι συγχρονισμένα μεταξύ τους.

#### 4) Υπολογισμός περιόδου δειγματοληψίας

Για τον υπολογισμό της περιόδου δειγματοληψίας έπρεπε να λάβουμε υπ' όψιν μας δύο βασικές παραμέτρους. Αρχικά, όσο μεγαλύτερη είναι η περίοδος δειγματοληψίας, τόσο μικρότερη πρέπει να είναι η ταχύτητα του εφαρμοζόμενου νόμου ελέγχου από τον αλγόριθμο παρακολούθησης έτσι ώστε να μην κάνει περιττές κινήσεις το όχημα την ώρα που δεν υπάρχει δειγματοληψία της θέσης (υπάρχει κίνδυνος αστάθειας όπου αναλύεται σε επόμενη ενότητα). Επίσης, όσο μικρότερη είναι περίοδος δειγματοληψίας τόσο πιο κοντά είναι στον μοναδιαίο κύκλο (εσωτερικά αυτού) οι πόλοι της συνάρτησης μεταφοράς του DC Motor. Παρακάτω φαίνεται ο γεωμετρικός τόπος των ριζών για περιόδους δειγματοληψίας  $T_s = 0.1, 10^{-6}$  s :

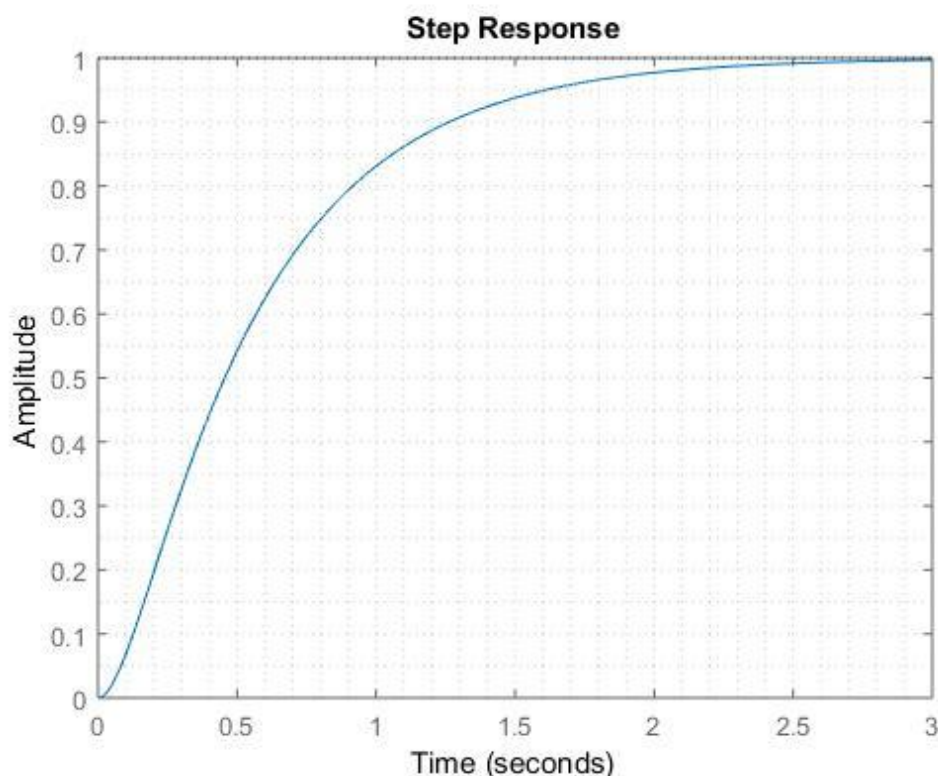


Η συνάρτηση που διακριτοποιούμε είναι η συνάρτηση μεταφοράς της παραγράφου 2 :  $G(s) = \frac{\theta(s)}{V(s)} = \frac{0.02}{0.025*s^2+0.3*s+0.5}$  .

Η καμπύλη που ξεκινά από τα όρια του μοναδιαίου κύκλου είναι για την μικρότερη περίοδο δειγματοληψίας. Κάθε γεωμετρικός τόπος για την  $G(s)$  με περίοδο δειγματοληψίας  $T_s \in (10^{-1}, 10^{-6})$  ανήκει εντός του χώρου που ορίζουν αυτοί οι δύο γεωμετρικοί τόποι. Συμπεραίνουμε λοιπόν, ότι χρειαζόμαστε ‘καλύτερο’ ελεγκτή όσο μικρότερη είναι η περίοδος δειγματοληψίας. Καταλήγουμε λοιπόν σε μία μέση λύση και ορίζουμε περίοδο δειγματοληψίας που θα χρησιμοποιείται από εδώ και στο εξής  $T_s = 0.0001$  s. Με αυτή την περίοδο δειγματοληψίας ο νόμος ελέγχου (μαζί με το όριο των 30 km/h που αναφέρθηκε στην παράγραφο(2)) δεν θα επιβάλλει το όχημα να διανύσει απόσταση μεγαλύτερη του ενός εκατοστού (cm) μεταξύ διαδοχικών δειγματοληψιών της θέσης. Αυτό είναι αναγκαίο για την σθεναρότητα του επιβαλλόμενου νόμου ελέγχου. Επίσης, δεν χρειάζεται να επιβάλλουμε μεγάλα κέρδη ( $>10^3$ ) στον ελεγκτή του DC Motor κάτι που θα ήταν εκτός λογικής πραγματικών μεγεθών για έναν PID Controller.

## 5) PID έλεγχος DC Motor

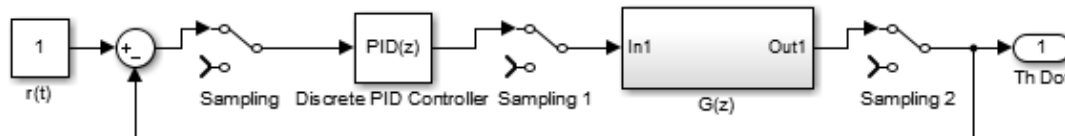
Δεδομένης της συνάρτησης μεταφοράς  $G(s) = \frac{\theta(s)}{V(s)} = \frac{0.02}{0.025s^2 + 0.3s + 0.5}$  και της περιόδου δειγματοληψίας  $T_s = 0.0001$  από προηγούμενες παραγράφους, καταλήγουμε στον εξής μετασχηματισμό Z για την  $G(z)$  με την μέθοδο του Zero Order Hold :  $G(z) = \frac{\theta(z)}{V(z)} = \frac{10^{-7}z + 10^{-7}}{z^2 - 1.999z + 0.9989}$  με βηματική απόκριση την παρακάτω :



Παρατηρούμε λοιπόν ότι η απόκριση είναι πάρα πολύ αργή σε σχέση με τον αλγόριθμο παρακολούθησης ο οποίος επεξεργάζεται τα δεδομένα και δειγματοληπτεί ενώ το σύστημα όπως φαίνεται στις πρώτες στιγμές της γραφικής παράστασης βρίσκεται ακόμα σε αδράνεια. Από τον γεωμετρικό τόπο των ριζών προηγούμενου ερωτήματος καταλαβαίνουμε ότι ανεξαρτήτως περιόδου δειγματοληψίας δεν μπορούμε να τοποθετήσουμε τους πόλους κοντά στο σημείο  $O(0,0)$  του μιγαδικού επιπέδου Z όπου η απόκριση είναι ταχύτερη. Καταλήγουμε λοιπόν στον PID έλεγχο και ορίζουμε τις εξής προδιαγραφές :

1. Τάχιστη απόκριση με αποδεκτές τιμές σταθερών  $K_p$ ,  $K_I$ ,  $K_d$
2. Μηδενική υπερύψωση (ή βύθιση κατά την εκκίνηση αντίστοιχα) διότι δημιουργούνται κίνδυνοι ευστάθειας του αλγόριθμου παρακολούθησης
3. Πολύ μικρό σφάλμα μόνιμης κατάστασης (δεν επηρεάζει πολύ τον αλγόριθμο μιας και ο νόμος ελέγχου δημιουργεί μία χρονικά μεταβαλλόμενη τροχιά στον τρισδιάστατο χώρο του διανύσματος  $Z = [X, Y, \theta]$ )

Αρχικά, αποφασίζουμε να υλοποιήσουμε έναν διακριτό PID ελεγκτή και όχι έναν ελεγκτή συνεχούς χρόνου τον οποίο εφαρμόζουμε στο σύστημα και ύστερα να διακριτοποιήσουμε το ολικό νέο σύστημα κλειστού βρόχου. Το μπλοκ διάγραμμα του DC Motor λοιπόν μετασχηματίζεται στο παρακάτω :



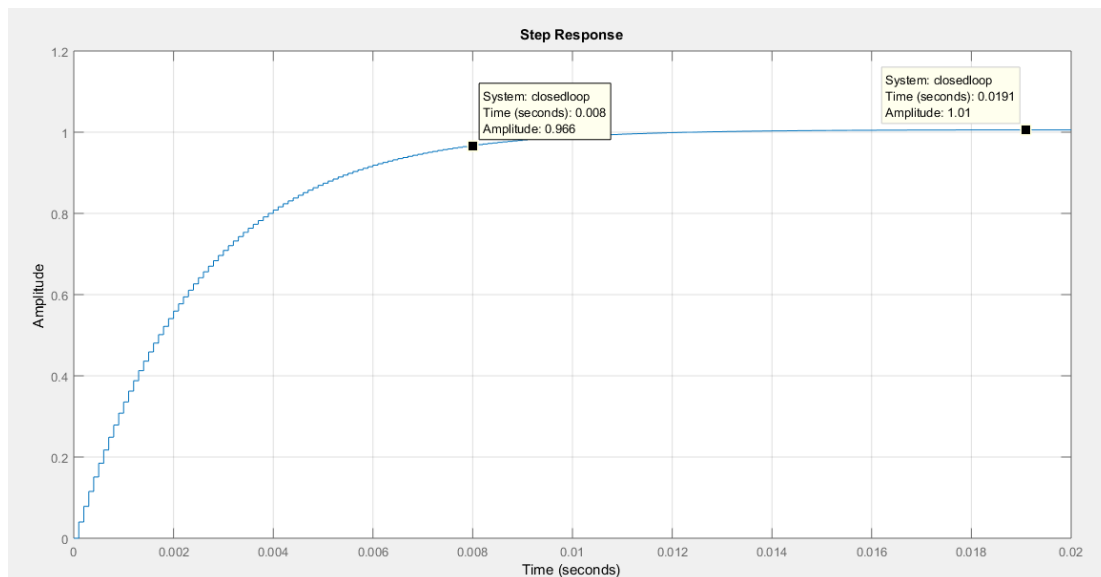
Στο σχήμα  $r(t)$  συμβολίζεται η επιθυμητή ταχύτητα και  $Th \text{ Dot}$  συμβολίζεται η γωνιακή ταχύτητα του κινητήρα.

Χρησιμοποιώντας μεθόδους εξαντλητικής αναζήτησης, αναζητήσαμε τα τρία κέρδη του PID Controller ξεκινώντας από το πεδίο του Laplace και μετασχηματίζοντας τον ελεγκτή με την μέθοδο του Tustin (1). Ο PID ελεγκτής που ταίριαζε καλύτερα στις προαναφερθείσες προδιαγραφές μας είναι ο εξής :

- Πεδίο του Laplace :  $PID(s) = 300 + \frac{400}{s} + 20 * s$
- Μιγαδικό επίπεδο Z :  $PID(z) = \frac{4.003 \cdot 10^5 - 8 \cdot 10^5 + 3.997 \cdot 10^5}{z^2 - 1}$

Οι τιμές του PID μπορεί να φαίνονται μεγάλες αλλά είναι ανηγμένες στο 1 Volt ενώ οι κινητήρες των αμαξιών λειτουργούν σε τάξεις τον kVolt.

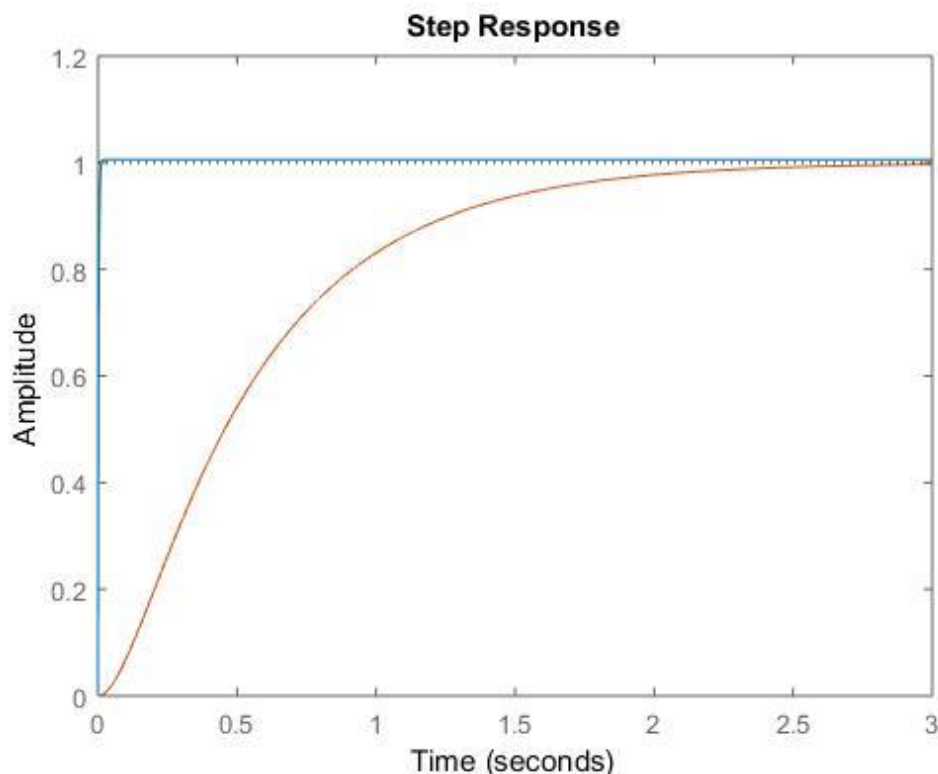
Η βηματική απόκριση του κλειστού συστήματος με τον παραπάνω ελεγκτή φαίνεται παρακάτω :



Παρατηρούμε λοιπόν, ένα αμελητέο σφάλμα μόνιμης κατάστασης, μηδενική υπερύψωση και ότι ο κινητήρας τείνει να πάρει την τελική του τιμή σε λιγότερο από 10 ms. Άρα πετύχαμε τις προδιαγραφές που ορίσαμε παραπάνω. Συγκριτικά με το σύστημα ανοιχτού βρόχου οι αποκρίσεις φαίνονται στην επόμενη σελίδα :



Σύγκριση απόκρισης κινητήρα ανοιχτού βρόχου και κλειστού βρόχου )



Η διαφορά αυτή είναι η μέγιστη που μπορούμε να πετύχουμε με τόσο μεγάλο φορτίο (που ορίσαμε στην συνάρτηση μεταφοράς του DC Motor) χωρίς να καταφεύγουμε σε εξωπραγματικά νούμερα.

Η συνάρτηση μεταφοράς, του κλειστού συστήματος λοιπόν (PID και DC Motor) είναι η εξής:  $G_{ClosedLoop}(z) = \frac{0.04001*z^3 - 0.03997*z^2 - 0.03998*z + 0.03994}{z^4 - 1.934*z^3 - 0.06395*z^2 + 1.934*z - 0.936}$ . Το σύστημα είναι ευσταθές μιας και έχει περισσότερους πόλους απ'ότι μηδενικά και παρατηρούμε φυσιολογικούς συντελεστές για ένα μηχανικό σύστημα. Η παραπάνω συνάρτηση μεταφοράς έχει πόλους στα σημεία (1, -1, 0.9673 + 0.0193\*j, 0.9673 - 0.0193\*j). Όλοι οι πόλοι είναι ευσταθείς αφού έχουν μέτρο μικρότερο του ένα. Επίσης, ο πόλος στο 1, κάνει το σύστημα τύπου 1 δηλαδή το κλειστό σύστημα τείνει ασυμπτωτικά να έχει μηδενικό σφάλμα μόνιμης κατάστασης.

(1) Η μέθοδος Tustin επιλέχθηκε για καλύτερη αριθμητική ακρίβεια αν και με zoh μέθοδο τα αποτελέσματα είναι παρόμοια



## 6) Αλγόριθμος παρακολούθησης

Όπως αναφέρθηκε σε προηγούμενη παράγραφο, η κατάσταση του συστήματος ανοιχτού βρόχου περιγράφεται από τις εξής καταστατικές εξισώσεις :

$$\dot{Z} = A * Z + B * U \text{ όπου το διάνυσμα } Z = [x \quad y \quad \theta]^T, \text{ ο πίνακας } A \text{ είναι μηδενικός,}$$

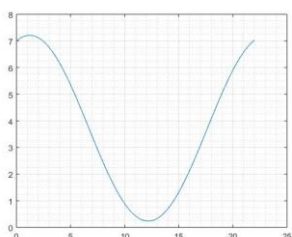
$$\text{ο πίνακας των εισόδων } B = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \text{ και οι εισοδοι θεωρούνται } U = \begin{bmatrix} \vec{u} \\ \gamma \end{bmatrix}.$$

Αρχικά, το σύστημα είναι πλήρως ελέγξιμο. Αυτό δεν μπορεί να αποδειχθεί ευθέως με την μήτρα ελεγχσιμότητας αλλά αποδεικνύεται εύκολα αν αναλογιστούμε ότι το  $\theta$ , εξαρτάται μόνο από το  $\gamma$  (είσοδος) άρα μπορεί να πάρει οποιαδήποτε τιμή. Για τα  $x, y$  τα πράγματα είναι λίγο πιο σύνθετα. Τα  $x, y$  είναι ελέγξιμα διότι ενώ και τα δύο εξαρτώνται από ένα διανυσματικό μέγεθος, στον πίνακα  $B$  δεν υπάρχουν σταθερές τιμές αλλά δύο συναρτήσεις. Αυτές οι συναρτήσεις ( $\cos \theta, \sin \theta$ ) είναι κάθετες μεταξύ τους ( $\|\cos \theta, \sin \theta\| = 0$ ) άρα οι μεταβλητές κατάστασης  $x, y$  είναι γραμμικώς ανεξάρτητες μεταξύ τους και μπορούν να βρεθούν σε οποιαδήποτε κατάσταση.

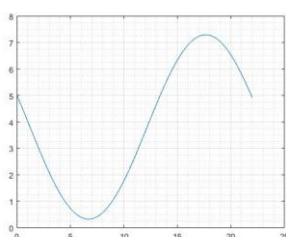
Για την παρατηρησιμότητα του συστήματος τα πράγματα είναι πιο απλά μιας και η κατάσταση  $Z = [x \quad y \quad \theta]^T$  είναι και η έξοδος του συστήματος μας την οποία θέλουμε να συγκλίνει ασυμπτωτικά στην κατάσταση  $[X_f, Y_f, \theta_f]$ .

Η βηματική απόκριση του συστήματος για  $|\vec{u}| = 100 \text{ m/s}$  (μεγάλη ταχύτητα για πιο γρήγορη προσομίωση), γωνία  $\gamma = -35^\circ$  και αρχική κατάσταση  $[x \quad y \quad \theta]^T = [7 \quad 5 \quad -35]$  φαίνεται στην επόμενη σελίδα :

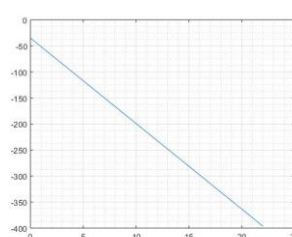
$$x_1 = X :$$



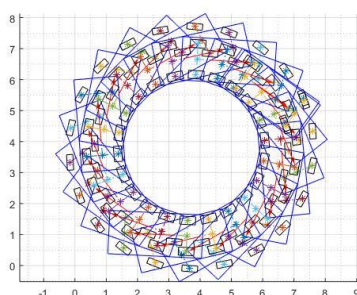
$$x_2 = Y :$$



$$x_3 = \theta :$$



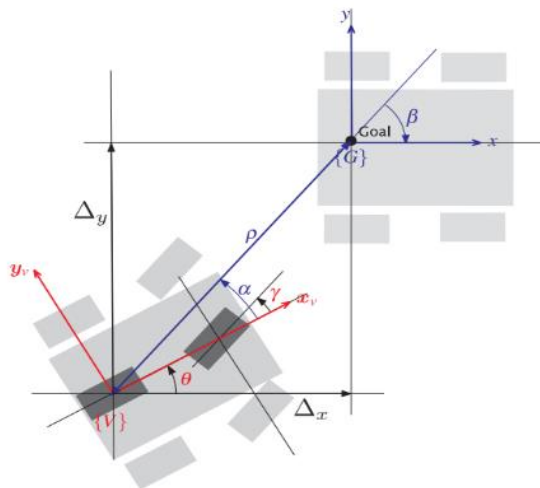
Αυτές οι συναρτήσεις είναι περιοδικές και ορίζουν έναν κύκλο. Παραπάνω φαίνεται η πρώτη περίοδος. Η γωνία  $\theta$  καταλήγει στις  $-360^\circ$  και από εκεί και πέρα είναι η ίδια με την αρχική αν κάνουμε την απλοποίηση της γωνίας σε έναν κύκλο ( $-360 < \theta < 360$ ). Κάποια στιγμιότυπα από την θέση του αμαξιού σε αυτή την πορεία φαίνεται παρακάτω :



Παρατηρούμε ότι το όχημα διαγράφει κυκλική τροχιά ες αεί. Αν η γωνία  $\gamma = 0^\circ$  οι πρώτες δύο μεταβλητές κατάστασης τείνουν στο  $\pm\infty$  αναλόγως τις αρχικές τιμές και η τρίτη μεταβλητή κατάστασης παραμένει σταθερή.

Στις παραπάνω καταστατικές εξισώσεις εφαρμόζουμε τις εξής αλλαγές μεταβλητής (πολικές συντεταγμένες) :

- I.  $\rho = \Delta x^2 + \Delta y^2$
- II.  $a = \tan^{-1}\left(\frac{\Delta y}{\Delta x}\right) - \theta$
- III.  $\beta = -\theta - \alpha$



Οι παραπάνω αλλαγές μεταβλητής ορίζουν ένα νέο σύστημα με κατάσταση  $W = [\rho \ a \ \beta]^T$  το οποίο έχει τις εξής καταστατικές εξισώσεις :

$$\dot{W} = \tilde{A} * W + \tilde{B} * U, \text{ όπου } \tilde{A} = A \text{ ίσος με τον μηδενικό πίνακα } 0_{2 \times 2} \text{ και } \tilde{B} = \begin{bmatrix} -\cos a & 0 \\ \frac{\sin a}{\rho} & -1 \\ -\frac{\sin a}{\rho} & 0 \end{bmatrix} \text{ και } U = \begin{bmatrix} |u| \\ \gamma \end{bmatrix}, \text{ όταν το } a \in (-180^\circ, 180^\circ), \text{ δηλαδή ο στόχος}$$

βρίσκεται μπροστά από το όχημα. Για αυτό τον λόγο χρειαζόμαστε μικρή περίοδο δειγματοληψίας που αναφέρθηκε σε προηγούμενη παράγραφο μιας και μία ‘απρόβλεπτη’ μετακίνηση του οχήματος μπορεί να το αναγκάσει να βγει εκτός ορίων του  $\alpha$  και να αλλάξει ο πίνακας  $\tilde{B}$  (κάποια πρόσημα αλλάζουν).

Στις παραπάνω καταστατικές εξισώσεις εφαρμόζουμε τον εξής νόμο ελέγχου ανατροφοδότησης κατάστασης :

- I.  $|u| = k_\rho * \rho$
- II.  $\gamma = k_\alpha * \alpha + k_\beta * \beta$

(Το μυστικό του ελέγχου της πρότασης II. είναι ότι ο όρος  $k_\alpha * \alpha$  τείνει να ευθυγραμμίσει το όχημα προς τον στόχο ενώ ο όρος  $k_\beta * \beta$  τείνει να το περιστρέψει έτσι ώστε να παραλληλιστεί με τον στόχο.)

Ο παραπάνω εξισώσεις μας οδηγούν στο εξής σύστημα κλειστού βρόχου :

$$\dot{W} = \begin{bmatrix} -k_\rho * \cos a \\ k_\rho * \sin a - k_\alpha * a - k_\beta * \beta \\ -k_\rho * \sin a \end{bmatrix} * W$$

Οι παραπάνω καταστατικές εξισώσεις είναι ευσταθής και τείνουν ασυμπτωτικά στο  $W_e = [0 \ 0 \ 0]^T$  αν η  $\dot{W}$  είναι αρνητικά ορισμένη. Παρατηρούμε ότι για να ισχύει το παραπάνω πρέπει κάθε γραμμή της παραπάνω μήτρας να είναι αρνητικά



ορισμένη (για  $\alpha \in (-90^\circ, 90^\circ)$ ). Συμπεραίνουμε λοιπόν για τα κέρδη  $k_\rho, k_\alpha, k_\beta$  ότι πρέπει να ισχύουν οι εξής συνθήκες :

- I.  $k_\rho > 0$
- II.  $k_\beta < 0$
- III.  $k_\alpha - k_\rho \geq 0$

Για να οδηγήσουμε το όχημα σε άλλη κατάσταση ισορροπίας αρκεί πλέον να κάνουμε μία τελική αλλαγή μεταβλητής στο σύστημα κλειστού βρόχου έτσι ώστε να κάνουμε το διάνυσμα  $W_e = [X_f \ Y_f \ \theta_f]^t$  κατάσταση ισορροπίας. Αντικαθιστούμε λοιπόν το διάνυσμα  $W = W - W_e$  και έτσι το σύστημα έχει τις ίδιες ιδιότητες όπως πριν αλλά συγκλίνει στην επιθυμητή κατάσταση. Παρακάτω παρατίθενται δύο παραδείγματα του αλγορίθμου μιας και ενώ διακριτοποιείται οι ιδιότητες ως προς την ευστάθεια παραμένουν. Παρακάτω παρατίθεται ένα παράδειγμα του νόμου ελέγχου που αναλύθηκε παραπάνω :

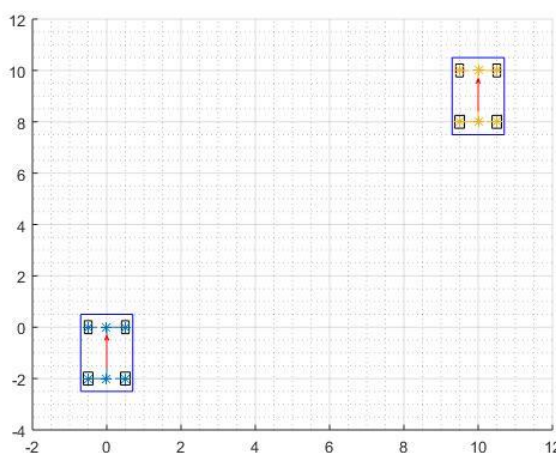
Αρχική κατάσταση :

$$W_0 = [0 \ 0 \ 90]^t$$

Τελική κατάσταση :

$$W_f = [10 \ 10 \ 90]^t$$

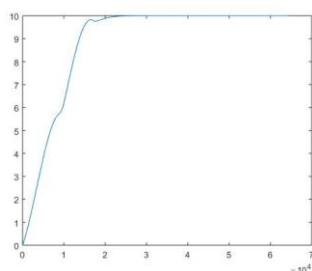
(Η αρχική και τελική κατάσταση οπτικοποιούνται στο διάγραμμα δίπλα)



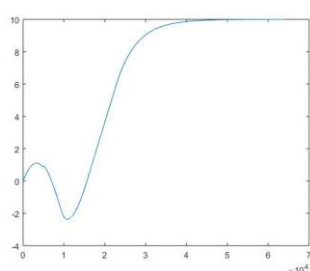
Οι τιμές που επιλέχθηκαν τελικά είναι :  $k_\rho = 2, k_\beta = -13, k_\alpha = 2$

Τρέχοντας τον παραπάνω αλγόριθμο στο MATLAB καταλήγουμε στα παρακάτω αποτελέσματα για την κατάσταση του αυτοκινήτου και τον νόμο ελέγχου :

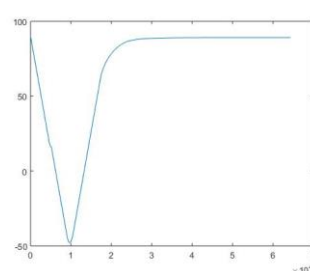
$x_1 = X :$



$x_2 = Y :$



$x_3 = \theta :$

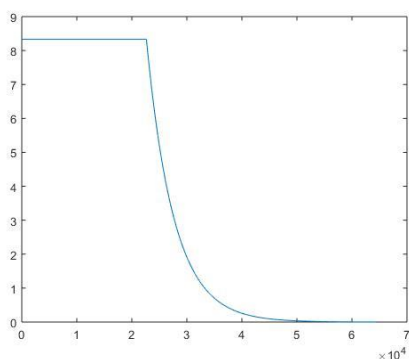




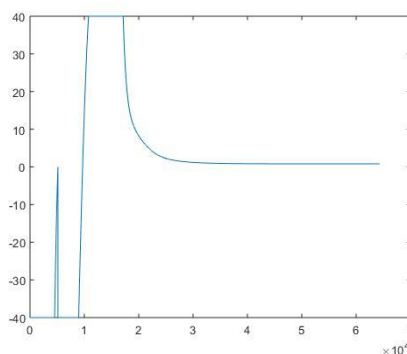
Οι γραφικές παραστάσεις έγιναν για  $\kappa = 1, 2, 3 \dots$  και για τον πραγματικό χρόνο αρκεί να πολλαπλασιάσουμε το  $\kappa$  με την περίοδο δειγματοληψίας. Παρατηρούμε την ασυμπτωτική σύγκλιση του συστήματος στις επιθυμητές τιμές.

Ο νόμος ελέγχου γεννά τα εξής σήματα ελέγχου για το σύστημα :

$|\vec{u}| :$



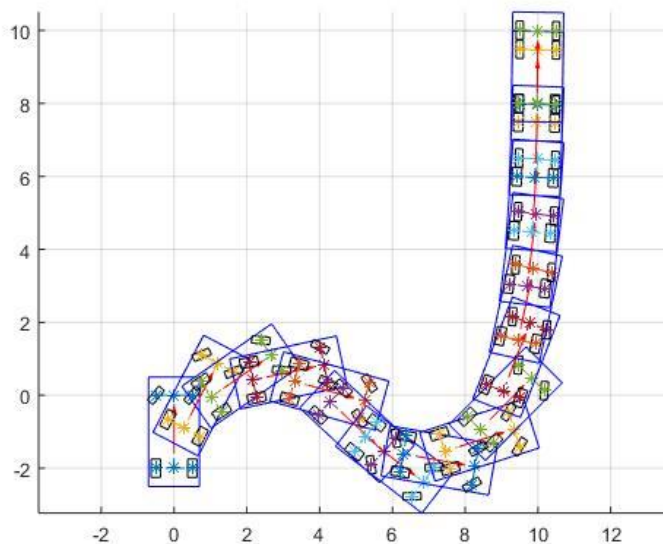
$\gamma :$



Παρατηρούμε τις μη γραμμικότητες του συστήματος για το όριο της ταχύτητας (μετρείται σε m/s στο παραπάνω παράδειγμα) και το όριο της γωνίας των μπροστινών τροχών (μετρείται σε μοίρες).

Τελικά, στιγμιότυπα της πορείας του οχήματος ανά 1.5 μέτρα φαίνονται παρακάτω :

Παρατηρούμε την παράξενη πορεία που γεννά ο νόμος ελέγχου έτσι ώστε να συγκλίνει το όχημα στην επιθυμητή πόζα (ενώ θα μπορούσε να κάνει μία ευθεία). Επίσης, ο συνολικός χρόνος για να διανύσει αυτή την απόσταση το όχημα είναι 6.432 δευτερόλεπτα.

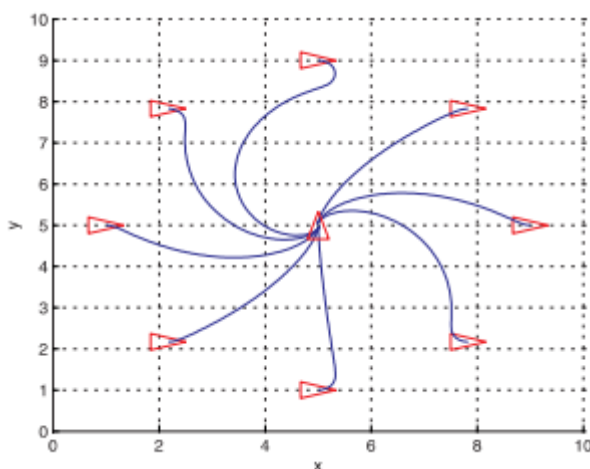


Διαλέγουμε συνθήκες τερματισμού για τον αλγόριθμο, σφάλμα στην θέση του οχήματος  $< 0.001$  m και σφάλμα στην γωνία  $< 1^\circ$  μιας και αφού το σύστημα συγκλίνει ασυμπτωτικά χρειάζεται (θεωρητικά) άπειρο χρόνο.

Για την περίπτωση που το όχημα χρειάζεται να προσεγγίσει ένα αντικείμενο που βρίσκεται πίσω του ( $\alpha \in (-90^\circ, -270^\circ)$ ) αρκεί να ορίσουμε όπου  $|u| = -|u|$  και το όχημα προσεγγίζει τον στόχο αυτή την φορά κινούμενο προς τα πίσω.



Συνοπτικά παραδείγματα της παραπάνω πρότασης φαίνονται παρακάτω όπου τα διαγράμματα είναι πιο συνεκτικά :



$$\triangleright W_f = [5 \ 5 \ 90]^t$$

Παρατηρούμε πως όταν ο στόχος βρίσκεται 'πίσω' από το όχημα, αυτό κινείται με αρνητική ταχύτητα.

Εν κατακλείδι, βρήκαμε έναν νόμο ελέγχου για ένα καθαρά μη γραμμικό σύστημα το οποίο οδηγεί το όχημα σε επιθυμητή κατάσταση εφαρμόζοντας κάθε φορά τον νόμο που αναφέρθηκε στην σελίδα 10.

## 7) Προσομοίωση ολικού συστήματος

Η απόκριση του ολικού συστήματος όπου η ταχύτητα  $|\vec{u}|$  του νόμου ελέγχου της παραγράφου 6, υπολογίζεται μέσω του συστήματος κλειστού βρόχου (DC Motor – PID) που αναλύεται στην παράγραφο 5 εξετάζεται παρακάτω.

Για να υλοποιηθεί το η προσομοίωση του ολικού συστήματος πρώτα πρέπει να μετατρέψουμε την συνάρτηση μεταφοράς  $G(z)$  της παραγράφου 5 σε εξισώσεις διαφορών έτσι ώστε να μπορούμε την χρησιμοποιήσουμε στο πεδίο του χρόνου κάθε  $T_s$  δευτερόλεπτα. Οι πράξεις φαίνονται παρακάτω :

$$G_{ClosedLoop}(z) = \frac{0.04001*z^3 - 0.03997*z^2 - 0.03998*z + 0.03994}{z^4 - 1.934*z^3 - 0.06395*z^2 + 1.934*z - 0.936}$$

Αρχικά, διαιρούμε με  $z^{-4}$  :

$$\frac{Y(z)}{U(z)} = \frac{0.04001*z^{-1} - 0.03997*z^{-2} - 0.03998*z^{-3} + 0.03994*z^{-4}}{1 - 1.934*z^{-1} - 0.06395*z^{-2} + 1.934*z^{-3} - 0.936*z^{-4}}$$

Ύστερα, πολλαπλασιάζουμε με τους παρονομαστές της παραπάνω εξίσωσης :

$$Y(z) * (1 - 1.959 * z^{-1} - 0.04117 * z^{-2} + 1.959 * z^{-3} - 0.9589 * z^{-4}) = U(z) * (0.04001 * z^{-1} - 0.03997 * z^{-2} - 0.03998 * z^{-3} + 0.03994 * z^{-4})$$

Τέλος εφαρμόζουμε αντίστροφο μετασχηματισμό  $Z$  ( $Z^{-1}$ ) και καταλήγουμε στις παρακάτω εξισώσεις διαφορών (αναλύονται γιατί αυτές χρησιμοποιούνται στο main script του MATLAB) :

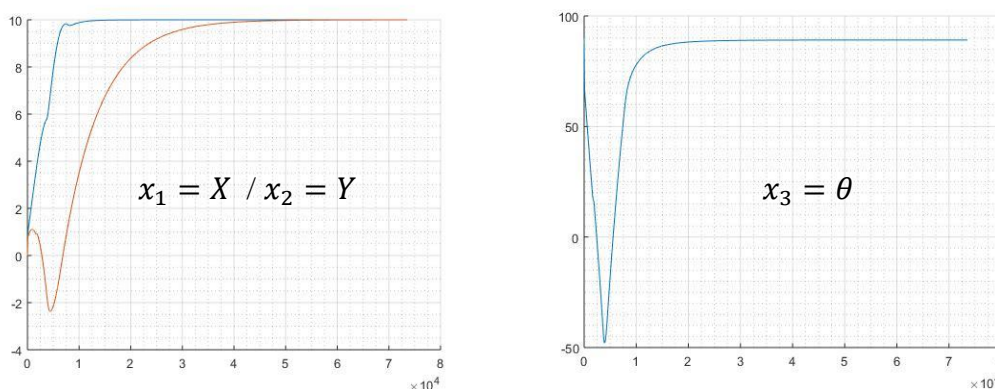




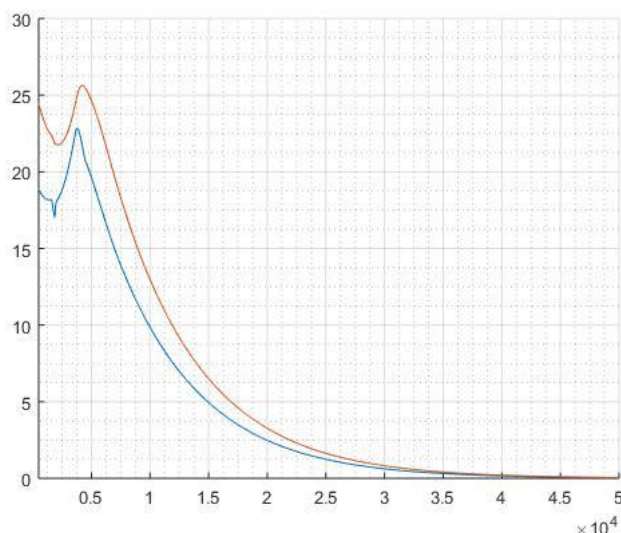
$$Y(k) = 0.04001 * U(k - 1) - 0.03997 * U(k - 2) - 0.03998 * U(k - 3) + 0.03994 * U(k - 4) + 1.934 * Y(k - 1) + 0.06395 * Y(k - 2) - 1.934 * Y(k - 3) + 0.936 * Y(k - 4)$$

Θεωρούμε το σύστημα σε χαλάρωση για  $k < 0$  δηλαδή  $Y(k) = 0$ ,  $k < 0$  και θεωρούμε ότι το σύστημα διεγείρεται για  $k = 0$ . (μπορούμε να θεωρήσουμε ότι για οποιοδήποτε  $k$  και να διεγείρουμε το σύστημα η απόκριση θα είναι ίδια μιας και το μοντέλο του DC Motor είναι χρονικώς αμεταβλητό). Πολλαπλασιάζουμε επίσης την έξοδο του μοντέλου του DC Motor με μία σταθερά η οποία είναι ουσιαστικά η μετάδοση σε ένα όχημα, έτσι ώστε η ταχύτητα να είναι ανάλογη των στροφών του κινητήρα (αυτό μας υποδεικνύει η συνάρτηση μεταφοράς).

Για δοκιμή των παραπάνω, δοκιμάζουμε ξανά το παράδειγμα που κάναμε στην παράγραφο 6 και τα αποτελέσματα φαίνονται παρακάτω στην επόμενη σελίδα :



Αφού είδαμε την τροχιά που ακολουθεί κάθε μεταβλητή κατάστασης, παρακάτω φαίνεται η έξοδος του υπό έλεγχο DC Motor σε σχέση με τον νόμο ελέγχου που γεννά ο αλγόριθμος παρακολούθησης :



Παρατηρούμε ότι η  $r(t) = \overline{u(t)}$  που γεννά ο αλγόριθμος παρακολούθησης ακολουθείται ικανοποιητικά από την έξοδο του DC Motor.

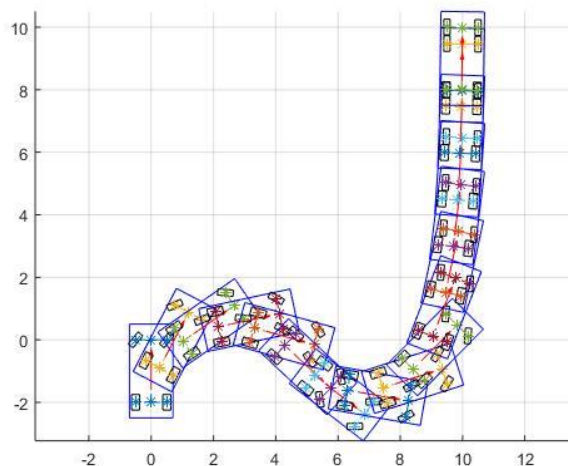
Τελικά στιγμιότυπα της ολικής πορείας που διαγράφει το όχημα με την συνεργασία των δύο ελεγκτών (ελεγκτής κατάστασης – αλγόριθμος παρακολούθησης και PID) φαίνονται παρακάτω :

Αρχική κατάσταση :

$$W_0 = [0 \quad 0 \quad 90]^t$$

Τελική κατάσταση :

$$W_f = [10 \quad 10 \quad 90]^t$$



Ο συνολικός χρόνος που χρειάζεται για να προσεγγίσει τον στόχο του το όχημα είναι λίγο μεγαλύτερος λόγω του μικρότερου μέτρου ταχύτητας που φτάνει ο κινητήρας σε σχέση με την ουτοπική υλοποίηση στην παράγραφο 6. Ο νόμος ελέγχου γεννά την επιθυμητή τροχιά που πρέπει να διανύσει το όχημα (ως συνάρτηση της θέσης) άρα είναι λογικό αφού η απόκριση του κινητήρα καθυστερεί να καθυστερεί και το όχημα.

## 8) Συμπεράσματα

Εν κατακλείδι λοιπόν μπορούμε να υποθέσουμε ότι υλοποιήθηκε ένα πλήρες σύστημα ψηφιακού ελέγχου για μετακίνηση ενός οχήματος, που περιγράφεται από τις εξισώσεις της παραγράφου 2, από μία αρχική κατάσταση  $[X_0 \quad Y_0 \quad \theta_0]$  σε μία επιθυμητή κατάσταση ισορροπίας  $[X_f \quad Y_f \quad \theta_f]$  με προϋπόθεση αυτή η κατάσταση να είναι ασυμπτωτικά ευσταθής. Πιο αναλυτικά μελετήθηκαν θέματα αναζήτησης της περιόδου δειγματοληψίας, διακριτού ελεγκτή PID, ελεγκτή κατάστασης μη γραμμικών καταστατικών εξισώσεων και διασύνδεση των παραπάνω υποσυστημάτων. Χρησιμοποιήθηκαν μέθοδοι σχεδιασμού (εξαντλητική αναζήτηση) ελεγκτή με ανατροφοδότηση εξόδου στο πεδίο του Laplace και διακριτοποίηση αυτού στο μιγαδικό πεδίο Z, μέθοδοι σχεδιασμού ελεγκτή με ανατροφοδότηση κατάστασης (για τον οποίο ανατρέξαμε σε βιβλιογραφία που αναφέρεται στο τέλος της αναφοράς) και μέθοδοι απεικόνισης των ιδιοτήτων τόσο στο πεδίο του χρόνου όσο και στο μιγαδικό πεδίο Z (βηματική απόκριση, σφάλμα μόνιμης κατάστασης, γεωμετρικός τόπος ριζών πολωνύμου).



## 9) Παράρτημα Α

### Ανάλυση κώδικα MATLAB

Για όλες τις παραπάνω παραγράφους που αναλύθηκαν υλοποιήθηκαν 4 scripts και δύο συναρτήσεις στο MATLAB. Αρχικά, για το DC Motor υλοποιήθηκε ένα script όπου μπορούμε να προσθέσουμε εμείς μία συνάρτηση μεταφοράς και να διαλέξουμε τα κέρδη του PID ελεγκτή. Διάφορα plots γίνονται κατά την προσπέλαση του κώδικα. Το συνολικό script φαίνεται παρακάτω :

1 -	<code>clc;clear all;close all force;</code>	[1-3] αρχικοποίηση workspace και καθαρισμός παλαιότερων μεταβλητών
2 -	<code>%hold on;</code>	
3 -	<code>grid on; grid minor;</code>	
4 -	<code>num = 0.5;</code>	
5 -	<code>den = [0.025 0.3 0.5];</code>	[4-6] ορισμός συνάρτησης μεταφοράς του DC Motor
6 -	<code>motor_tf = tf(num,den);</code>	
7 -	<code>%step(motor_tf);</code>	[7] Βηματική απόκριση
8 -	<code>Ts = 0.0001;</code>	
9 -	<code>motord = c2d(motor_tf,Ts,'zoh');</code>	[8-9] Χρόνος δειγματοληψίας και διακριτοποίηση συνάρτησης μεταφοράς
10 -	<code>s = tf('s');</code>	
11 -	<code>Kp = 300; Ki = 400; Kd = 20;</code>	
12 -	<code>pid = Kp + Ki/s + Kd*s;</code>	[10-13] ορισμός και διακριτοποίηση PID ελεγκτή
13 -	<code>pidd = c2d(pid,Ts,'tustin');</code>	
14 -	<code>closedloop = feedback(pidd*motord,1);</code>	[14-19] ορισμός κλειστού συστήματος, γ.τ.ρ συνάρτησης μεταφοράς ανοιχτού βρόχου και βηματική απόκριση ανοιχτού και κλειστού βρόχου.
15 -	<code>hold on; grid on; grid minor;</code>	
16 -	<code>%rlocus(motord)</code>	
17 -	<code>step(closedloop);</code>	
18 -	<code>step(motord);</code>	
19 -	<code>grid on; grid minor;</code>	

Ύστερα, για την ανάγκη διάφορων άλλων scripts ορίστηκε μία συνάρτηση όπου απλοποιεί διάφορες γωνίες (σε μοίρες) σε όρια  $(-360,360)$ . Η συνάρτηση φαίνεται παρακάτω :

1 -	<code>function [th] = twoRad(th1)</code>	
2 -	<code>while (abs(th1)&gt;=360)</code>	
3 -	<code>if th1&gt;=360</code>	
4 -	<code>th1 = th1 - 360;</code>	
5 -	<code>elseif th1&lt;=-360</code>	
6 -	<code>th1 = th1 + 360;</code>	
7 -	<code>end</code>	
8 -	<code>end</code>	
9 -	<code>th = th1;</code>	
10 -	<code>end</code>	

Η συνάρτηση αυτή αφαιρεί η προσθέτει 360 μοίρες όσο η μεταβλητή th1 βρίσκεται εκτός των ορίων που προαναφέρθηκαν και την επιστρέφει μόλις βρεθεί εντός αυτών.

Επόμενη συνάρτηση που υλοποιήθηκε ήταν η συνάρτηση απεικόνισης του οχήματος με την οποία έγιναν όλα τα plots της αναφοράς που περιέχουν ένα όχημα στις παραπάνω παραγράφους. Η συνάρτηση αυτή παίρνει ως εισόδους τις μεταβλητές  $[x \ y \ \theta \ \gamma]$  και σφειδιάζει το όχημα. Ο κώδικας φαίνεται στην επόμενη σελίδα :



Επειδή η συνάρτηση είναι πολλές γραμμές παρατίθενται οι πρώτες λίγες και οι τελευταίες μιας και ο κώδικας είναι επαναλαμβανόμενος.

```

1 function [chassis,flwheel,frwheel,rlwheel,rrwheel,Laxis] = myCar(x,y,theta1,thetawheels1)
2 - theta = twoRad(theta1);thetawheels = twoRad(thetawheels1);
3 % chassis plot
4 - chassisx = [x-(0.5*cosd(180-theta)+0.7*cosd(theta-90)),...
5 - x-(0.5*cosd(180-theta)-0.7*cosd(theta-90)),...
6 - x+(0.5*cosd(180-theta)+0.7*cosd(theta-90))-2*cosd(-theta), ...
7 - x+(0.5*cosd(180-theta)-0.7*cosd(theta-90))-2*cosd(-theta)];
8 - chassisy = [y+(0.7*cosd(theta)+0.5*sind(theta)),...
9 - y+(-0.7*cosd(theta)+0.5*sind(theta)),...
10 - y+(-0.7*cosd(theta)+0.5*sind(theta))-3*sind(theta),...
11 - y+(0.7*cosd(theta)+0.5*sind(theta))-3*sind(theta)];
12 - chassis = [chassisx,chassisx(1);chassisy,chassisy(1)];
13 % front wheels plot
14 - thetan = theta+thetawheels;
15 - flwheelxc = x-0.5*cosd(theta-90);frwheelxc = x+0.5*cosd(theta-90);
16 - flwheelyc = y+0.5*cosd(theta);frwheelyc = y-0.5*cosd(theta);
17 - flwheelx = [flwheelxc+0.25*cosd(thetan)-0.1*sind(thetan),...

```

Η λογική της συνάρτησης είναι να υπολογισθούν οι ακμές του σχήματος με βάση τις διαστάσεις του οχήματος και ύστερα να τις ενώσει και να δημιουργήσει και ένα διάνυσμα κατά την διεύθυνση που κοιτάει το όχημα. Οι τελευταίες γραμμές της συνάρτησης αυτής όπου και γίνεται απεικόνιση των παραπάνω φαίνονται παρακάτω :

```

64 %% plots
65 - hold on; grid on; grid minor;
66 - plot(chassis(1,:),chassis(2:),'b');
67 - plot(flwheel(1,:),flwheel(2:),'k');
68 - plot(frwheel(1,:),frwheel(2:),'k');
69 - plot(rlwheel(1,:),rlwheel(2:),'k');
70 - plot(rrwheel(1,:),rrwheel(2:),'k');
71 - plot(Laxis(1,:),Laxis(2:),'-.*');
72 - p1 = [(rlwheelxc+rrwheelxc)/2+0.4*cosd(theta) ...
73 - (rlwheelyc+rrwheelyc)/2+0.4*sind(theta)];
74 - p2 = [(flwheelxc+frwheelxc)/2-0.3*cosd(theta) ...
75 - (flwheelyc+frwheelyc)/2-0.3*sind(theta)];
76 - dp = p2-p1;
77 - quiver(p1(1),p1(2),dp(1),dp(2),'LineWidth',0.75,...
78 - 'MaxHeadSize',0.5,'AutoScale','off','color','r');
79 - end

```

[65-71] plot των σημείων που υπολογίστηκαν παραπάνω

[72-78] σχεδιασμός διανύσματος κατά την διεύθυνση κίνησης του οχήματος.

Ύστερα χρειάστηκε να οριστεί μία συνάρτηση έτσι ώστε να υπολογίζει την πορεία που διαγράφει το όχημα σύμφωνα με τον νόμο ελέγχου της παραγράφου 6 μεταξύ δύο διαδοχικών στιγμών δειγματοληψίας μιας και δεν υπάρχει κλειστός μαθηματικός τύπος να λυθούν οι μη γραμμικές εξισώσεις στο πεδίο Z. Η συνάρτηση αυτή φαίνεται παρακάτω :

```

1 function [newCoords] = arcCompute(Coords,Length)
2 - backWheels = [Coords(1)-2*cosd(Coords(3));Coords(2)-2*sind(Coords(3))];
3 - if Coords(4)>0
4 - L = 2/tand(Coords(4));
5 - irc = [backWheels(1)+L*cosd(Coords(3)+90);...
6 - backWheels(2)+L*sind(Coords(3)+90)];
7 - %scatter(irc(1),irc(2),'b');
8 - R = norm([Coords(1)-irc(1);Coords(2)-irc(2)]);
9 - theta = Length*360/(2*pi*R);
10 - newX = irc(1)-R*sind(-Coords(3)-Coords(4)-theta);
11 - newY = irc(2)-R*cosd(-Coords(3)-Coords(4)-theta);
12 - newCoords = [newX;newY;Coords(3)+theta];

```



```

13 - elseif Coords(4)<0
14 -     L = 2/tand(Coords(4));
15 -     irc = [backWheels(1)+L*cosd(Coords(3)+90);...
16 -           backWheels(2)+L*sind(Coords(3)+90)];
17 -     %scatter(irc(1),irc(2),'b');
18 -     R = norm([Coords(1)-irc(1);Coords(2)-irc(2)]);
19 -     theta = Length*360/(2*pi*R);
20 -     newX = irc(1)-R*sind(Coords(3)+Coords(4)-theta);
21 -     newY = irc(2)+R*cosd(Coords(3)+Coords(4)-theta);
22 -     newCoords = [newX;newY;Coords(3)-theta];
23 - elseif Coords(4)==0
24 -     newCoords = [Coords(1)+Length*cosd(Coords(3));...
25 -                 Coords(1)+Length*sind(Coords(3));Coords(3)];
26 - end

```

[4-9, 14-29] υπολογισμός κέντρου καμπυλότητας και ακτίνας

[9-12, 20-22, 24-25] υπολογισμός εξόδου

Η συνάρτηση αυτή επιστρέφει τις νέες συντεταγμένες του οχήματος σε σχέση με την κατάσταση και τον νόμο ελέγχου

Τέλος, τα δύο τελευταία scripts είναι παρόμοια. Αφορούν τον αλγόριθμο παρακολούθησης. Το ένα είναι χωρίς το μοντέλο του συστήματος DC Motor – PID και χρησιμοποιήθηκε για την παράγραφο 6 ενώ το άλλο αφορά την πλήρη προσομοίωση του ολικού συστήματος που σχεδιάστηκε. Παρατίθεται μόνο το τελευταίο ενώ θα γίνει αναφορά για τις αλλαγές που έγιναν σε σχέση με το αρχικό (παράγραφος 6).

```

1 - clc; clear all; close all force;
2 - Tsampling = 0.001;
3 - %% inputs
4 - prompt = {'Enter x desired coordinate:',...
5 -           'Enter y desired coordinate:',...
6 -           'Enter desired angle:',...
7 -           'Enter x initial coordinate:',...
8 -           'Enter y initial coordinate:', 'Enter initial angle:'};
9 - dlg_title = 'Algorithm Inputs';
10 - num_lines = 1;
11 - defaultans = {'10','10','45','0','0','90'};
12 - answer = str2double(inputdlg(prompt,dlg_title,...
13 -                               num_lines,defaultans));
14 - initCoords = answer(4:6);
15 - desiredCoords = answer(1:3);

```

Αρχικά, ορίζεται η περίοδος δειγματοληψίας και ένα παράθυρο για επικοινωνία με τον χρήστη για είσοδο αρχικής και τελικής κατάστασης.

Δίπλα φαίνεται το παράθυρο επικοινωνίας με τον χρήστη για να ορίσει την αρχική και τελική κατάσταση :

Στις παρακάτω γραμμές φαίνονται οι αρχικοποιήσεις για τον αλγόριθμο παρακολούθησης ( $K_a$ ,  $K_b$ ,  $K_p$  [17]) καθώς και οι συντελεστές των εξισώσεων διαφορών της παραγράφου 7 (WeightsU, WeightsY [22-23]) :

```

16 - %% plane position inspection algorithm
17 - ka = 2 ; kb = -13 ; kr = 2;
18 - l2plot = 3.1;
19 - instantCoords = initCoords(1:3);
20 - x1 = []; x2 = []; x3 = []; vp = []; gp = []; vp1 = [];
21 - Uprev = [0 0 0 0]; Yprev = [0 0 0 0];
22 - WeightsU = [0.04001 -0.03997 -0.03998 0.03994];
23 - WeightsY = [1.9346 0.06395822 -1.9346 0.93604178];

```



24 -	<code>while sum(abs(instantCoords(1:2)-desiredCoords(2:2)))&gt;0.001</code>	[24] έλεγχος αν έφτασε το όχημα.
25 -	<code>dx = -desiredCoords(1)+instantCoords(1);</code>	
26 -	<code>dy = -desiredCoords(2)+instantCoords(2);</code>	
27 -	<code>r = sqrt(dx^2+dy^2);</code>	
28 -	<code>a = twoRad(atan2(dy/dx)-twoRad(instantCoords(3)));</code>	[25-31] υπολογισμός σημμάτων ελέγχου
29 -	<code>b = twoRad(-instantCoords(3)-a+desiredCoords(3));</code>	
30 -	<code>v1 = kr*r;</code>	
31 -	<code>g = twoRad(ka*a+kb*b);</code>	
32 -	<code>v = Uprev*WeightsU'+Yprev*WeightsY';</code>	[32] υπολογισμός εξόδου DC Motor
33 -	<code>Uprev(2:4) = Uprev(1:3);</code>	
34 -	<code>Uprev(1) = v1;</code>	
35 -	<code>Yprev(2:4) = Yprev(1:3);</code>	[33] ενημέρωση προηγούμενων τιμών εξισώσεων διαφορών
36 -	<code>Yprev(1) = v;</code>	
37 -	<code>if g&gt;40</code>	
38 -	<code>g = 40;</code>	
39 -	<code>elseif g &lt; -40</code>	
40 -	<code>g = -40;</code>	[37-38] όρια και μη γραμμικότητες του μοντέλου
41 -	<code>end</code>	
42 -	<code>if v &gt; 30/3.6</code>	
43 -	<code>v = 30/3.6;</code>	
44 -	<code>end</code>	
45 -	<code>if l2plot &gt; 1.5</code>	[45-48] plot του οχήματος
46 -	<code>myCar(instantCoords(1),instantCoords(2),instantCoords(3),g);</code>	
47 -	<code>l2plot = 0;</code>	
48 -	<code>end</code>	
49 -	<code>instantCoords = arcCompute([instantCoords(1);instantCoords(2);...</code>	
50 -	<code>instantCoords(3);g],22*v*Tsampling)</code>	
51 -	<code>instantCoords(3) = twoRad(instantCoords(3));</code>	[49-51] υπολογισμός νέας θέσης του οχήματος μετά από Ts
52 -	<code>l2plot = l2plot + abs(22*v*Tsampling);</code>	
53 -	<code>x1 = [x1 instantCoords(1)];</code>	
54 -	<code>x2 = [x2 instantCoords(2)];</code>	
55 -	<code>x3 = [x3 instantCoords(3)];</code>	
56 -	<code>vp = [vp v];</code>	[53-58] αποθήκευση κατάστασης και σημμάτων ελέγχου σε κάθε βήμα του αλγόριθμου.
57 -	<code>vp1 = [vp1 v1];</code>	
58 -	<code>gp = [gp g];</code>	
59 -	<code>end</code>	

βήμα του αλγόριθμου.

Τα διαγράμματα της παραγράφου 7 έγιναν με της μεταβλητές των γραμμών [53-58] ενώ για την παράγραφο 6 χρησιμοποιήθηκε ο ίδιος κώδικας με μόνη διαφορά ότι δεν χρησιμοποιήθηκαν οι γραμμές [32-33], και η μεταβλητή v1 αντικαταστάθηκε από την v.

## 10) Παράρτημα Β

### Θεωρητικό υπόβαθρο

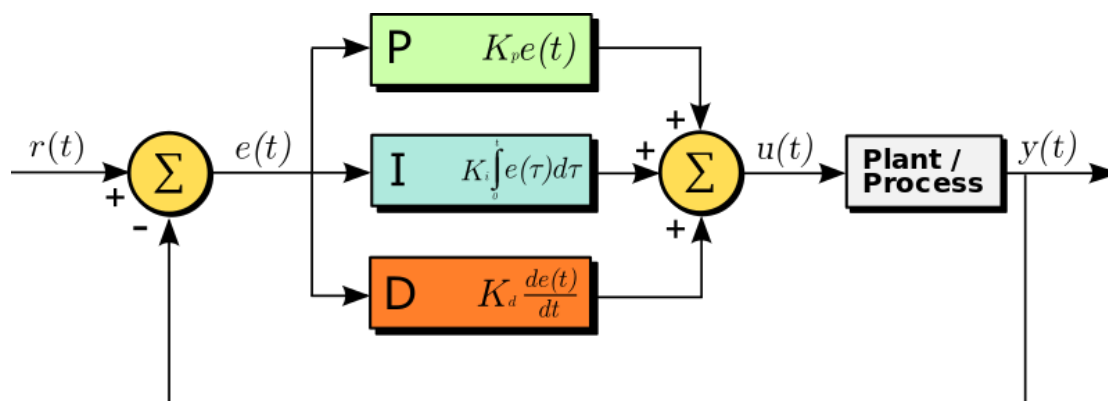
Μέθοδος Zero Order Hold : Η μέθοδος Zero Order Hold μετασχηματίζει ένα σήμα συνεχούς χρόνου σε διακριτό κρατώντας σταθερή την τιμή του μεταξύ δύο στιγμιότυπων δειγματοληψίας. Για διακριτοποίηση συστημάτων με Zero Order Hold πολλαπλασιάζουμε πρώτα την συνάρτηση μεταφοράς με  $H_{zoh}(s) = \frac{1-e^{-sT}}{s}$  και ύστερα διακριτοποιούμε.

Μέθοδος Tustin : Η μέθοδος Tustin ή μετασχηματισμός Mobius ουσιαστικά προσεγγίζει έναν First Order Hold χρησιμοποιώντας όμως νούμερα με διαφορά  $\pm$  μία ημιπερίοδο από την στιγμή κ. Η ένα προς ένα απεικόνιση από το πεδίο Laplace στο μιγαδικό επίπεδο Z γίνεται με την εξής αλλαγή μεταβλητής :  $z \approx \frac{1+s\frac{T}{2}}{1-s\frac{T}{2}}$ .

Ελεγχιμότητα : Η ελεγχιμότητα περιγράφει την ικανότητα ενός εξωτερικού σήματος να διεγείρει ένα σύστημα και να το αναγκάσει να μεταβεί από μία αρχική κατάσταση σε μία τελική σε πεπερασμένο χρόνο.

Παρατηρησιμότητα : Ένα σύστημα που περιγράφεται από καταστατικές εξισώσεις λέγεται παρατηρήσιμο αν δεδομένης αρχικής κατάστασης και εξόδου του συστήματος μπορούμε να υπολογίσουμε την κατάσταση σε πεπερασμένο χρόνο.

PID tuner : Ο ρυθμιστής PID ελέγχει ένα σύστημα χρησιμοποιώντας ως συνάρτηση εισόδου (κατά κανόνα) την διαφορά της εξόδου από μία επιθυμητή τιμή αυτής πολλαπλασιάζοντας, ολοκληρώνοντας και διαφορίζοντας αυτή την διαφορά όπως φαίνεται παρακάτω :



Γεωμετρικός τόπος ριζών : Η μέθοδος του τόπου των ριζών είναι μία γραφική απεικόνιση των ριζών μίας εξίσωσης σε σχέση με κάποιο κριτήριο σχεδιασμού, συνήθως το κέρδος κάποιου ελεγκτή.

Ασυμπτωτική ευστάθεια : Η ασυμπτωτική ευστάθεια ή ευστάθεια κατά Lyapunov ορίζεται ως η ιδιότητα της κατάστασης ενός συστήματος όταν βρεθεί κοντά σε μία κατάσταση  $x_e$  να συγκλίνει ασυμπτωτικώς σε αυτήν.

Σφάλμα μόνιμης κατάστασης : Η διαφορά της επιθυμητής εξόδου από την πραγματική όταν το σύστημα βρίσκεται στην μόνιμη κατάσταση.

Υπερύψωση : Όταν κατά την μεταβατική κατάσταση το σύστημα τείνει να προσπεράσει την επιθυμητή έξοδο και ύστερα συγκλίνει σε αυτήν. Αυτό γίνεται όταν το σύστημα έχει μιγαδικούς πόλους.



## 11) Βιβλιογραφία

- *Digital Control Engineering Analysis and Design (Second Edition) / M. Sami Fadali, Antonio Visioli*
- *Ψηφιακός Έλεγχος Συστημάτων / Γεώργιος Π. Σύρκος*
- *Robotics, Vision and Control Fundamental algorithms in MATLAB / Peter Corke*
- *Discrete-Time PID Controller Tuning Using Frequency Loop-Shaping / Konstantinos Tsakalis (abstract)*
- *Ηλεκτρικές Μηχανές / Stephen J. Chapman*
- [www.mathworks.com](http://www.mathworks.com)
- [www.wikipedia.com](http://www.wikipedia.com)
- *Πανεπιστημιακές σημειώσεις στο μάθημα Ψηφιακός Έλεγχος [EE883] / Πέτρος Γρουμπός*