

PROJECT ΔΙΚΤΥΩΜΕΝΑ ΡΟΜΠΟΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ : Λάμπης Παπακώστας

ΑΜ : 228467

ΕΤΟΣ : 5ο

ΤΟΜΕΑΣ : ΣΑΕ

ΑΚΑΔΗΜΑΙΚΟ ΕΤΟΣ: 2017 – 2018

“Distributed Formation Control of Nonholonomic Mobile Robots by Bounded Feedback in the Presence of Obstacles “

ΠΕΡΙΕΧΟΜΕΝΑ:

1. ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ.....σελ. 2	2
2. ΣΥΝΘΕΣΗ ΠΡΟΒΛΗΜΑΤΟΣ.....σελ. 3	3
3. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....σελ. 5	5
4. ΥΛΟΠΟΙΗΣΗ ΣΕ MATLAB/SIMULINK.....σελ. 8	8
5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ.....σελ. 16	16
6.ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΑΡΑΤΗΡΗΣΕΙΣ.....σελ. 21	21

ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ :

Σε αυτήν την εργασία , θα υλοποιήσουμε έναν κατανεμημένο έλεγχο για τον σχηματισμό μη ολονομικών κινητών robot(πράκτορες) υπό την παρουσία εμποδίων χρησιμοποιώντας περιορισμένες τιμές στα μεγέθη που θέλουμε να ελέγξουμε.

Οι πράκτορες είναι σχεδιασμένοι να ακολουθούν μια συγκεκριμένη πορεία χωρίς να υπάρχει σύγκρουση μεταξύ άλλων πρακτόρων αλλά και είναι δυνατή η αποφυγή τυχόν εμποδίων κατά την διάρκεια της πορείας τους.

Για την αποφυγή συγκρούσεων μεταξύ πρακτόρων χρησιμοποιούμε έναν νόμο ελέγχου βασισμένο σε απωθητικές δυνάμεις. Η δημιουργία των απωθητικών δυνάμεων έχει ως συνέπεια την αλλαγή της φοράς περιστροφής που επιτρέπει στους πράκτορες την αποφυγή και την προσπέραση εμποδίων.

Με την εργασία αυτή , θέλουμε να προσομοιώσουμε ένα σύστημα λαμβάνοντας υπόψιν τους φυσικούς περιορισμούς που επιβάλλονται σε αυτό , οπότε χρησιμοποιούμε πεπερασμένα όρια στις εισόδους του κάθε πράκτορα.

ΣΥΝΘΕΣΗ ΠΡΟΒΛΗΜΑΤΟΣ :

Το συλλογικό σύστημα που θα χρησιμοποιήσουμε αποτελείται από N πανομοιότυπα αυτόνομα κινητά robot , των οποίων η δυναμική περιγράφεται από τις παρακάτω εξισώσεις :

$$\begin{aligned}\dot{x}_i &= v_i \cos(\theta_i) \\ \dot{y}_i &= v_i \sin(\theta_i) \\ \dot{\theta}_i &= u_i\end{aligned}$$

όπου $i = 1, \dots, N$

$p_i = [x_i, y_i]^T$, θ_i η θέση και η γωνία προσορισμού του i -οστού robot στο καρτεσιανό σύστημα

v_i η γραμμική ταχύτητα του i -οστού robot

u_i η γωνιακή ταχύτητα του i -οστού robot

Το πρόβλημα του ελέγχου τροχιάς για το παραπάνω σύστημα είναι να εκφράσουμε τις μεταβλητές v_i και u_i ως συναρτήσεις της θέσης , της γωνίας , της ταχύτητας και της γωνιακής ταχύτητας με πεπερασμένο εύρος τιμών με έναν κατανομημένο τρόπο ώστε να επιτευχθούν οι παρακάτω στόχοι :

1) Ακολουθία Τροχιάς :

$$\lim_{t \rightarrow \infty} (\dot{p}_i(t) - \dot{p}_{id}(t)) = 0, \forall i, j = 1, \dots, N$$

όπου p_{id} είναι η τροχιά αναφοράς του i πράκτορα , η οποία παράγεται από συναρτήσεις στο καρτεσιανό επίπεδο.

2) Αποφυγή Συγκρούσεων :

$$r_{ij}(t) = \|p_i(t) - p_j(t)\| \geq r_0, \forall t \geq 0, \forall i \neq j$$

3) Αποφυγή Εμποδίων :

$$r_{ij}(t) \leq R_0, \forall t \geq 0, \forall i \neq j$$

Για να επιτύχουμε τον 2ο και τον 3ο στόχο , θεωρούμε μια συνάρτηση συντονισμού

$$V_a = \sum_{i=1}^{i=N} \sum_{j=1, j \neq i}^{j=N} V_{ij} = \sum_{i=1}^{i=N} \sum_{j=1, j \neq i}^{j=N} V(r_{ij})$$

η οποία έχει τις παρακάτω ιδιότητες :

A) Υπάρχει θετική σταθερά V_m και μια μεταβλητή r , που εκφράζει απόσταση , στο διάστημα $[a, b]$ τέτοια ώστε :

$$0 \leq V(r) \leq V_m$$

B) Η $V(r)$ είναι συνεχώς παραγωγίσιμη και γνησίως φθίνουσα στο διάστημα $[a, b]$.

$$\Gamma) \quad \lim_{r \rightarrow a^+} V(r) = Vm$$

Παρατηρούμε ότι διατηρώντας την συνάρτηση συντονισμού μικρότερη του Vm , υπονοούμε ότι η απόσταση μεταξύ δύο robot i και j είναι μεγαλύτερη από την ελάχιστη τιμή a . Με βάση την παραπάνω παρατήρηση, θα προσπαθήσουμε να διατηρήσουμε την απόσταση μεταξύ πράκτορα – πράκτορα ή πράκτορα – αντικείμενο μεγαλύτερη από τιμή a που θα ορίσουμε αργότερα.

Τέλος, ορίζουμε επίσης την περιοχή αποφυγής

$$\Omega = \{p \in \mathbb{R}^2 \mid \|p_i - p_a\| \leq a\}$$

και την περιοχή ανίχνευσης

$$\Gamma = \{p \notin \Omega \mid a < \|p_i - p_a\| \leq b\}$$

όπου $p_a = [x_a, y_a]^T$ είναι η θέση του αντικειμένου ή του πράκτορα που θα αποφύγει ο i -οστός πράκτορας.

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ:

Σε αυτό το σημείο , θα αναλύσουμε τον έλεγχο που θα χρησιμοποιήσουμε για να υλοποιήσουμε τους 3 στόχους που αναφέραμε παραπάνω. Θα χωρίσουμε την ανάλυση σε 3 κομμάτια. Αρχικά , θα αναλύσουμε τον αλγόριθμο ελέγχου ώστε ο κάθε πράκτορας να ανιχνεύει την τροχιά που δίνουμε ως αναφορά (Trajectory Tracking Control). Έπειτα , την αποφυγή συγκρούσεων (Collision Avoidance) μεταξύ πρακτόρων χρησιμοποιώντας μια συνάρτηση δυναμικού. Τέλος , την αποφυγή εμποδίων (Obstacle Avoidance) με χρήση ενός εικονικού πράκτορα.

A) Trajectory Tracking Control:

Υποθέτουμε ότι η επιθυμητή τροχιά περιγράφεται από τις μεταβλητές $x_{di}(t)$ και $y_{di}(t)$, οι οποίες έχουν πεπερασμένες παραγώγους.

Ορίζουμε το σφάλμα θέσης στον άξονα x , στον άξονα y και το σφάλμα γωνίας :

$$e_{xi}(t) = x_i(t) - x_{di}(t) \quad e_{yi}(t) = y_i(t) - y_{di}(t)$$

$$e_{\theta i} = \theta_i(t) - \theta_{di}(t)$$

και η επιθυμητή γωνία προορισμού είναι :

$$\theta_{di} = \text{atan2}(-e_{yi}(t), -e_{xi}(t))$$

Ο στόχος μας είναι να οδηγήσουμε το κάθε ρομπότ στην επιθυμητή του τροχιά από μια τυχαία αρχική θέση με πεπερασμένη γραμμική και γωνιακή ταχύτητα.

Κάνουμε τις εξής υποθέσεις :

α) Το $\cos(e_{\theta i}(t))$ είναι διάφορο του μηδενός

β) Ορίζουμε μια εκτίμηση της παραγώγου της επιθυμητής γωνίας προορισμού :

$$\dot{\theta} = \frac{e_{xi}(t)\dot{e}_{yi} - e_{yi}(t)\dot{e}_{xi}}{D_i^2}$$

όπου

$$D_i = \sqrt{e_{xi}^2 + e_{yi}^2}$$

$$\dot{e}_{xi} = \frac{e_{xi}(t) - e_{xi}(t - T)}{T} \quad \dot{e}_{yi} = \frac{e_{yi}(t) - e_{yi}(t - T)}{T}$$

για κάποια μικρή τιμή του T .

Πρέπει να τονίσουμε η απόλυτη διαφορά μεταξύ του σφάλματος της παραγώγου της γωνίας και της εκτίμησης αυτού είναι πεπερασμένη και συγκεκριμένα είναι της τάξης της περιόδου T .

Με βάση τα παραπάνω , ο ελεγκτής είναι :

$$v_i = K_i \cos(e_{\theta i}) \min(D_{maxi}, D_i)$$

$$u_i = -K_{\theta i} e_{\theta i} + \dot{\theta}_{di}$$

όπου $K_{\theta i}$, K_i και D_{maxi} είναι θετικές παράμετροι που ορίζονται από τον σχεδιαστή.

Ο ελεγκτής είναι ευσταθής κατά Lyapunov και αξίζει να παρατηρήσουμε ότι ο έλεγχος για την ταχύτητα είναι πεπερασμένος από την τιμή του D_{\max} .

Επειδή η θέση του αντικειμένου (x,y) εξαρτάται από την ταχύτητα άρα και τα αντίστοιχα σφάλματα καταλήγουμε ότι η εκτίμηση της παραγώγου της επιθυμητής γωνίας είναι πεπερασμένη, όπως και η επιθυμητή γωνία. Άρα συμπεραίνουμε ότι και ο έλεγχος για την γωνιακή ταχύτητα είναι πεπερασμένος.

B) Collision Avoidance:

Η στρατηγική για την αποφυγή άλλων robot είναι βασισμένη σε μια συνάρτηση δυναμικού, η οποία ασκεί μια απωθητική δύναμη για κάθε πράκτορα. Η συνάρτηση δυναμικού είναι:

$$V_{ij}(p_i, p_j) = K_{ij} \ln(\cosh(p_{ij})) h_{ij}(p_i, p_j)$$

όπου $K_{ij} > 0$

$$p_{ij} = \|p_i - p_j\| - c_i$$

$$h_{ij}(p_i, p_j) = \begin{cases} 1, & \|p_i - p_j\| < b_i \\ 0, & \text{else} \end{cases}$$

και $0 < a_i < b_i < c_i$.

Οι παράμετροι a_i και b_i είναι οι παράμετροι που ορίσαμε στην περιοχή αποφυγής και στην περιοχή ανίχνευσης αντίστοιχα.

Η ολική δυναμική ενέργεια του πράκτορα από όλους τους γειτονικούς είναι:

$$U_i(p_i) = \sum_{j \in N_i(t)} V_{ij}(p_i, p_j)$$

όπου $N_i(t)$ εκφράζει τα robot που βρίσκονται σε μικρή απόσταση από το i-οστό robot. Η μεταβολή της δυναμικής ενέργειας είναι η δύναμη που αλληλεπιδρά μεταξύ του i-οστού και του j-οστού πράκτορα:

$$f_{ij}(p_i, p_j) = \Delta V_{ij} = K_{ij} \tanh(p_{ij}) \frac{p_j - p_i}{\|p_j - p_i\|}$$

Η δύναμη αυτή εκφράζει μια απωθητική δύναμη μόνο όταν η βηματική συνάρτηση h_{ij} είναι ενεργή, αλλιώς είναι μηδενική. Εάν και άλλοι πράκτορες πλησιάζουν τον i-οστό πράκτορα στην περιοχή αποφυγής, η ολική δύναμη είναι:

$$F_i(p_i) = \sum_{j \in N_i(t)} f_{ij}(p_i, p_j)$$

με f_{ix} και f_{iy} είναι οι συνιστώσες της ολικής δύναμης στον άξονα x και στον άξονα y αντίστοιχα. Τέλος, ορίζουμε την επιθυμητή γωνία:

$$\theta_{di} = \text{atan2}(-f_{iy}(t), -f_{ix}(t))$$

και την ποσότητα D:

$$D_i = \sqrt{f_{ix}^2 + f_{iy}^2}$$

Η αποφυγή εμποδίων είναι εγγυημένη εάν το σύστημα των N robot αρχικοποιηθεί έτσι ώστε $V_m > V_a$ και οι νόμοι ελέγχου είναι:

$$\begin{aligned} v_i &= K_i \cos(e_{\theta i}) \min(D_{maxi}, D_i) \\ u_i &= -K_{\theta i} e_{\theta i} \end{aligned}$$

όπου $K_{\theta i}$, K_i , D_{maxi} είναι θετικές παράμετροι που ορίζονται από τον σχεδιαστή και

$$V_m = K_{ij} \ln(\cosh(a_i - c_i)) \quad V_a = \sum_{i=1}^{i=N} \sum_{j \in N_i(t)} V_{ij}$$

Επειδή η παράγωγος του V_a είναι γνησίως φθίνουσα και η συνάρτηση δυναμικού είναι αντιστρόφως ανάλογη της απόστασης μεταξύ πρακτόρων, συμπεραίνουμε ότι οι πράκτορες δεν θα φτάσουν ποτέ στην περιοχή αποφυγής.

Γ) Obstacle Avoidance:

Επειδή τα εμπόδια μπορεί να έχουν μια πληθώρα σχημάτων και μεγέθους, θεωρούμε ως εμπόδιο τον περιγεγραμμένο κύκλο με κέντρο O_k (x_k , y_k) και ακτίνα r_k . Η προβολή του i-οστού πράκτορα πάνω στην επιφάνεια του εμποδίου είναι:

$$p_{obs} = \frac{r_{obs}}{\|p_i - O_k\|} p_i + \left(1 - \frac{r_{obs}}{\|p_i - O_k\|}\right) O_k$$

όπου $p_{obs} = [x_{obs}, y_{obs}]^T$.

Η προβολή του αντικειμένου μπορεί να θεωρηθεί ως ένα εικονικό robot με την δικιά του θέση στους άξονες x και y. Η αλληλεπίδραση μεταξύ του i-οστού πράκτορα και του εικονικού περιγράφεται με την συνάρτηση δυναμικού που αναφέραμε στο Collision Avoidance. Ο νόμος ελέγχου είναι ίδιος με τις σχέσεις που περιγράψαμε στο (B) με μόνη διαφορά την επιθυμητή γωνία προσανατολισμού που επιλέγεται ως προς την εφαπτομενική διεύθυνση των ορίων του αντικειμένου που πρέπει να προσπεράσει ο πράκτορας.

Με βάση τα παραπάνω, η επιθυμητή γωνία επιλέγεται ως:

$$\theta_{di} = -\frac{\pi}{2} + \gamma + \beta, \gamma > 0 \quad \text{ή} \quad \theta_{di} = +\frac{\pi}{2} + \gamma + \beta, \gamma \leq 0$$

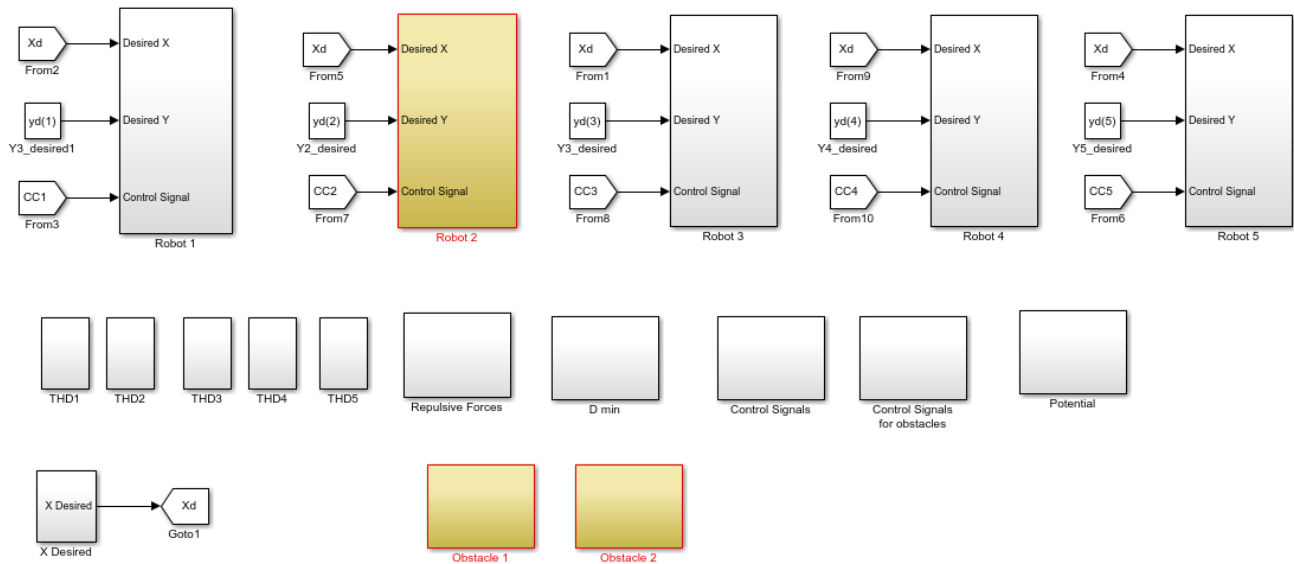
όπου γ είναι η γωνία που μετράμε από την προορισμού του i-οστού πράκτορα μέχρι την ευθεία που ενώνει τον i-οστό πράκτορα με το εμπόδιο και

β η γωνία ανάμεσα στο διάνυσμα που ενώνει το robot με τον επιθυμητό προορισμό και τον άξονα x.

Η επιλογή των ελέγχων γίνεται με βάση την περιοχή ανίχνευσης ανάμεσα στον πράκτορα i και στους άλλους πράκτορες ή εμπόδια. Εάν είναι ενεργή, χρησιμοποιούμε τις σχέσεις του (B) αλλιώς του (A).

ΥΛΟΠΟΙΗΣΗ ΜΕ MATLAB/SIMULINK :

Για την υλοποίηση της συγκεκριμένης εργασίας , χρησιμοποιήσαμε το παρακάτω αρχείο Simulink

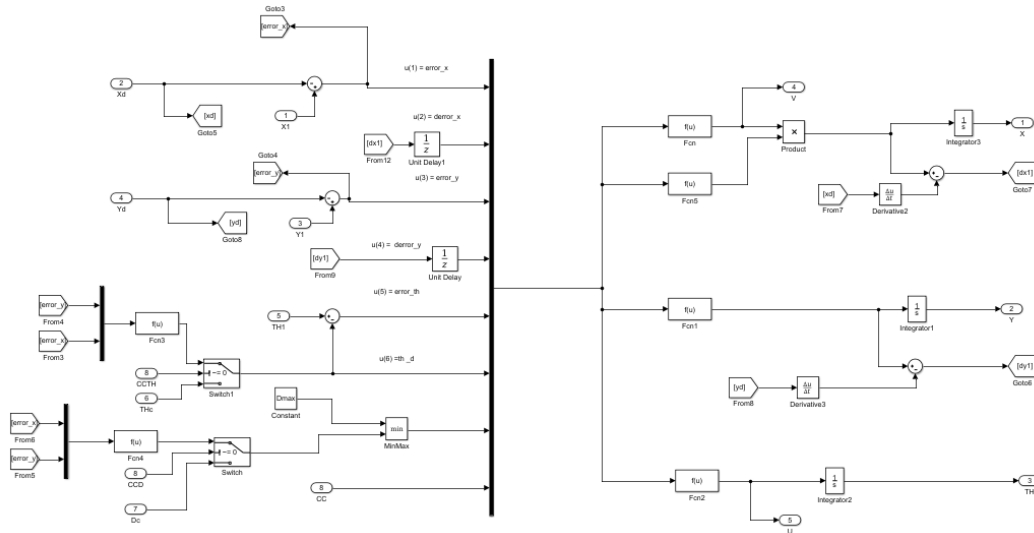


που αποτελείται από τα εξής υποσυστήματα :

Robot i , *THDi* , *Repulsive Forces* , *Dmin* , *Control Signals* , *Potential* , *Xdesired* και *Obstacle* τα οποία θα αναλύσουμε παρακάτω.

- Robot :

Το υποσύστημα αυτό δέχεται ως εισόδους την επιθυμητή τροχιά στους άξονες x και y όπως και το σήμα εισόδου CC για την αλλαγή του νόμου ελέγχου. Όταν το σήμα είναι 1 , ενεργοποιείται ο έλεγχος για την ακολούθηση της επιθυμητής τροχιάς εν,ω όταν είναι 0 , ενεργοποιείται ο έλεγχος για την αποφυγή συγκρούσεων. Επίσης , καθώς αλλάζει ο έλεγχος , αλλάζουν και οι παράμετροι D_i και θ_{di} , όπως αναφέραμε στην θεωρία. Αφού ανοίξουμε το υποσύστημα , βλέπουμε το παρακάτω σχήμα



Το σχήμα αυτό έχει ως εισόδους το σφάλμα της θέσης στον άξονα x (error_x) από την επιθυμητή, την παράγωγο αυτού (derror_x), το σφάλμα της θέσης στον άξονα y (error_y) και αντίστοιχα την παράγωγο του (derror_y), το σφάλμα της γωνίας προσανατολισμού (error_th), την επιθυμητή γωνία (thdi), το ελάχιστο μεταξύ των παραμέτρων D_i και D_{max} και τέλος, το σήμα ελέγχου (CC).

Με αυτόν τον τρόπο, επιλύουμε τις διαφορικές εξισώσεις που εκφράζουν την κίνηση του ρομπότ:

$$\begin{aligned}\dot{x}_i &= K_i \cos(e_{\theta_i}) \min(D_{max}, D_i) \cos(\theta_i) \\ \dot{y}_i &= K_i \cos(e_{\theta_i}) \min(D_{max}, D_i) \sin(\theta_i) \\ \dot{\theta}_i &= K_{\theta_i} e_{\theta_i} + CC \left(\frac{e_x \dot{e}_y - e_y \dot{e}_x}{D_i^2} \right)\end{aligned}$$

Η γωνία θ_i υπολογίζεται από το άθροισμα της επιθυμητής γωνίας και του σφάλματος. Βλέπουμε ότι με την αλλαγή των παραμέτρων D_i και θ_{di} και τις τιμές της μεταβλητής CC μπορούμε να προσομοιώσουμε και τις δυο περιπτώσεις σε ένα μόνο υποσύστημα, βελτιώνοντας έτσι την απόκριση κατά την μετάβαση από τον ένα έλεγχο στον άλλο.

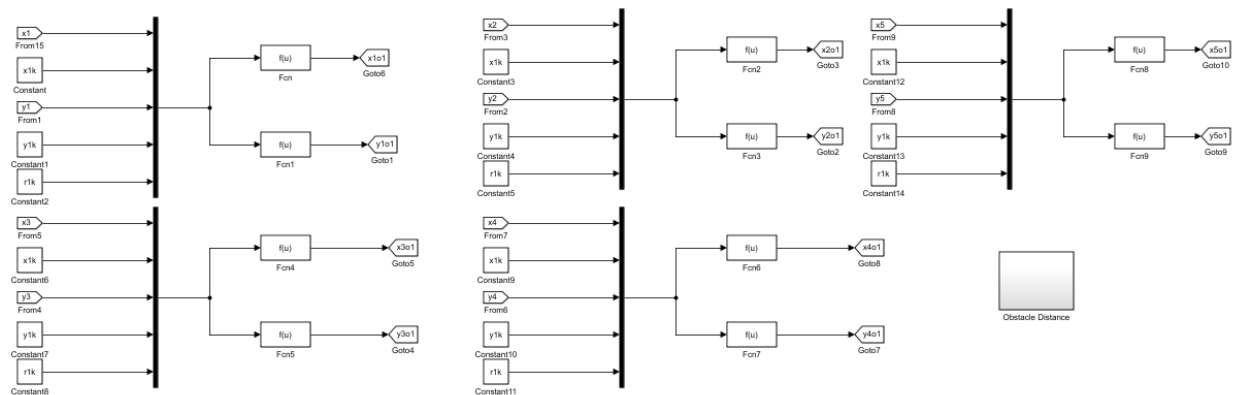
Τα unit delays τοποθετούνται ώστε να απαλλαγούμε από τα algebraic loops που εισέρχονται στην προσομοίωση και οι ολοκληρωτές έχουν ως αρχικές συνθήκες από που δίνουμε από τον κώδικα MATLAB.

Τέλος, οι έξοδοι που λαμβάνουμε είναι η θέση στους άξονες x και y (X,Y), η γωνία (TH) και η γραμμική και γωνιακή ταχύτητα (V, U). Αυτές οι τιμές απεικονίζονται με Scores και αποθηκεύονται σε πίνακες για την δημιουργία των γραφικών παραστάσεων.

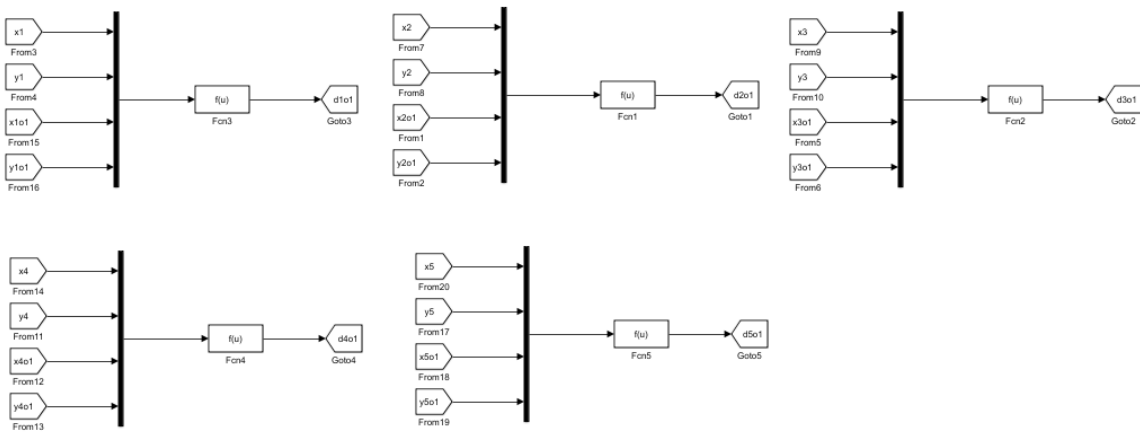
Τα υπόλοιπα robot δουλεύουν με την ίδια ακριβώς λογική.

- Obstacle :

Εδώ υπολογίζουμε τους εικονικούς πράκτορες που προκύπτουν από κάθε ρομπότ στην επιφάνεια του αντικειμένου



όπως αναφέραμε στο θεωρητικό υπόβαθρο και στο υποσύστημα υπολογίζουμε τις αποστάσεις μεταξύ ρομπότ και εικονικών πρακτόρων:

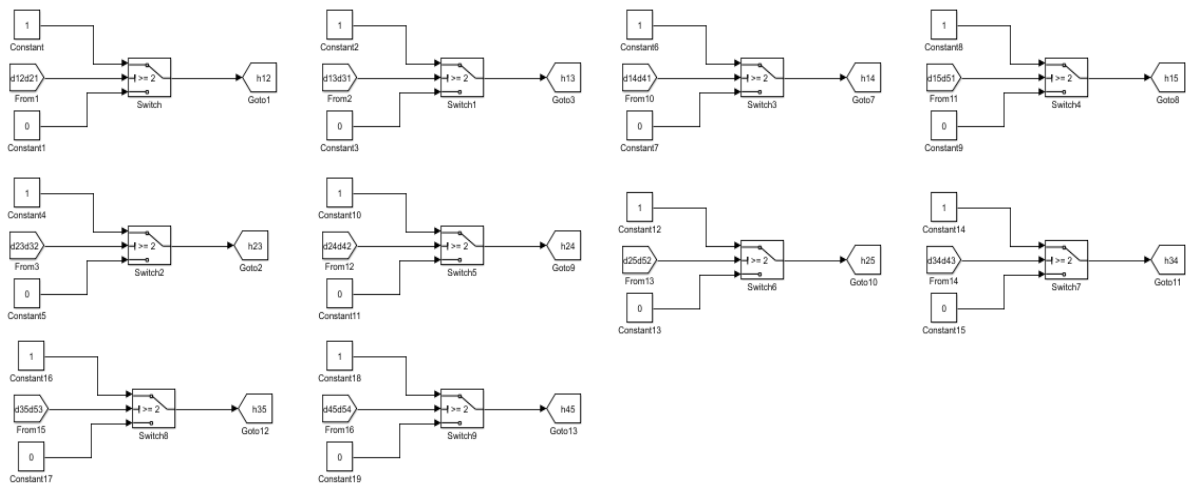


Πρέπει να τονίσουμε ότι κάθε ρομπότ έχει τον δικό του εικονικό πράκτορα και προφανώς και την δικιά του απόσταση από αυτόν.

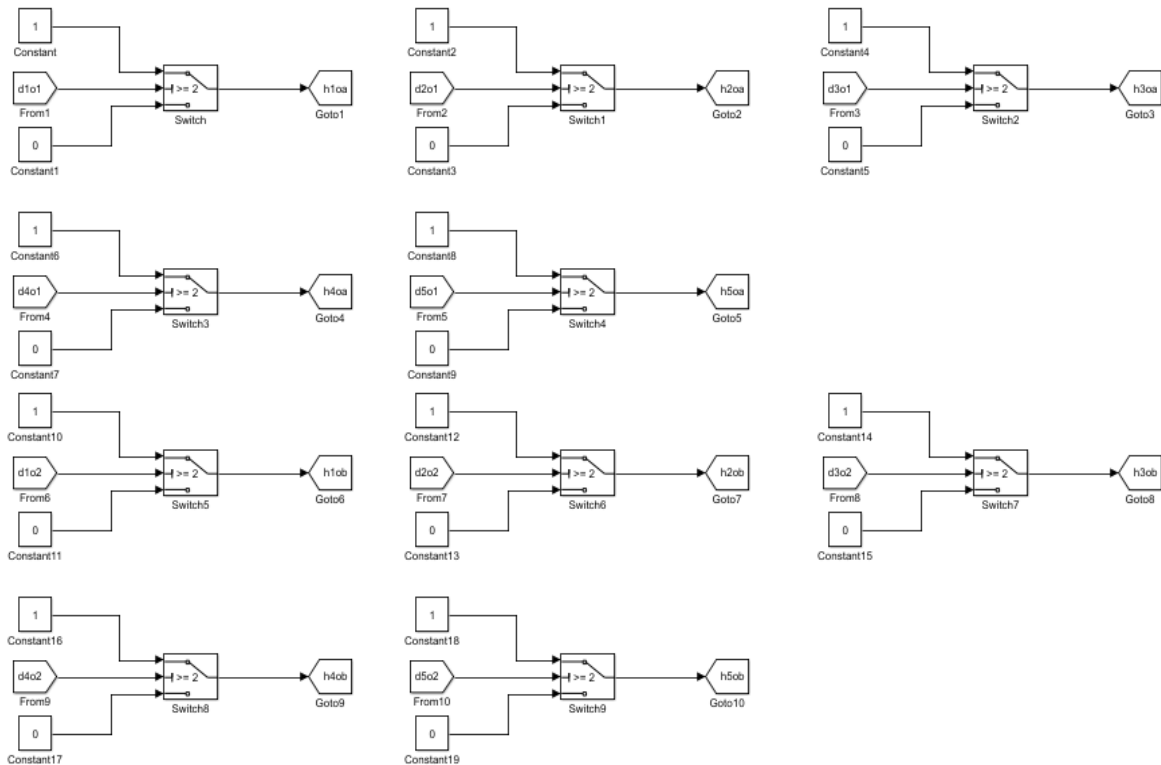
‘Όλα τα εμπόδια δουλεύουν με τον ίδιο τρόπο.

- Control Signals/Control Signals for Obstacles :

Η συνθήκη για το πότε αλλάζουν οι νόμοι ελέγχου είναι το αν η περιοχή ανίχνευσης είναι ενεργή μεταξύ δύο robot ή ενός robot και του αντίστοιχού εικονικού. Αυτό μεταφράζεται με το αν η απόσταση μεταξύ τους είναι μικρότερη της παραμέτρου b που ορίζει ο σχεδιαστής. Παρακάτω φαίνονται όλοι οι επιμέρους έλεγχοι hij για κάθε robot

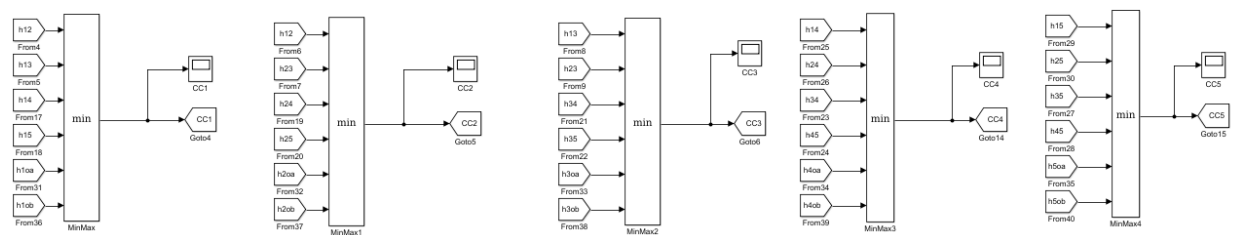


και για κάθε robot και εικονικό πράκτορα



Πρέπει να τονίσουμε ότι $h_{ij} = h_{ji}$ και ότι τοποθετούμε 0 ή 1 με αντίθετο τρόπο από αυτόν που προτείνει ο συγγραφέας ώστε να υπολογίζονται με ευκολία οι νόμοι ελέγχου της κίνησης του robot.

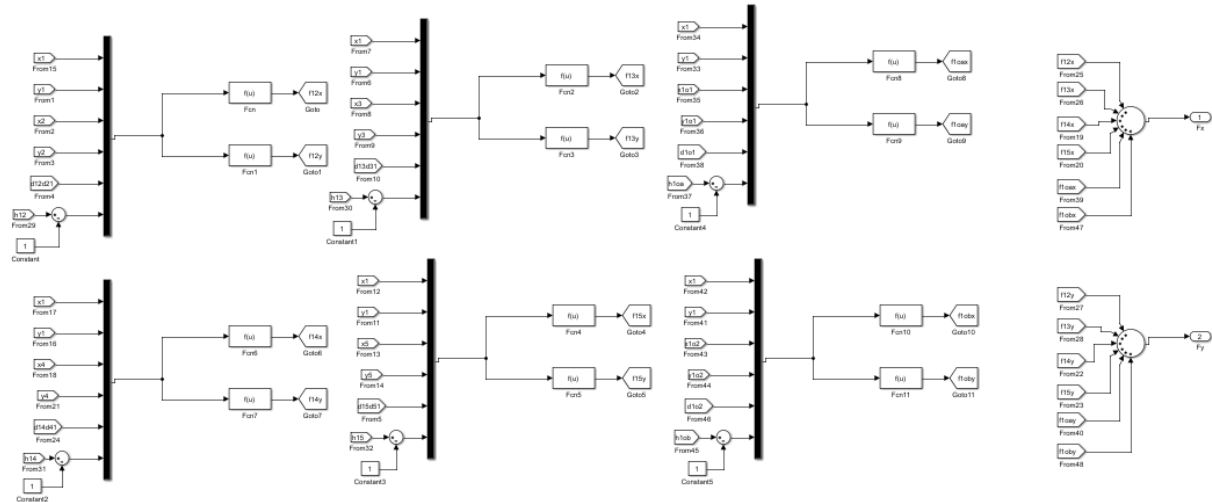
Για να υπολογίσουμε τα σήματα ελέγχου CC, βρίσκουμε την ελάχιστη τιμή από τα επιμέρους h_{ij}



Η στρατηγική που υλοποιήσαμε είναι πλήρως αποδεκτή καθώς μας ενδιαφέρει να αποφεύγουν όλα τα άλλα robot ή εμπόδια και όχι κάποιο συγκεκριμένο στην περίπτωση που βρίσκονται στην περιοχή ανίχνευσης.

- Repulsive Forces :

Οι απωθητικές δυνάμεις μεταξύ robot (εικονικών και μη) υπολογίζονται για όλα τα πραγματικά robot και ύστερα αθροίζονται ώστε να βρούμε την συνισταμένη δύναμη στον άξονα x και y που θα ασκηθεί στον εκάστοτε πράκτορα



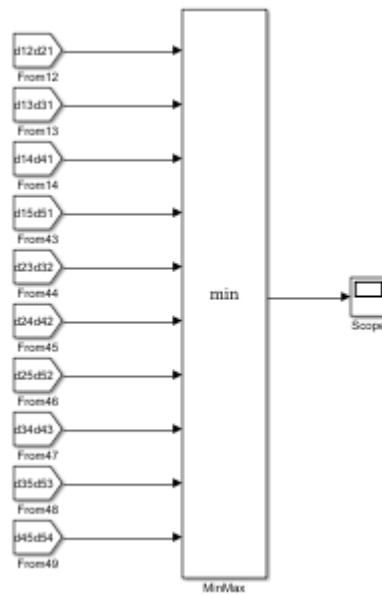
Οι δυνάμεις , σύμφωνα με την θεωρία , είναι διάφορες του μηδενός όταν η συνάρτηση hij είναι ίση με 1. Όμως με την παραδοχή που κάναμε , ορίζοντας την λειτουργία της συνάρτησης αντίστροφα , αφαιρούμε τον αριθμό 1 από το σήμα hij και πολλαπλασιάζουμε με το -1 εντός του κάθε function block στο συγκεκριμένο υποσύστημα. Έτσι έχουμε το ίδιο αποτέλεσμα. Πρέπει να σημειωθεί ότι το fij είναι διάφορο του fji όποτε υπολογίζονται για κάθε robot διαφορετικά. Τα υπόλοιπα υποσυστήματα μέσα έχουν την ίδια λειτουργία.

- Potential :

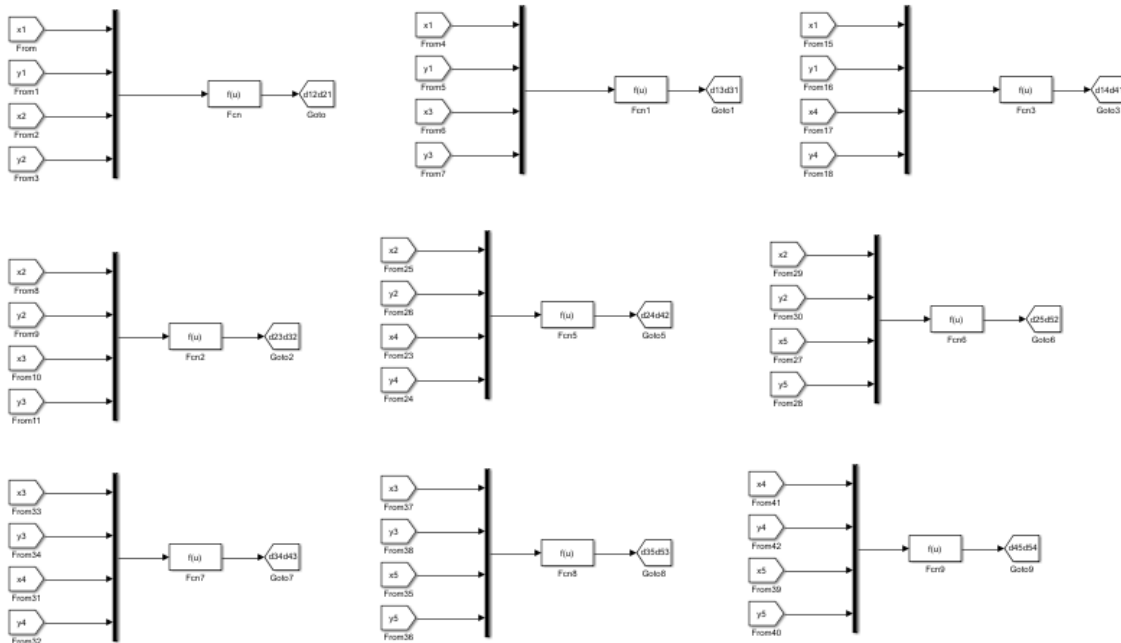
Το ολικό δυναμικό δεν χρησιμοποιείται σε κανένα άλλο σύστημα παρά μόνο μας βοηθάει στο επιλέξουμε τις αρχικές συνθήκες λειτουργίας του συλλογικού συστήματος οπότε δεν θα επεκταθούμε σε περισσότερες λεπτομέρειες για αυτό.

- Dmin :

Εδώ απλά υπολογίζουμε την ελάχιστη απόσταση μεταξύ robot , τοποθετώντας σε ένα block min-max όλες τις αποστάσεις , ώστε να δούμε αν όντως υπάρχει σύγκρουση.



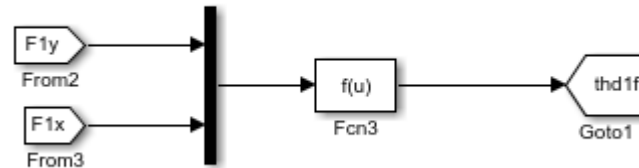
Επίσης , ο υπολογισμός των αποστάσεων μεταξύ πραγματικών πρακτόρων υπολογίζεται εδώ.



- THD :

Υπολογίζουμε την επιθυμητή γωνία μόνο στην περίπτωση που θέλουμε να αποφύγουμε άλλο πράκτορα ή εμπόδιο καθώς στην περίπτωση που θέλουμε να υπολογίσουμε την τροχιά , η γωνία υπολογίζεται εντός του υποσυστήματος του robot.

Όταν θέλουμε να αποφύγουμε άλλο πράκτορα η γωνία υπολογίζεται με βάση τις αποθηκευμένες δυνάμεις σύμφωνα με την θεωρία



Όταν θέλουμε να αποφύγουμε εμπόδιο , τότε για τον υπολογισμό της επιθυμητής γωνίας πρέπει να υπολογίσουμε τις παραμέτρους γ και β , όπως αναφέραμε στην θεωρία.

Η γωνία γ υπολογίζεται ως εξής:

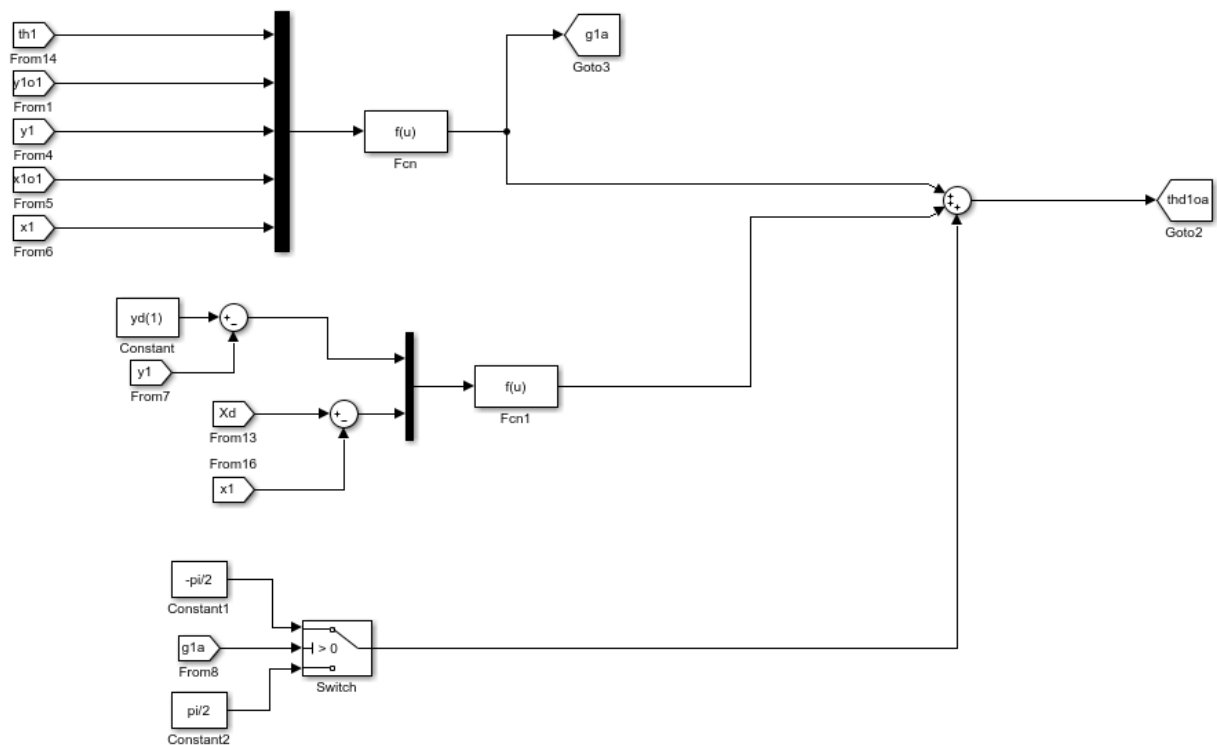
$$\gamma = \theta_i + \text{atan2}(y_{obs} - y_i, x_{obs} - x_i)$$

ενώ η β

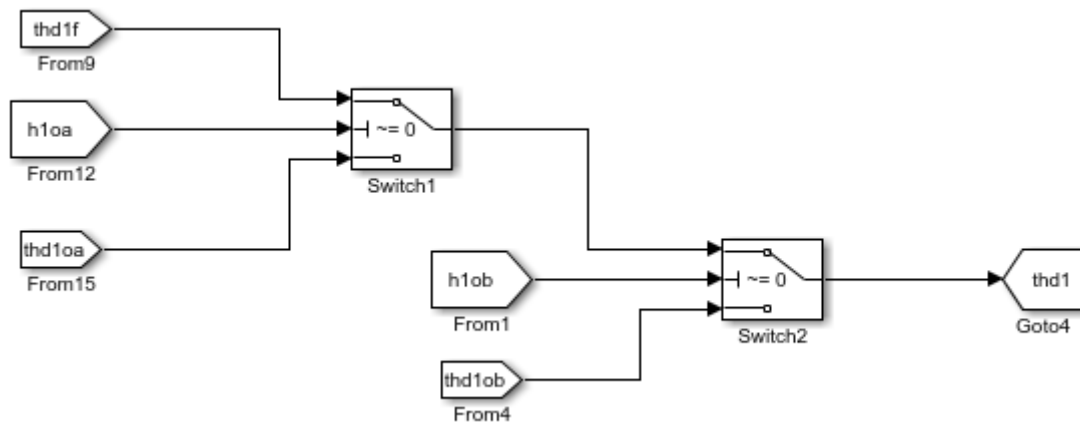
$$\beta = \text{atan2}(y_{di} - y_i, x_{di} - x_i)$$

και εξετάζοντας αν γωνία γ είναι θετική ή όχι προσθέτουμε $-\pi/2$ ή $\pi/2$ αντίστοιχα.

Η διαδικασία αυτή φαίνεται παρακάτω :

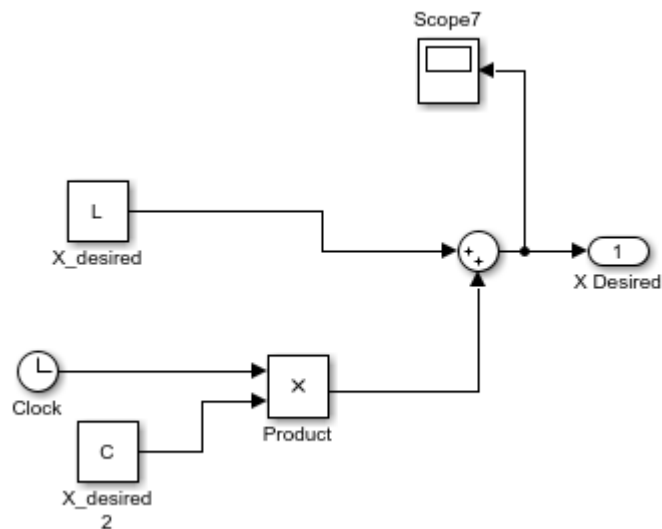


Αφού υπολογίσουμε τις γωνίες , μέσω των σημάτων ελέγχου hij επιλέγουμε την κατάλληλη



Όλα τα υποσυστήματα υπολογίζονται με τον ίδιο τρόπο.

- *Xdesired* :
Υπολογίζουμε την επιθυμητή τροχιά στον άξονα x συναρτήσει του χρόνου και την αποθηκεύουμε μέσω Scope σε ένα πίνακα για την απεικόνιση σε διαγράμματα



Ο κώδικας για την παραπάνω είναι ο project.m και το αρχείο Simulink ονομάζεται rfive.slx.

ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΣΟΜΟΙΩΣΗΣ:

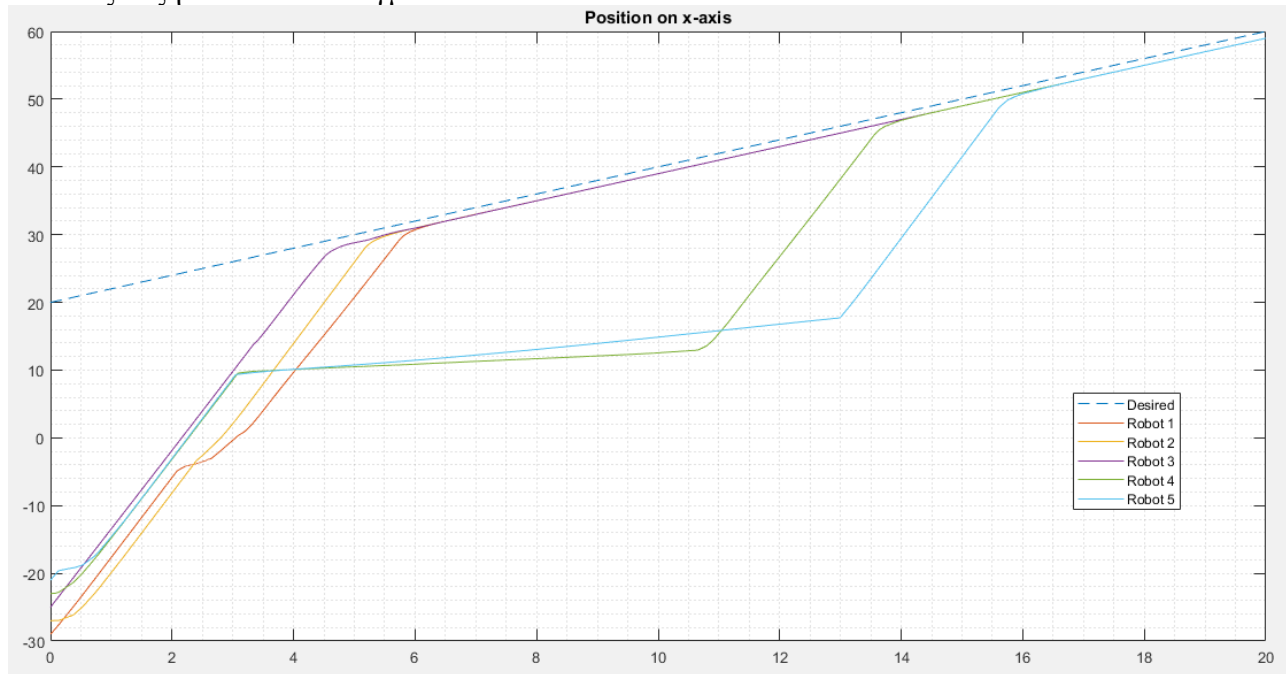
Η προσομοίωση αποτελείται από 5 robot και 2 εμπόδια – αντικείμενα. Οι παράμετροι των δυο αντικειμένων είναι $x_{obs} = [0 ; 15]$, $y_{obs} = [-20 ; -5]$ και $r_{obs} = [3 ; 4]$. Οι παράμετροι της συνάρτησης δυναμικού V_{ij} είναι $K_{ij} = 3$, $a_i = 1$, $b_i = 2$ και $c_i = 4$. Οι παράμετροι για τους νόμους ελέγχου είναι $K_{\theta i} = 3$, $K_i = 4$ και $D_{maxi} = 3$. Η τροχιά που πρέπει να ακολουθήσουν όλοι οι πράκτορες είναι

$$\begin{aligned}x_{di} &= 2t + 20 \\y_{di} &= 4i - 20\end{aligned}$$

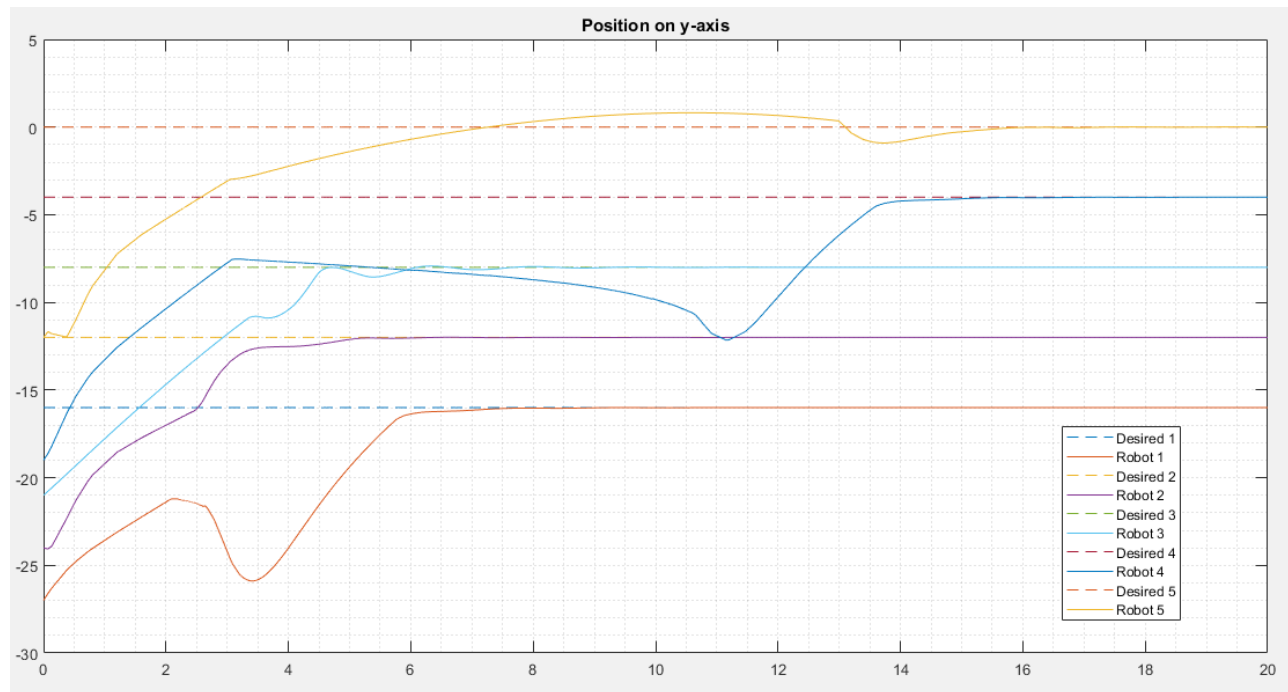
όπου $i = 1, \dots, N$ και t η τρέχουσα χρονική στιγμή. Η προσομοίωση διαρκεί 20 δευτερόλεπτα και οι αρχικές συνθήκες του κάθε robot επιλέγονται ώστε $V_m > V_a$.

Τα αποτελέσματα της προσομοίωσης φαίνονται παρακάτω:

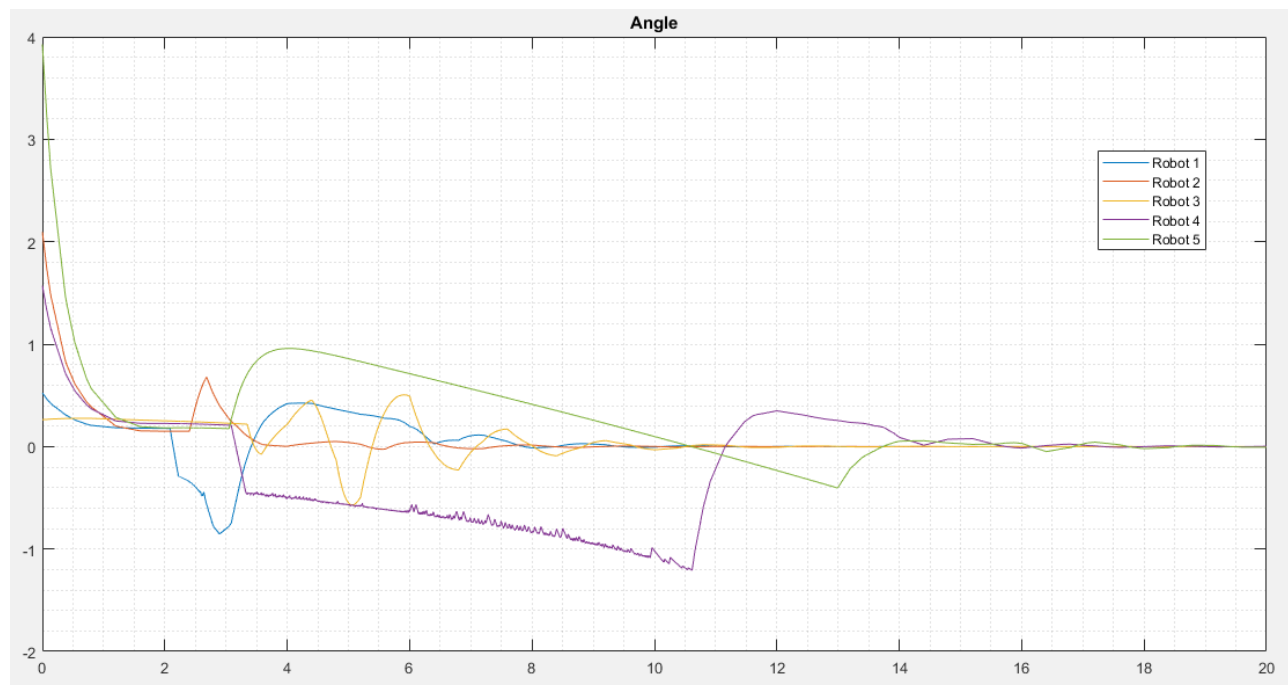
- Εξέλιξη των x συντεταγμένων :



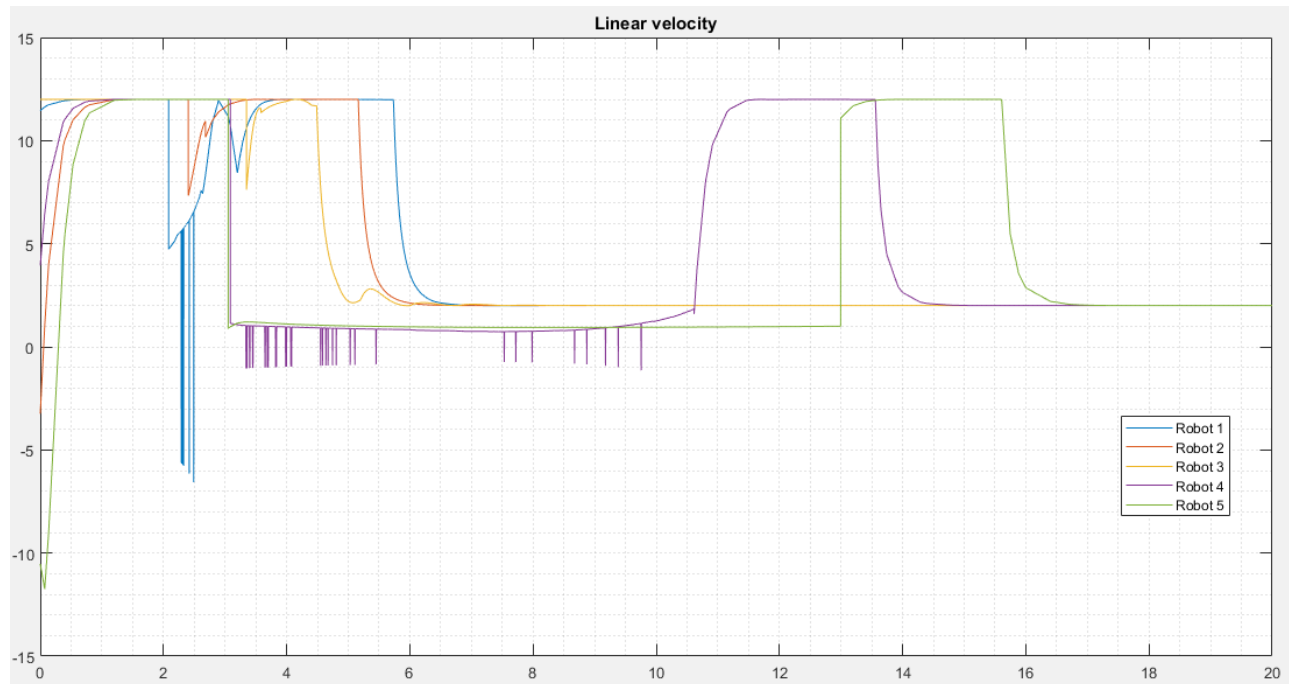
- Εξέλιξη των y συντεταγμένων :



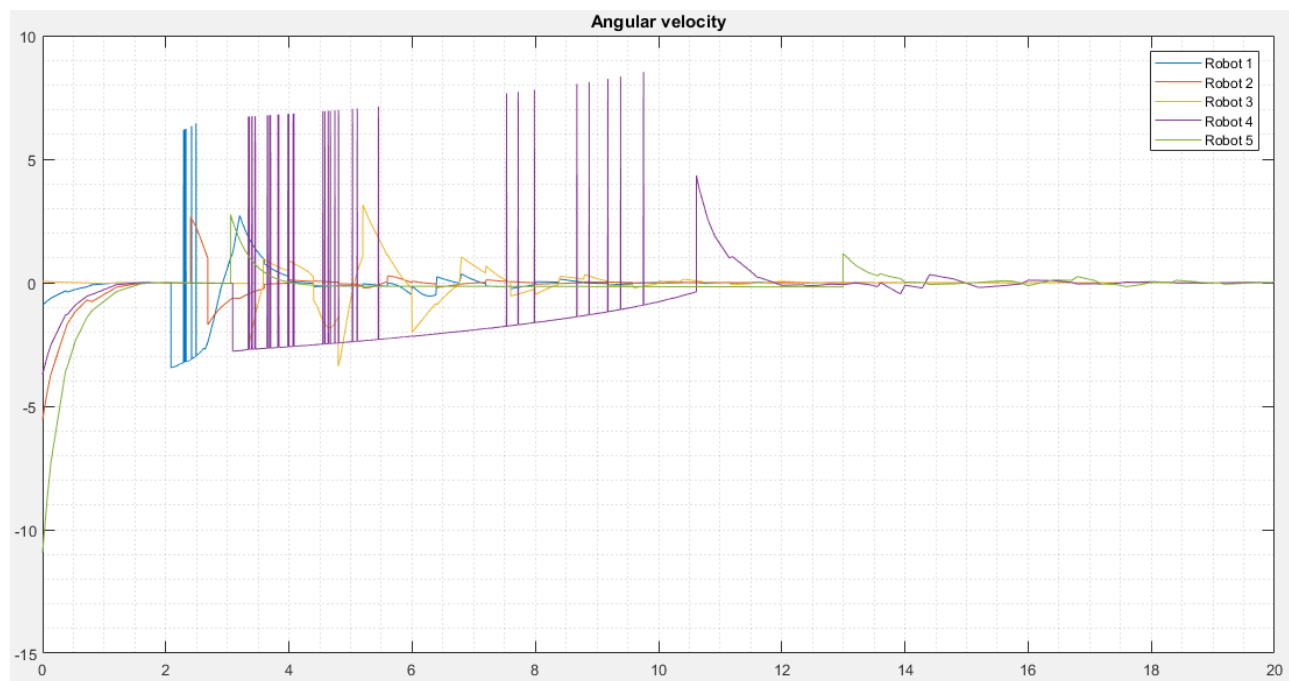
- Γωνίες προορισμού :



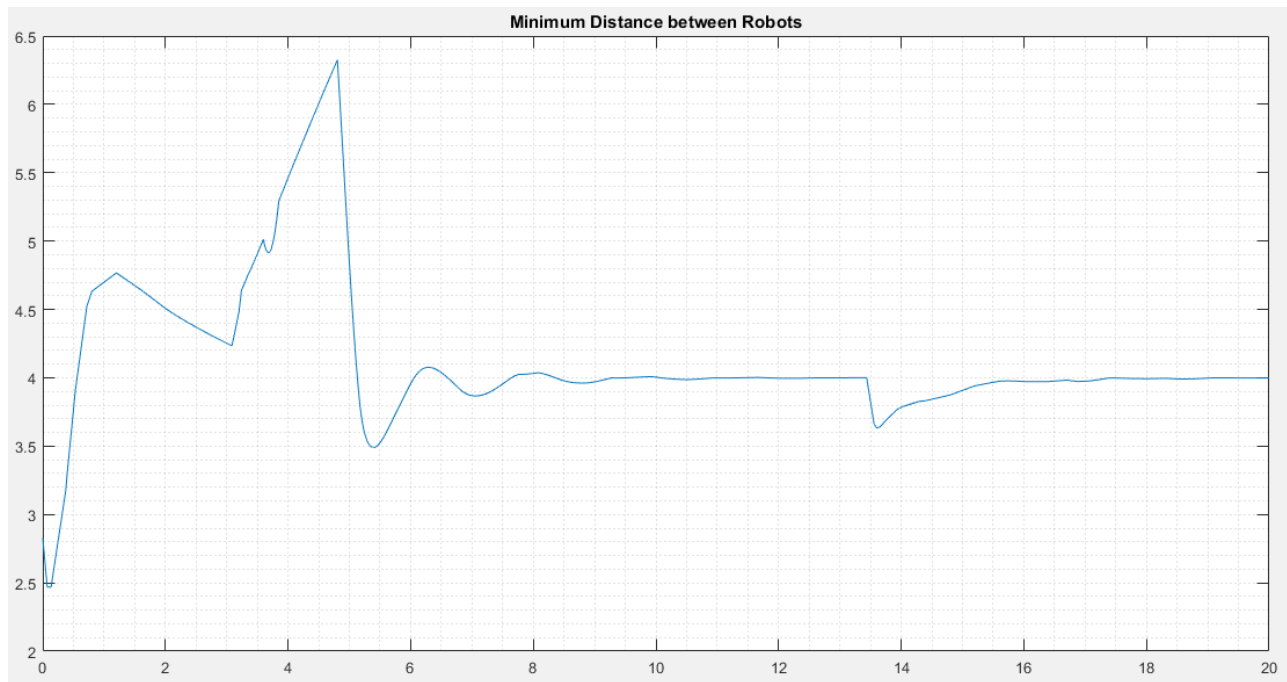
- Γραμμικές ταχύτητες :



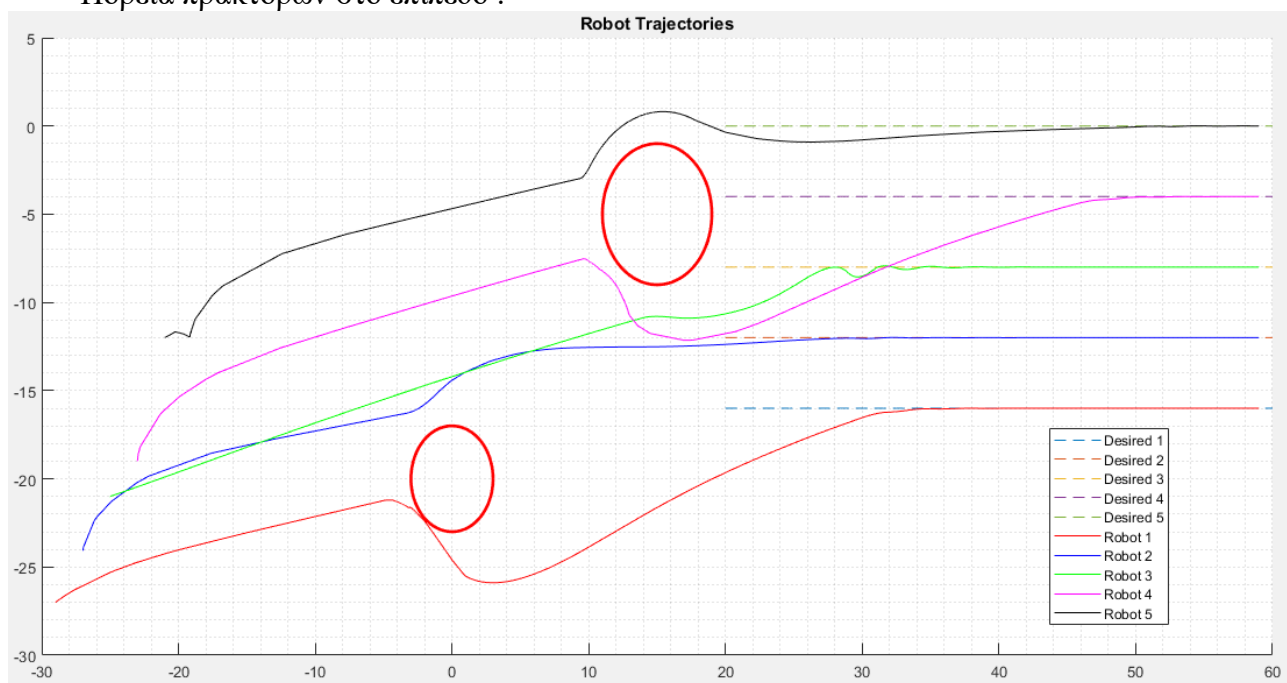
- Γωνιακές ταχύτητες :



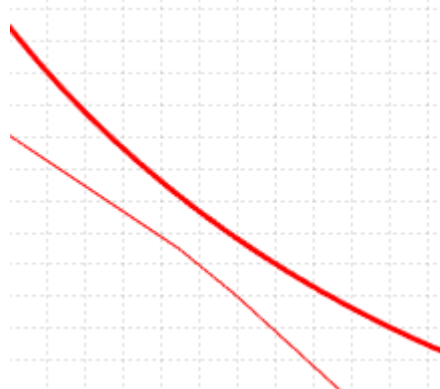
- Ελάχιστες αποστάσεις μεταξύ robot :



- Πορεία πρακτόρων στο επίπεδο :



Βλέπουμε ότι το κόκκινο robot βρίσκεται πάρα πολύ κοντά στο αντικείμενο αλλά δεν το “χτυπάει”



ΣΥΜΠΕΡΑΣΜΑΤΑ-ΠΑΡΑΤΗΡΗΣΕΙΣ:

Συμπεραίνουμε ότι η προσομοίωση έχει τα επιθυμητά αποτελέσματα με την διατριβή καθώς όλα τα robot φτάνουν στον τελικό προορισμό τους , δεν υπάρχει σύγκρουση μεταξύ πρακτόρων όπως βλέπουμε από την γραφική παράσταση με την ελάχιστη απόσταση μεταξύ πρακτόρων και τα robot αποφεύγουν όλα τα εμπόδια χωρίς να συντριβούν.

Παρόλο που προτίνεται η χρήση 9 robot , εμείς χρησιμοποιήσαμε 5 λόγω της μεγάλης πολυπλοκότητας και μεγέθους του αρχείου Simulink καθώς πρέπει να δημιουργηθούν όλες οι αποστάσεις μεταξύ όλων των robot από τα υπόλοιπα αλλά και από όλα τα εμπόδια , όπως και οι αντίστοιχες αποθηκευτικές δυνάμεις και τα σήματα ελέγχου.

Από τις γραφικές παραστάσεις , βλέπουμε ότι όλα τα μεγέθη συγκλίνουν σε μια τελική τιμή και βρίσκονται εντός πεπερασμένων ορίων τιμών.

Η μόνη διαφορά με το paper είναι η ταχύτητα σύγκλισης των ζητούμενων μεγεθών. Αυτό οφείλεται στην χρήση μεγάλου βήματος κατά την προσομοίωση στο Simulink , καθώς η χρήση αρκετά μικρότερου βήματος καθυστερεί στην επίλυση των διαφορικών εξισώσεων και τον υπολογισμό διαφόρων ποσοτήτων και ως συνέπεια όλη την προσομοίωση.