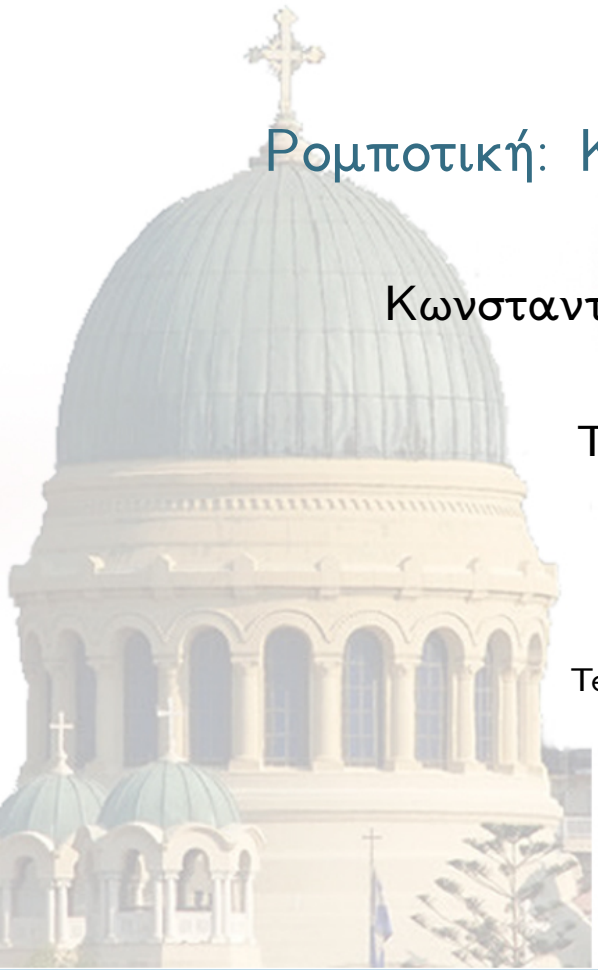


# Τεχνολογίες Ευφυών Συστημάτων και Ρομποτική

## Ρομποτική: Κινηματική Ανάλυση και Ιακωβιανοί Πίνακες

Κωνσταντίνος Χατζηλυγερούδης - costashatz@upatras.gr

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

2 Μαρτίου 2023

Template made by **Παναγιώτης Παπαγιαννόπουλος**

# Υποχρεώσεις-Αξιολόγηση

- **Μία εργασία/project (για το κομμάτι της Ρομποτικής)**
  - Μία εργασία πάνω στη δημιουργία ενός ελεγκτή για ένα ρομποτικό σύστημα
  - Παραδίδονται κώδικας και αναφορά
  - Αξιολόγηση των εργασιών βάσει των παραδοτέων και **προφορικής εξέτασης**
  - Ατομικές ή ομαδικές μέχρι δύο (2) άτομα
  - Η εργασία πιάνει το 50% του συνολικού βαθμού

# Περιεχόμενα - Θεωρία

- **Ρομποτική (Robotics)**
  - Κίνηση Στερεών Σωμάτων σε 3Δ χώρο
  - Κινηματική Ανάλυση Ρομποτικών Συστημάτων
  - Δυναμική Ανάλυση Ρομποτικών Συστημάτων
  - Δημιουργία Ελεγκτών
  - Έλεγχος σε Καρτεσιανό Χώρο (ή Χώρο Εργασίας)
  - Behavior Trees για έλεγχο ρομποτικών συστημάτων
  - Optimal Control/Trajectory Optimization

# Συγγράμματα

**Προτεινόμενα**:

- **Modern Robotics: Mechanics, Planning, and Control**, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press. ebook

**Από Εύδοξο**:

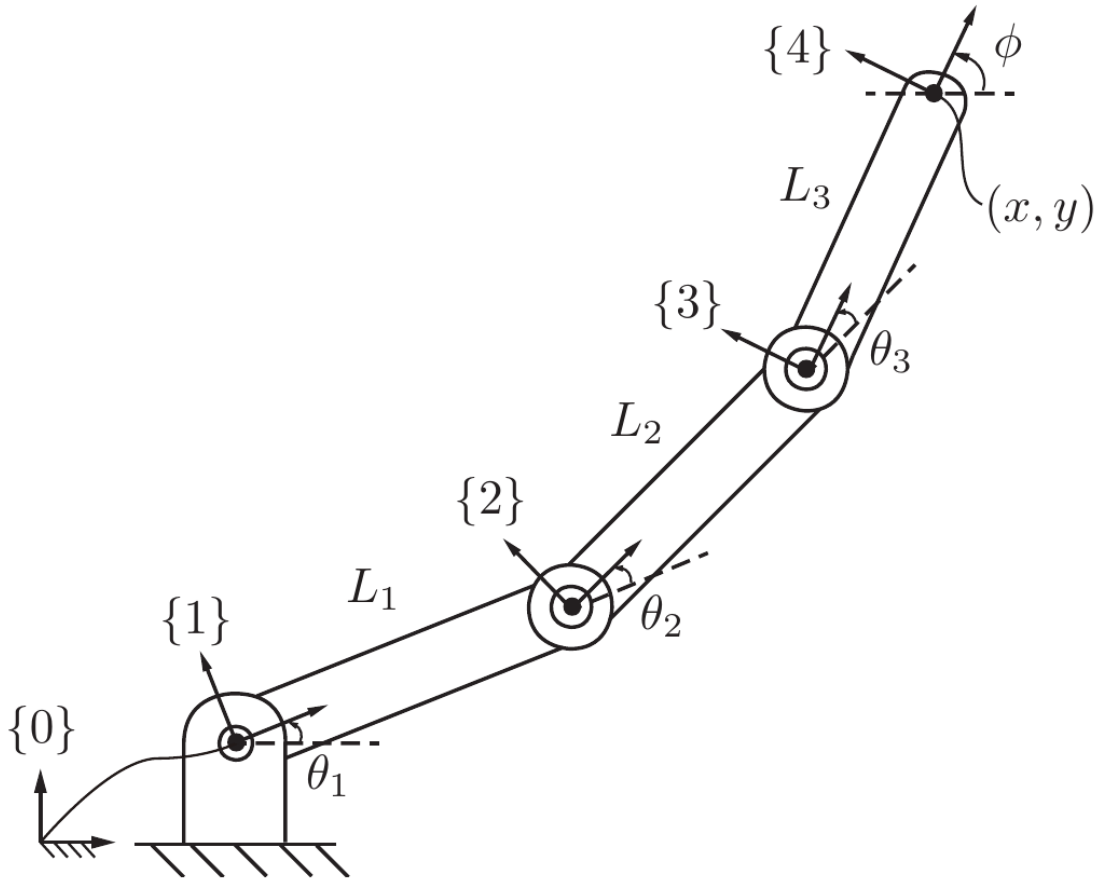
- **Εισαγωγή στη Ρομποτική**, *J. Craig*, 2020, Τζιόλα
- **Πιθανοτική Ρομποτική**, *S. Thrun, W. Burgard, D. Fox*, 2005, Κλειδάριθμος
- **Ρομποτική**, *Δ. Εμίρης, Δ. Κουλουριώτης*, 2020, Τζιόλα

# End-effectors

# End-effectors (2)

# Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics) (1)

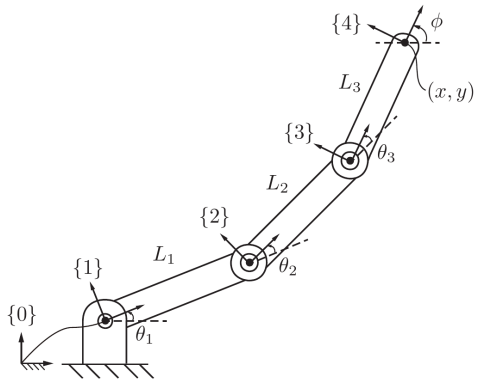
End-effector




**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

# Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics) (2)

End-effector

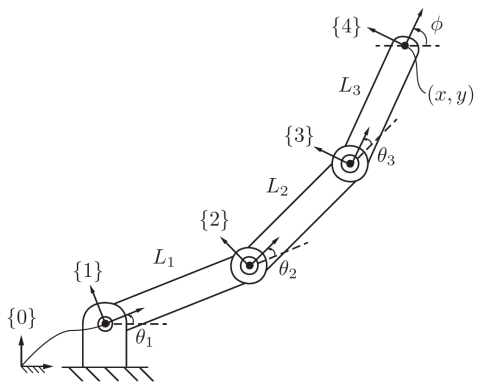



**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

**Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics)**: να βρούμε $(x, y, \phi)$ όταν γνωρίζουμε $\theta_1, \theta_2, \theta_3$.

# Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics) (2)

End-effector

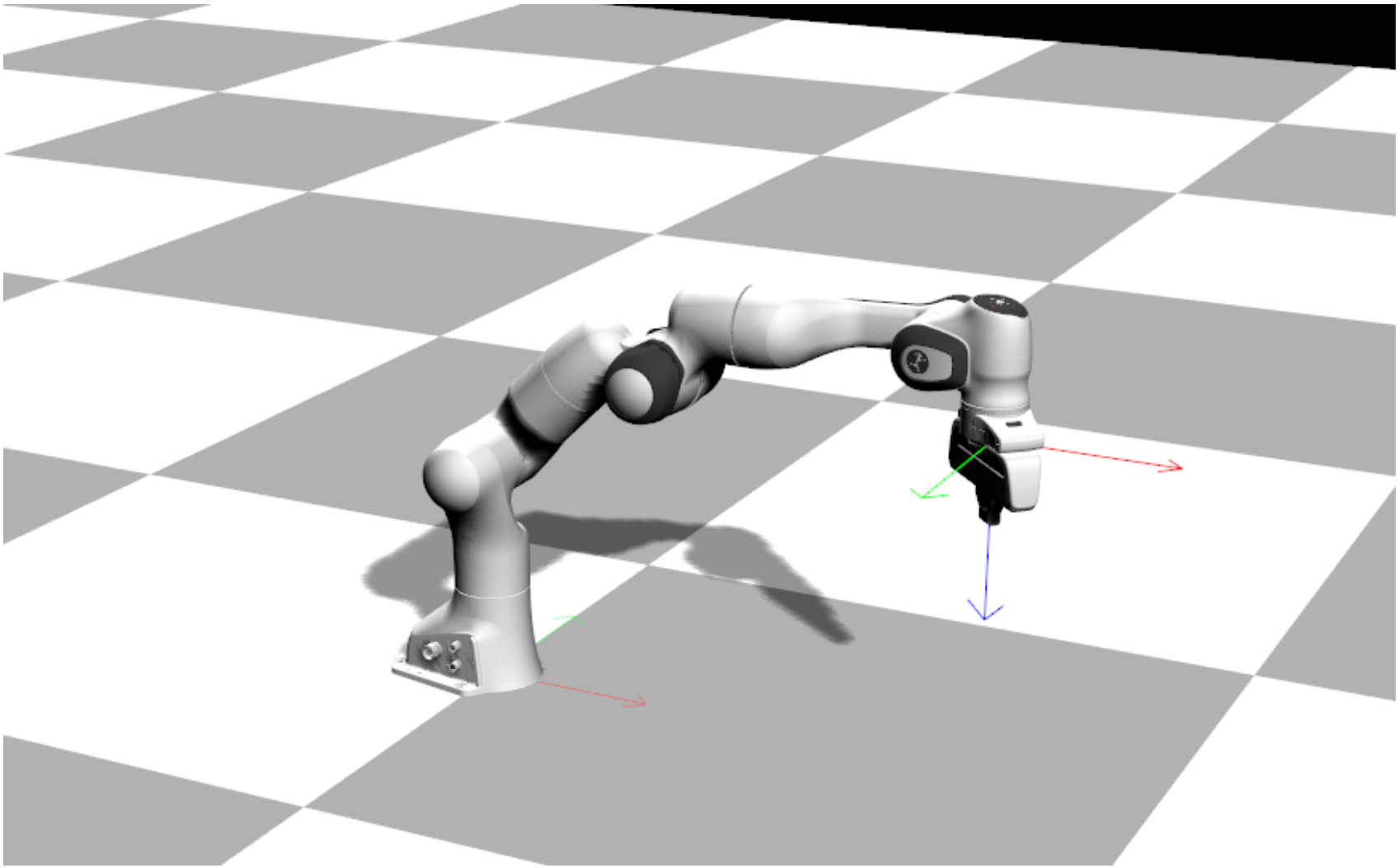



**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

**Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics)**: να βρούμε $(x, y, \phi)$ όταν γνωρίζουμε $\theta_1, \theta_2, \theta_3$.

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2) + L_3\cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1 + \theta_2) + L_3\sin(\theta_1 + \theta_2 + \theta_3)$$
$$\phi = \theta_1 + \theta_2 + \theta_3$$

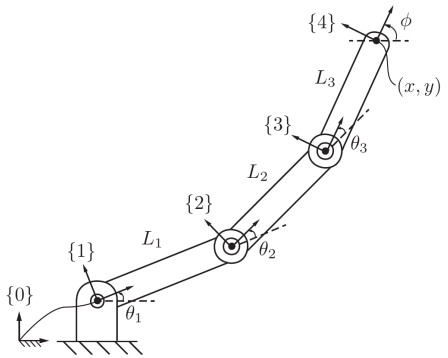
# Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics) (3)



# Ευθύ Κινηματικό Πρόβλημα (Forward Kinematics) (4)

Μπορούμε να μοντελοποιήσουμε το πρόβλημα με μητρώα μετασχηματισμού:

End-effector




**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

$$(x, y, \phi) \equiv T_{04} = T_{01}\, T_{12}\, T_{23}\, T_{34}$$

όπου

$$T_{01} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{12} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_1 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{23} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{34} = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Πώς περιγράφουμε τα δικά μας ρομπότ;

- Με μητρώα μετασχηματισμού
  - Πολλοί παράμετροι
  - Χρειαζόμαστε body frames για κάθε σώμα
- Παράμετροι Denavit–Hartenberg (DH parameters)
  - Βέλτιστος αριθμών παραμέτρων
  - Χρειαζόμαστε body frames για κάθε σώμα
  - "Ταυτόσημος" με τα μητρώα μετασχηματισμού αλλά αποθηκεύει τις λιγότερες δυνατές παραμέτρους
- Product of Exponentials (PoE)
  - Μη βέλτιστος αριθμών παραμέτρων, αλλά λιγότεροι από μητρώα μετασχηματισμού
  - ΔΕΝ χρειαζόμαστε body frames για κάθε σώμα

# Universal Robot Description Format (1)

**Τα αρχεία URDF είναι ένας XML τύπος αρχείων για να περιγράφουμε ρομπότ:**

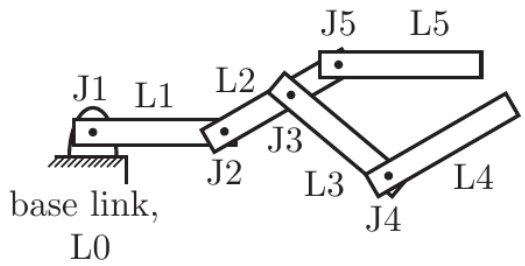
- Ξεκίνησε ως κομμάτι του ROS, αλλά πλέον υπάρχει και αυτόνομα
- Επιτρέπει τη μοντελοποίηση κινηματικών, δυναμικών και γεωμετρικών στοιχείων των ρομπότ
- Επιτρέπει τη μοντελοποίηση ρομποτικών συστημάτων που σχηματίζουν κινηματικά δέντρα (open-chain structures)
- Τα περισσότερα λογισμικά προσομοίωσης μπορούν να τα διαβάσουν

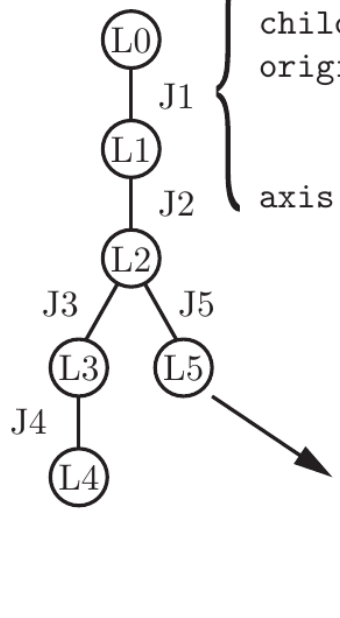
# Universal Robot Description Format (2)

Κάθε URDF αρχείο ορίζει αρθρώσεις (joints) και σώματα (links):

- **Joints**:
  - Συνδέουν δύο (ακριβώς) links: το parent link με το child link
  - Δυνατοί τύποι: revolute, prismatic και fixed
  - origin frame: περιγράφει τη θέση του child link ως προς το parent link όταν το joint είναι στην αρχική (ή μηδενική) θέση
  - axis (3Δ διάνυσμα): περιγράφει τον άξονα κίνησης του joint
  - ...
- **Links**:
  - Intertial parameters: μάζα, κέντρο βάρους, ...(σημαντικό για προσομοίωση)
  - Γεωμετρία για visualization
  - Γεωμετρία για collision detection (σημαντικό για προσομοίωση)
  - ...

# Universal Robot Description Format (3)




`parent:` J1's parent link, L0
`child:` J1's child link, L1
`origin:` the $x$–$y$–$z$ and roll–pitch–yaw coords of the L1 frame relative to the L0 frame when J1 is zero
`axis:` the $x$–$y$–$z$ unit vector along the rotation axis in the L1 frame

`mass:` L5's mass
`origin:` the $x$–$y$–$z$ and roll–pitch–yaw coords of a frame at the center of mass of L5, relative to the L5 frame
`inertia:` six unique entries of inertia matrix in the `origin` frame

**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

# Ιακωβιανοί Πίνακες (Jacobians) (1)

Ας υποθέσουμε ότι ο end-effector του ρομπότ μας κινείται με ταχύτητα $\dot{x}$. Σε αυτή την περίπτωση το ευθύ κινηματικό πρόβλημα γράφεται ως εξής:

$$x(t) = f(\theta(t))$$

όπου $f$ η "συνάρτηση" που μας δίνει τα forward kinematics, $x \in \mathbb{R}^m$ η θέση του end-effector, και $\theta \in \mathbb{R}^n$ οι θέσεις των αρθρώσεων του ρομπότ. Αν πάρουμε την παράγωγο, έχουμε:

$$\begin{aligned} \dot{x} &= \frac{\partial f(\theta)}{\partial \theta} \frac{\partial \theta(t)}{\partial t} \\ &= J(\theta)\dot{\theta} \end{aligned}$$

όπου ο $J(\theta) \in \mathbb{R}^{m \times n}$ ορίζεται ως ο Ιακωβιανός (Jacobian).

# Ιακωβιανοί Πίνακες (Jacobians) (2)

- Έστω ότι έχουμε το twist $\mathcal{V}_w$ του end-effector ως προς το world frame
- Ισχύει ότι: $\mathcal{V}_w = J_w(\theta)\dot{\theta}$, όπου $J_w(\theta) \in \mathbb{R}^{6\times n}$
- Αντίστοιχα έχουμε $\mathcal{V}_b = J_b(\theta)\dot{\theta}$, όπου $J_b(\theta) \in \mathbb{R}^{6\times n}$ για το body frame

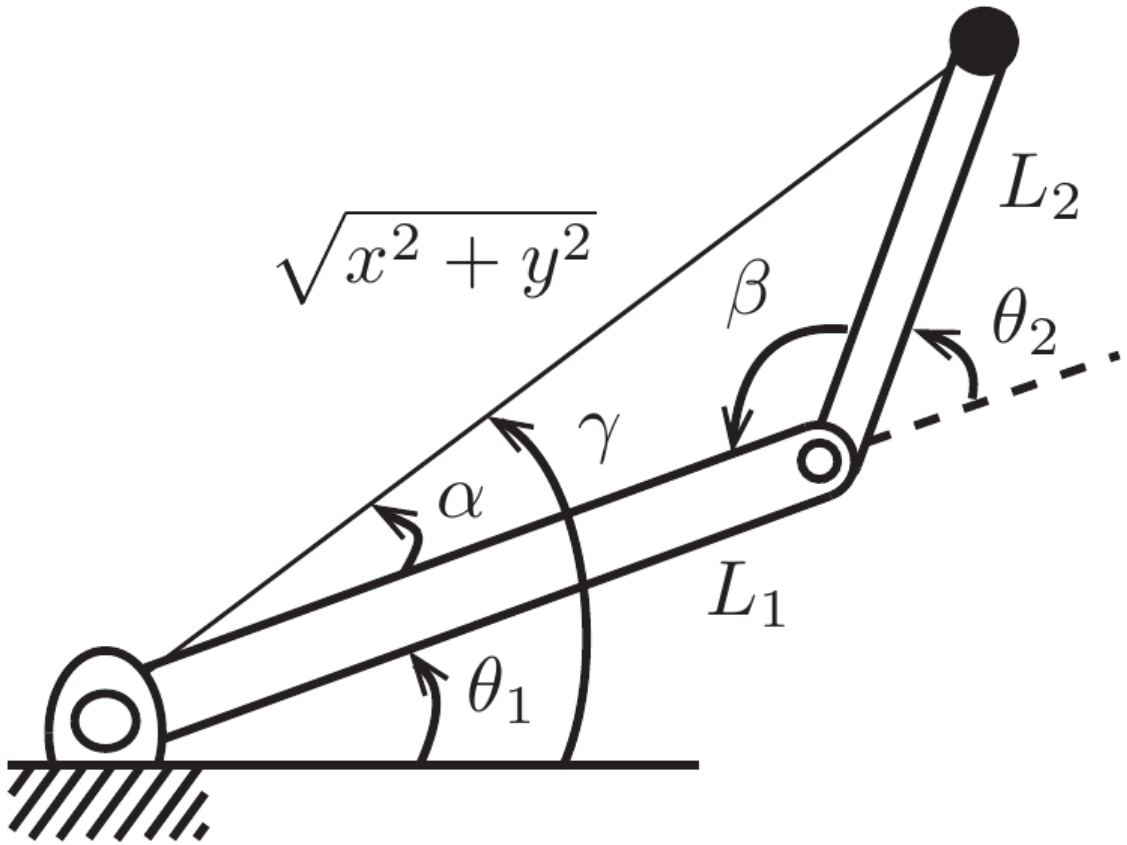
Από την αρχή διατήρησης της ενέργειας, μπορούμε να εξάγουμε σχέση και για τα wrenches:

$$\tau = J_w(\theta)^T \mathcal{F}_w$$
$$\tau = J_b(\theta)^T \mathcal{F}_b$$

όπου $\tau \in \mathbb{R}^n$ είναι οι ροπές (torques) των αρθρώσεων.

# Αντίστροφο Κινηματικό Πρόβλημα (1)

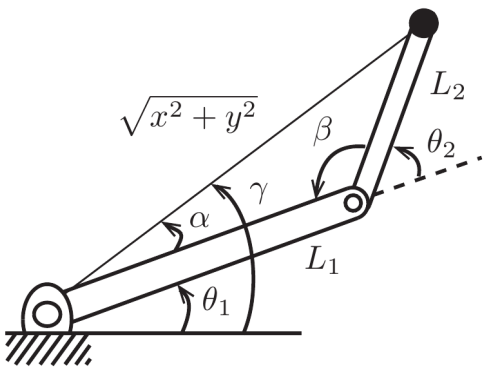
End-effector




**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

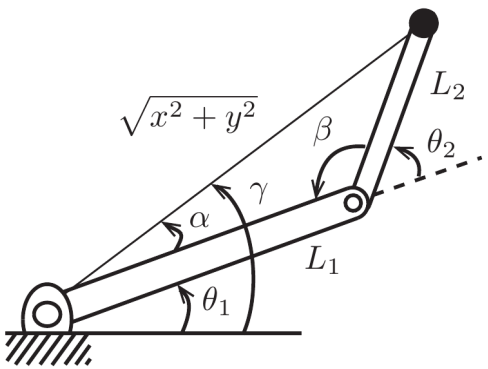
# Αντίστροφο Κινηματικό Πρόβλημα (2)

**Αντίστροφο Κινηματικό Πρόβλημα (Inverse Kinematics)**: να βρούμε τα $\theta_1, \theta_2$ όταν γνωρίζουμε τα $(x, y)$.




**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

# Αντίστροφο Κινηματικό Πρόβλημα (2)

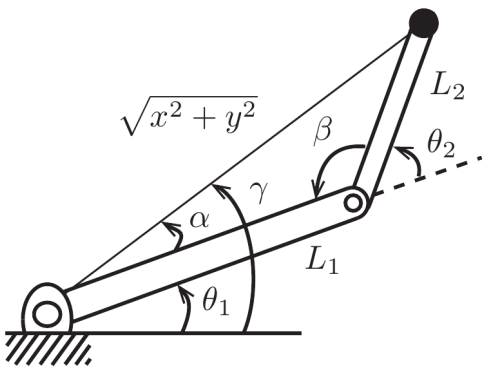



**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

**Αντίστροφο Κινηματικό Πρόβλημα (Inverse Kinematics)**: να βρούμε τα $\theta_1, \theta_2$ όταν γνωρίζουμε τα $(x, y)$. **Ξεκινάμε με Forward Kinematics:**

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2)$$
$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1 + \theta_2)$$

# Αντίστροφο Κινηματικό Πρόβλημα (2)




**Πηγή**: Modern Robotics: Mechanics, Planning, and Control, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press.

**Αντίστροφο Κινηματικό Πρόβλημα (Inverse Kinematics)**: να βρούμε τα $\theta_1, \theta_2$ όταν γνωρίζουμε τα $(x, y)$. **Ξεκινάμε με Forward Kinematics:**

$$x = L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2)$$
$$y = L_1\sin(\theta_1) + L_2\sin(\theta_1 + \theta_2)$$

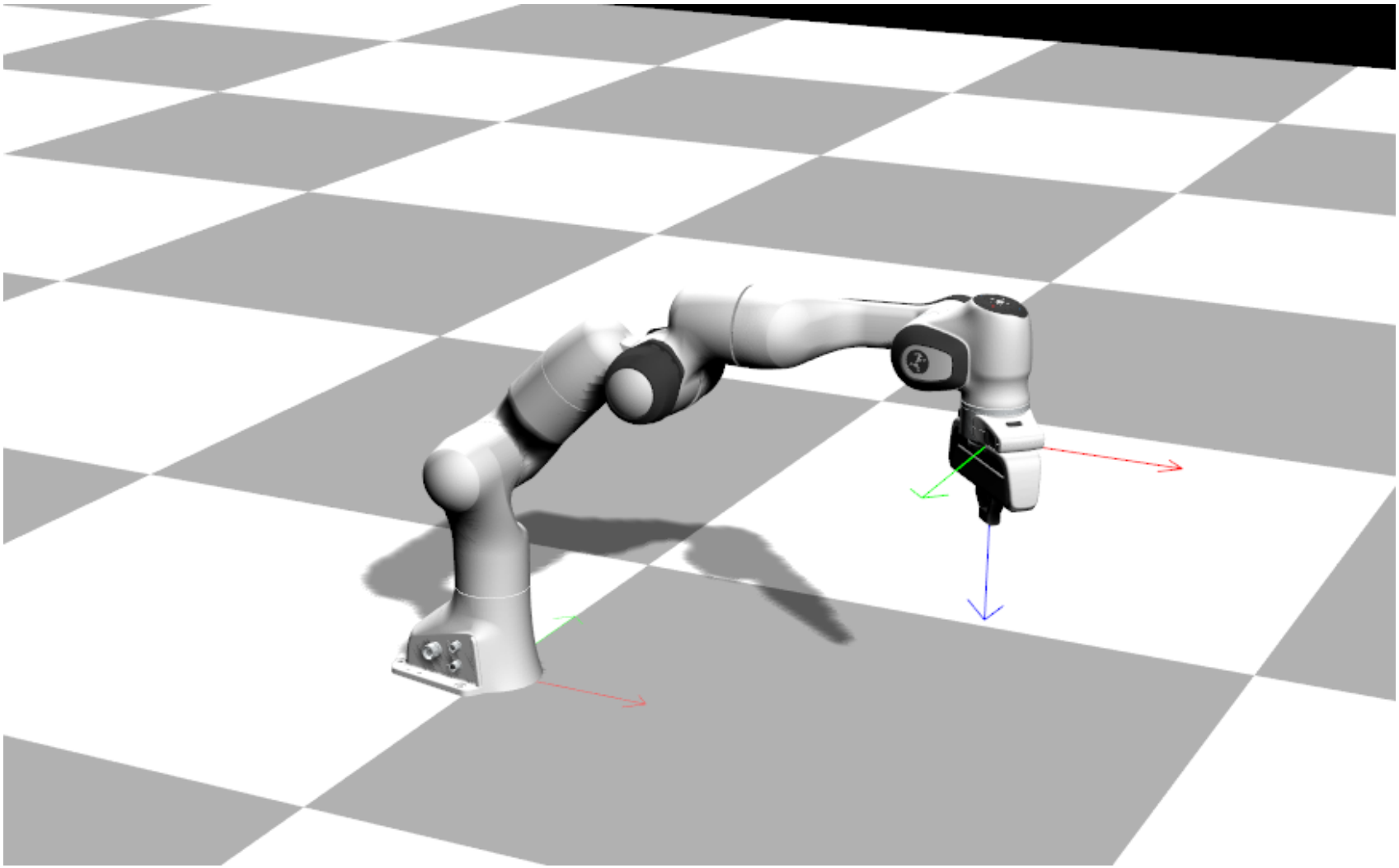
**Έπειτα από αρκετή τριγωνομετρία έχουμε:**

$$\theta_1 = \gamma \pm \alpha$$
$$\theta_2 = \pm(\pi - \beta)$$

με $\beta = \cos^{-1}\left(\frac{L_1^2+L_2^2-x^2-y^2}{2L_1L_2}\right)$, $\alpha = \cos^{-1}\left(\frac{x^2+y^2+L_1^2-L_2^2}{2L_1\sqrt{x^2+y^2}}\right)$

και $\gamma = \text{atan2}(y, x)$.

# Αντίστροφο Κινηματικό Πρόβλημα (3)



# Inverse Kinematics ως Βελτιστοποίηση (1)

- Έχουμε τη θέση του end-effector, $x \in \mathbb{R}^m$, η οποία δίνεται από τα forward kinematics $x = f(\theta)$, όπου $\theta \in \mathbb{R}^n$ είναι οι τιμές των αρθρώσεων του ρομπότ ($n$ βαθμοί ελευθερίας)
- Έστω ότι θέλουμε να φτάσουμε στη θέση $x_d$
- Ορίζουμε την συνάρτηση $g(\theta) = x_d - f(\theta)$
- Ψάχνουμε τις παραμέτρους $\theta_d$ έτσι ώστε $g(\theta_d) = x_d - f(\theta_d) = 0$
- Έχουμε (Taylor Expansion):

$$x_d - f(\theta_d) = 0$$

$$x_d = f(\theta_d) = f(\theta_0) + \frac{\partial f}{\partial \theta}\Big|_{\theta=\theta_0} (\theta_d - \theta_0) + \ldots$$

# Inverse Kinematics ως Βελτιστοποίηση (2)

**Τι μας θυμίζει το $\frac{\partial f}{\partial \theta}$;**

# Inverse Kinematics ως Βελτιστοποίηση (2)

**Τι μας θυμίζει το $\frac{\partial f}{\partial \theta}$;**

$$x_d = f(\theta_d) = f(\theta_0) + \frac{\partial f}{\partial \theta}\Big|_{\theta=\theta_0}(\theta_d - \theta_0) + \ldots$$
$$= f(\theta_0) + J(\theta_0)\Delta\theta + \ldots$$
$$J(\theta_0)\Delta\theta = x_d - f(\theta_0)$$

**Αν θεωρήσουμε ότι ο $J$ είναι αντιστρέψιμος:**

$$\Delta\theta = J^{-1}(\theta_0)(x_d - f(\theta_0))$$

**Επαναληπτικός Αλγόριθμος - Newton-Raphson:**

1. $\theta_i = \theta_0, i = 0$
2. $\Delta\theta = J^{-1}(\theta_i)(x_d - f(\theta_i))$
3. $\theta_{i+1} = \theta_i + \Delta\theta$
4. Αν $x_d - f(\theta_{i+1}) \approx 0$, τέλος, αλλιώς $i = i + 1$ και πίσω στο βήμα 2

# Inverse Kinematics ως Βελτιστοποίηση (3)

**Αν ο $J$ δεν είναι αντιστρέψιμος;**

# Inverse Kinematics ως Βελτιστοποίηση (3)

**Αν ο $J$ δεν είναι αντιστρέψιμος;**
**Κανένα πρόβλημα! Μπορούμε να χρησιμοποιήσουμε τον Moore–Penrose ψευδο-αντίστροφο (pseudoinverse)**:

$$J^{\dagger} = J^T(JJ^T)^{-1}, \text{ αν } n > m, (J^{\dagger}J = I)$$
$$J^{\dagger} = (J^TJ)^{-1}J^T, \text{ αν } n < m, (JJ^{\dagger} = I)$$

Και άρα έχουμε:

$$\Delta\theta = J^{\dagger}(\theta_0)(x_d - f(\theta_0))$$

# Inverse Kinematics ως Βελτιστοποίηση (4)

**Αν έχουμε ένα μητρώο μετασχηματισμού για την θέση/προσανατολισμό του end-effector;**
Έστω $T_{wd}$ η θέση/προσανατολισμός στόχος, έχουμε τον εξής αλγόριθμο:

1. $\theta_i = \theta_0, i = 0$
2. $[\mathcal{V}_b] = \log(T_{wb}^{-1}(\theta_i) T_{wd})$, $T_{wb}(\theta)$ δίνει τα forward kinematics
3. $\Delta\theta = J_b^{\dagger} \mathcal{V}_b$, $J_b$ είναι το body jacobian
4. $\theta_{i+1} = \theta_i + \Delta\theta$
5. Αν $||\mathcal{V}_b|| \approx 0$, τέλος, αλλιώς $i = i + 1$ και πίσω στο βήμα 2

Αντίστοιχα μπορούμε να κάνουμε τους υπολογισμούς στο world frame, με $[\mathcal{V}_w] = [Ad_{T_{wb}}]\mathcal{V}_b$ και χρησιμοποιώντας τον world jacobian, $J_w$.

# Inverse Kinematics ως Βελτιστοποίηση (5)

**Η μέθοδος με τον ψευδο-αντίστροφο έχει μερικά προβλήματα:**

- Πρέπει το $\theta_0$ να είναι κοντά σε καλή τιμή
- "Σπάει" όταν είμαστε σε singularities
  - όταν χάνει βαθμό/rank ο Ιακωβιανός πίνακας
  - όταν δηλαδή χάνουμε βαθμούς ελευθερίας στον end-effector
- Οι τελικές τιμές $\theta$ μπορεί να μην υπακούουν τα όρια του ρομπότ

**Εναλλακτική μέθοδος:**

- Ελαχιστοποίηση του λάθους
- $[\mathcal{V}_b] = \log(T_{wb}^{-1}(\theta_i)\, T_{wd})$ μας δίνει το σφάλμα!
- Θέλουμε να γίνει όσο το δυνατόν μικρότερο
- Άρα ελαχιστοποιούμε το $||\mathcal{V}_b||$ με όποιον αλγόριθμο βελτιστοποίησης θέλουμε
- Μπορούμε να βάλουμε όρια στην βελτιστοποίηση πιο εύκολα

# Βιβλιογραφία

Κεφάλαια 3, 4 και 5 από **Modern Robotics: Mechanics, Planning, and Control**, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press. ebook