Autonomous Robotic Vehicles

Hellenic Mediterranean University

Lecture 4

Dr. Alina Eqtami

Lecture Goals - Στόχοι σημερινού μαθήματος!

Μετά το σημερινό μάθημα θα μπορείτε:

- Να περιγράφετε το κινηματικό μοντέλο του unicycle και τη σύνδεσή του με διαφορικά και Ackermann ρομποτικά οχήματα.
- Να πραγματοποιείτε τον μετασχηματισμό από καρτεσιανές σε πολικές συντεταγμένες (ρ, α, β).
- Να διατυπώνετε το πρόβλημα **σταθεροποίησης σε σημείο στόχο** για ρομπότ τύπου unicycle.
- Να εξηγείτε τον **νόμο ελέγχου Samson (1990)** και τις **θεμελιώδεις συνθήκες ευστάθειας** για τα κέρδη $(k_\rho, k_\alpha, k_\beta)$.
- Να χρησιμοποιείτε έναν απλό κώδικα σε Python για να προσομοιώνετε την κίνηση ενός unicycle προς στόχο και να σχεδιάζετε την τροχιά του.

Differential Drive Kinematics: a Recap

Δεδομένα:

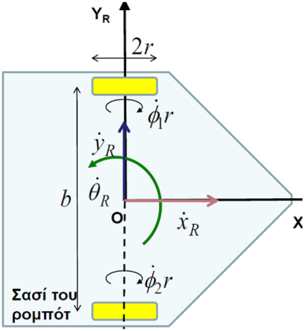
- Απόσταση τροχών: b
- Ακτίνα τροχών: r
- Γωνιακές ταχύτητες: $\dot{\phi}_1$ (δεξιός), $\dot{\phi}_2$ (αριστερός)

Κινηματικό μοντέλο (στο R-frame):

$$\dot{x}_R = \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2)$$

$$\dot{y}_R = 0$$

$$\dot{\theta}_R = \frac{r}{b}(\dot{\phi}_1 - \dot{\phi}_2)$$



Διαφορικά οδηγούμενο ρομπότ.

Παρατηρήσεις:

- Κίνηση πάνω σε ευθεία $\rightarrow \dot{\phi}_1 = \dot{\phi}_2$
- Περιστροφή επί τόπου $\rightarrow \dot{\phi}_1 = -\dot{\phi}_2$

Velocity Mapping for Differential Drive

Απεικόνιση ταχυτήτων για διαφορικό ρομπότ:

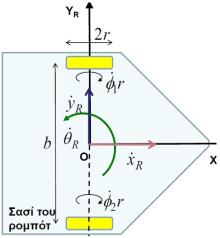
- Ορίζουμε τις **ελεγχόμενες ταχύτητες**:

$$v = \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2), \qquad \omega = \frac{r}{b}(\dot{\phi}_1 - \dot{\phi}_2)$$

- Το v είναι η **γραμμική ταχύτητα** στο μέσο της απόστασης μεταξύ των δύο τροχών.
- Το ω είναι η **γωνιακή ταχύτητα περιστροφής** γύρω από τον κάθετο άξονα.
- Αυτή η παράμετρος μετάβασης:

$$(\dot{\phi}_1, \dot{\phi}_2) \quad \longrightarrow \quad (v, \omega)$$

απλοποιεί το μοντέλο και οδηγεί στη μορφή **unicycle**.



Διαφορικά οδηγούμενο ρομπότ.

Ορισμός των v και ω στο διαφορικό ρομπότ

- Κάθε τροχός έχει **περιφερειακή ταχύτητα**:

$$v_1 = r\,\dot{\phi}_1 \quad (\text{δεξιός}), \qquad v_2 = r\,\dot{\phi}_2 \quad (\text{αριστερός})$$

- Η **γραμμική ταχύτητα του ρομπότ** στο κέντρο του άξονα των τροχών είναι ο **μέσος όρος**:

$$v = \frac{v_1 + v_2}{2} = \frac{r}{2}(\dot{\phi}_1 + \dot{\phi}_2)$$

- Η **γωνιακή ταχύτητα περιστροφής** γύρω από τον κάθετο άξονα προκύπτει από τη **διαφορά** ταχυτήτων:

$$\omega = \frac{v_1 - v_2}{b} = \frac{r}{b}(\dot{\phi}_1 - \dot{\phi}_2)$$

- Αν $v_1 = v_2 \Rightarrow \omega = 0$: ευθύγραμμη κίνηση.
- Αν $v_1 \neq v_2$: το ρομπότ στρίβει με ρυθμό ανάλογο της διαφοράς.
- Έχουμε μια **μεταφορική** και μία **περιστροφική** συνιστώσα της ταχύτητας— βάση για το μοντέλο unicycle.

Μοντέλο Unicycle –1

- Στο **παγκόσμιο (inertial) σύστημα αναφοράς**, η κατάσταση και οι ταχύτητες του ρομπότ είναι:

$$\xi_I(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \qquad \dot{\xi}_I(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix}$$

- Στο **πλαίσιο του ρομπότ** (R) έχουμε, αντίστοιχα:

$$\xi_R(t) = \begin{bmatrix} x_R(t) \\ y_R(t) \\ \theta_R(t) \end{bmatrix} \qquad \dot{\xi}_R(t) = \begin{bmatrix} v(t) \\ 0 \\ \omega(t) \end{bmatrix}$$

- Ο μετασχηματισμός από το πλαίσιο R στο I δίνεται από:

$$\dot{\xi}_I(t) = R^\top(\theta(t))\,\dot{\xi}_R(t)$$

- Ο **Πίνακας περιστροφής** δίνεται ως: $R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Μοντέλο Unicycle –2

- Από τον μετασχηματισμό ταχυτήτων προκύπτουν οι **κινηματικές εξισώσεις** του μοντέλου Unicycle:

$$\dot{x}(t) = v(t)\cos\theta(t)$$
$$\dot{y}(t) = v(t)\sin\theta(t)$$
$$\dot{\theta}(t) = \omega(t)$$

- Το μοντέλο αυτό προκύπτει άμεσα από τη χαρτογράφηση των $(\dot{\phi}_1, \dot{\phi}_2)$ σε (v, ω) και τη χρήση του πίνακα περιστροφής.
- Είναι ένα από τα **πιο θεμελιώδη κινηματικά μοντέλα** σε ρομποτική πλοήγηση.

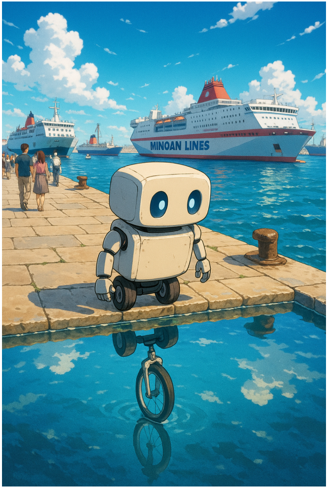
Σχόλιο

Το μοντέλο Unicycle περιγράφει με απλό τρόπο τη μη ολονομική κίνηση στο επίπεδο, χρησιμοποιώντας μόνο δύο ελέγχους (v, ω).

- Το κινηματικό μοντέλο μπορεί να γραφεί σε **matrix form**:

$$\dot{\xi}_I(t) = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$$

- **Non-holonomic**: Υπάρχει κινηματικός περιορισμός πλευρικής κίνησης — η y-συνιστώσα ταχύτητας στο σώμα R είναι **0**.
- **Underactuated**: Η διάσταση της κατάστασης είναι 3 (x, y, θ) αλλά υπάρχουν μόνο 2 είσοδοι (v, ω) $\rightarrow$ Δεν μπορούν να ελεγχθούν ανεξάρτητα και οι τρεις μεταβλητές.



Από Ackermann σε Unicycle —1

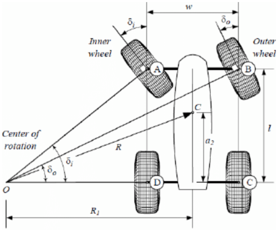
Ένα Ackermann steering όχημα (π.χ. αυτοκίνητο) χαρακτηρίζεται από:

$$\mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad u = \begin{bmatrix} v \\ \delta \end{bmatrix}$$

όπου x, y, θ η στάση στο παγκόσμιο πλαίσιο, v η ταχύτητα του κέντρου των πίσω τροχών και δ η γωνία διεύθυνσης του εμπρόσθιου τροχού.
Οι **κινηματικές εξισώσεις** του Ackermann μοντέλου:

$$\dot{x} = v \cos\theta$$
$$\dot{y} = v \sin\theta$$
$$\dot{\theta} = \frac{v}{L} \tan\delta$$

όπου L το μεταξόνιο.



Ackermann model. A front-wheel-steering vehicle and steer angles of the inner and outer wheels.

*Image source: ResearchGate

Από Ackermann σε Unicycle —2

- **Ackermann (bicycle) κινηματική**:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \frac{v}{L}\tan\delta$$

  όπου v η ταχύτητα στον πίσω άξονα, L το *μεταξόνιο*, δ η γωνία διεύθυνσης (των τροχών, όχι του αμαξιού!).

- **Γεωμετρία στροφής**: Η τροχιά έχει **ακτίνα στροφής**

$$R = \frac{L}{\tan\delta}$$

  και **καμπυλότητα** $\kappa = \frac{1}{R} = \frac{\tan\delta}{L}$.

- Ο **ρυθμός στροφής (yaw rate)** είναι

$$\omega = \dot{\theta} = v\,\kappa = v\,\frac{\tan\delta}{L}$$

  δηλαδή $\boxed{\omega = \frac{v}{L}\tan\delta}$.

- **Διαισθητικά**: Μεγαλύτερο δ μικρότερο R μεγαλύτερη κ γρηγορότερο ω.

Από Ackermann σε Unicycle —3

- Ορισμός εισόδου unicycle:

$$\omega \;:=\; \frac{v}{L}\tan\delta$$

  τότε το Ackermann γράφεται ως:

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = \omega$$

  $\Rightarrow$ **κινηματική ισοδυναμία με unicycle.**
- Μικρές γωνίες διεύθυνσης, άρα μπορούμε να γραμμικοποιήσουμε γύρω απο το $\delta = 0$:

$$\tan\delta \approx \delta \quad (\delta \text{ σε rad}, \ |\delta| \ll 1)$$

  άρα $\omega \approx \frac{v}{L}\delta$.
- **Πότε είναι καλή προσέγγιση**:
  - μικρές–μέτριες δ, μέτριες ταχύτητες v,
  - σχεδιασμός/έλεγχος σε κινηματικό επίπεδο.
- **Σχόλιο**: Στο unicycle ελέγχουμε άμεσα (v, ω), ενώ στο Ackermann ελέγχουμε (v, δ) και ω προκύπτει από τη γεωμετρία $(\omega = \frac{v}{L}\tan\delta)$.

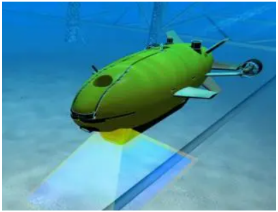
Πλήρες Μοντέλο Υποβρυχίων – 6 DOF

- Τα υποβρύχια οχήματα (UUVs) έχουν **6 βαθμούς ελευθερίας (6-DOF)**:

$$\mathbf{q} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^\top$$

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^\top$$

- x, y, z: θέσεις — ϕ, θ, ψ: γωνίες (roll, pitch, yaw).
- u, v, w: μεταφορικές ταχύτητες (surge, sway, heave).
- p, q, r: στροφές γύρω από τους άξονες (roll, pitch, yaw rates).



A typical UUV.

*Image source: Tactical Report

Τα πλήρη κινηματικά/δυναμικά μοντέλα είναι **πολύπλοκα** και απαιτούν 6 εξισώσεις κίνησης.

Μοντέλο Υποβρυχίων - Απλοποιήσεις για 2D Πλοήγηση

- Στην πράξη, συχνά ενδιαφερόμαστε για **πλοήγηση σε οριζόντιο επίπεδο**:

$$z = \text{σταθερό}, \quad \phi \approx 0, \quad \theta \approx 0$$

(σταθεροποίηση ή μικρές κλίσεις μέσω ελέγχου).

- Αυτό μειώνει το μοντέλο σε:

$$\mathbf{q}_{2D} = \begin{bmatrix} x & y & \psi \end{bmatrix}^\top, \quad \boldsymbol{\nu}_{2D} = \begin{bmatrix} u & v & r \end{bmatrix}^\top$$

- Συνήθως:
  - Έλεγχος μέσω **πρόωσης** στον surge άξονα (u).
  - Έλεγχος yaw ρυθμού (r)
    $\Rightarrow$ Πώς; Είτε με 2 Lateral Thrusters, είτε με Thruster+Rudder
  - **Όχι έλεγχος sway (v) $\rightarrow$ non-holonomic**.
- Σε αυτό το επίπεδο, ένα UUV έχει 3 **καταστάσεις και** 2 **ελέγχους** — άρα είναι **underactuated**.

Κινηματικό Μοντέλο Υποβρυχίου σε 2D – 1

- **Στάση στο παγκόσμιο πλαίσιο (Inertial)**:

$$\xi_I = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix}$$

- **Ταχύτητες στο πλαίσιο του σώματος (Body)**:

$$\dot{\xi}_R = \begin{bmatrix} u \\ 0 \\ r \end{bmatrix}$$

όπου: - u: surge ταχύτητα (μπροστά), - r: yaw rate, - πλευρική ταχύτητα $= 0$ λόγω μη-ολονομικού περιορισμού.

- **Πίνακας περιστροφής Body → Inertial**:

$$R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$$

Κινηματικό Μοντέλο Υποβρυχίου σε 2D – 2

- Εφαρμόζουμε $R(\psi)$ στις μεταφορικές ταχύτητες:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = R(\psi) \begin{bmatrix} u \\ 0 \end{bmatrix} = \begin{bmatrix} u\cos\psi \\ u\sin\psi \end{bmatrix}$$

- Προσθέτοντας τη στροφή γύρω από τον κάθετο άξονα:

$$\dot{\psi} = r$$

- Το πλήρες κινηματικό μοντέλο γράφεται:

$$\dot{\xi}_I = G(\psi) \begin{bmatrix} u \\ r \end{bmatrix}, \qquad G(\psi) = \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix}$$

- Αναλυτικά:

$$\dot{x} = u\cos\psi, \quad \dot{y} = u\sin\psi, \quad \dot{\psi} = r$$

- Αυτό είναι **κινηματικά ισοδύναμο με το μοντέλο unicycle** με $v = u$, $\omega = r$.

Κινηματικό Μοντέλο Υποβρυχίου σε 2D – 3

- Ισοδυναμία με μοντέλο Unicycle:

$$v \equiv u, \quad \omega \equiv r, \quad \theta \equiv \psi$$

- **Non-holonomic:** Δεν υπάρχει ελεγχόμενη πλευρική κίνηση — η κίνηση ευθυγραμμίζεται με τον διαμήκη άξονα.
- **Underactuated:** 3 καταστάσεις (x, y, ψ) με 2 ελέγχους (u, r).
- **Χρήσεις**:
  - Σχεδιασμός τροχιάς (path planning).
  - Ελεγκτές στάσης/πορείας (Heading-Course/ Position-Station Keeping Controlllers).
  - Ευθυγράμμιση (χρήση) με αλγορίθμους διαφορικών ρομπότ.
- **Περιορισμοί προσέγγισης**:
  - Σημαντικά ρεύματα $\Rightarrow$ πλευρική ολίσθηση.
  - Μεγάλες κλίσεις (pitch/roll) ή επιθετικοί ελιγμοί 3D.
  - Πλήρως ενεργοποιημένα UUVs με πλευρικούς προωστήρες $\rightarrow$ το unicycle δεν επαρκεί.

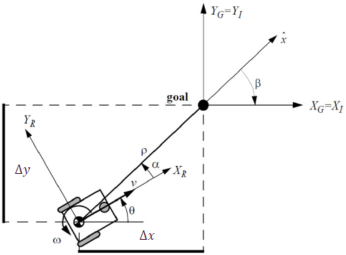
Kinematic Position Control — Motion Control

Έλεγχος Κίνησης

- The kinematics of a unicycle, described in the inertial frame, are:

$$\dot{x} = v\cos\theta$$
$$\dot{y} = v\sin\theta$$
$$\dot{\theta} = \omega$$

- Let α denote the angle between the robot's local x_R axis and the vector connecting the axle center to the final position.
- Motion control aims to design (v, ω) so that the robot reaches the desired position and orientation.



Γεωμετρία ελέγχου κίνησης.

Kinematic Position Control — Polar Coordinates

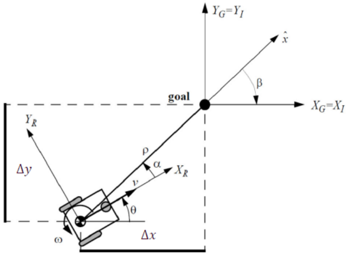
Μετασχηματισμός σε **πολικές συντεταγμένες**:

- Ορίζουμε νέο σύστημα αναφοράς με **αρχή στο σημείο στόχο**.
- Οι μεταβλητές είναι:

$$\rho = \sqrt{(\Delta x)^2 + (\Delta y)^2},$$
$$\alpha = -\theta + \operatorname{atan2}(\Delta y, \Delta x),$$
$$\beta = -\theta - \alpha$$

όπου:

- ρ: απόσταση από το στόχο.
- α: σχετική γωνία στόχου–προσανατολισμού.
- β: προσανατολισμός σε σχέση με το στόχο.



Γεωμετρία ελέγχου κίνησης.

Kinematic Position Control — Polar Coordinates Kinematics -1

Κινηματικό Σύστημα στις Πολικές Συντεταγμένες

- Στο νέο σύστημα (ρ, α, β), οι εξισώσεις κίνησης είναι:

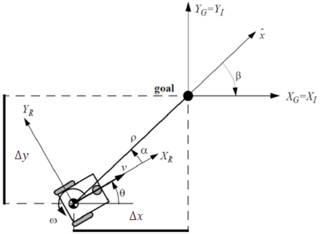
$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos\alpha & 0 \\ \frac{\sin\alpha}{\rho} & -1 \\ -\frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \text{for } \alpha \in I_1 = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

- Καθώς και:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ -\frac{\sin\alpha}{\rho} & 1 \\ \frac{\sin\alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \text{for } \alpha \in I_2 = \left[-\pi, -\frac{\pi}{2}\right] \cup \left[\frac{\pi}{2}, \pi\right]$$

Kinematic Position Control — Polar Coordinates Kinematics -2

- Ο μετασχηματισμός δεν ορίζεται στο στόχο, i.e. $x = y = 0$. $\Rightarrow$ The controller must **stop very close to the goal**, αλλιώς προκύπτει singularity: Stop when $\rho < \epsilon$, with ϵ small enough.
- For $\alpha \in I_1 = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ the **forward direction** of the robot points toward the goal. And for $\alpha \in I_2 = \left[-\pi, -\frac{\pi}{2}\right] \cup \left[\frac{\pi}{2}, \pi\right]$, the **backward direction** points toward the goal.
- By properly defining the **forward direction at the initial configuration**, it is always possible to have $\alpha \in I_1$ at $t = 0$. However, α **may leave I_1 during the motion**.



Γεωμετρία ελέγχου κίνησης.

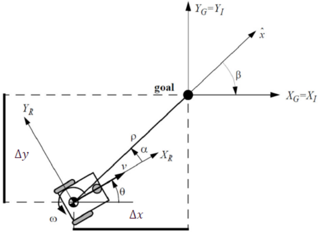
Kinematic Position Control — The Control Law

Feedback control law

$$v = k_\rho\, \rho, \qquad \omega = k_\alpha\, \alpha + k_\beta\, \beta$$

$$\Rightarrow \quad \begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\, k_\rho\, \rho \cos\alpha \\ -\, k_\rho \sin\alpha \;-\; k_\alpha\, \alpha \;-\; k_\beta\, \beta \\ -\, k_\rho \sin\alpha \end{bmatrix}$$

- The closed-loop system drives the robot to $(\rho, \alpha, \beta) = (0, 0, 0)$.
- The translational command v keeps a **constant sign**:
  - the direction of motion stays *either* forward *or* backward,
  - parking is executed in a natural way without ever reversing.



Γεωμετρία ελέγχου κίνησης.

Kinematic Position Control — Control Gains

Control Law

$$v = k_\rho \, \rho, \qquad \omega = k_\alpha \, \alpha + k_\beta \, \beta$$

- k_ρ – *Translational gain*
  - Ρυθμίζει πόσο γρήγορα πλησιάζει το ρομπότ τον στόχο.
  - Μεγάλο k_ρ: ταχεία προσέγγιση, κίνδυνος ταλαντώσεων.
  - Μικρό k_ρ: αργή κίνηση.
- k_α – *Orientation-to-goal gain*
  - Ελέγχει πόσο δυνατά το ρομπότ «στρίβει» για να ευθυγραμμιστεί προς τον στόχο.
  - Μεγάλο k_α: ταχεία περιστροφή, πιθανές ταλαντώσεις.
  - Μικρό k_α: αργή ευθυγράμμιση.
- k_β – *Final orientation gain*
  - Ελέγχει τη γωνία β, δηλ. την τελική ευθυγράμμιση με τον στόχο.
  - Επιλέγεται **αρνητικό** για ευστάθεια.

Θεμελιώδεις Συνθήκες Ευστάθειας

Συνθήκες στα κέρδη (Samson 1990)

Για τον νόμο ελέγχου:

$$v = k_\rho \rho, \qquad \omega = k_\alpha \alpha + k_\beta \beta$$

οι παράμετροι πρέπει να ικανοποιούν:

$$\boxed{k_\rho > 0}, \qquad \boxed{k_\beta < 0}, \qquad \boxed{k_\alpha - k_\rho > 0}.$$

- k_ρ: θετικό, ώστε η ταχύτητα να κατευθύνεται προς τον στόχο.
- k_β: αρνητικό, για σταθεροποίηση της τελικής γωνίας προσανατολισμού.
- $k_\alpha - k_\rho > 0$: διασφαλίζει την αρνητική οριστικότητα της $\dot{V}$.

Ανάλυση Lyapunov

Συνάρτηση Lyapunov

$$V(\rho, \alpha, \beta) = \frac{1}{2}\rho^2 + \frac{1}{2}\left(\alpha^2 + \beta^2\right)$$

είναι θετικά ορισμένη.

Χρονική Παράγωγος

Με χρήση των κινηματικών εξισώσεων και του νόμου ελέγχου:

$$\dot{V} = -k_\rho \rho^2 \cos^2\alpha - k_\alpha \alpha^2 - |k|\beta^2 \leq 0.$$

- Η $\dot{V}$ είναι **αρνητικά ορισμένη** υπό τις παραπάνω συνθήκες.
- Η προέλευση $(\rho, \alpha, \beta) = (0, 0, 0)$ είναι **παγκόσμια ασυμπτωτικά ευσταθής**.

Περιγραφή Λειτουργίας Κώδικα

Στόχος

Να μετακινηθεί ένα ρομπότ τύπου **unicycle** από μια αρχική θέση προς το σημείο στόχο $(0,0)$, εφαρμόζοντας τον νόμο ελέγχου του Samson (1990).

- Ορίζονται η **αρχική θέση** (x, y, θ) και ο **στόχος** $(0,0)$.
- Σε κάθε βήμα υπολογίζονται τα ρ, α, β (σφάλματα σε πολικές συντεταγμένες).
- Υπολογίζονται οι ταχύτητες v και ω με βάση:

$$v = k_\rho \rho, \qquad \omega = k_\alpha \alpha + k_\beta \beta.$$

- Ενημερώνεται η θέση του ρομπότ μέσω των κινηματικών εξισώσεων του unicycle.
- Η διαδικασία επαναλαμβάνεται μέχρι το ρομπότ να φτάσει κοντά στον στόχο.
- Τέλος, γίνεται **σχεδίαση της τροχιάς** του ρομπότ σε γράφημα.