Τεχνολογίες Ευφυών Συστημάτων και Ρομποτική

Ρομποτική: Δυναμική Ανάλυση και Ελεγκτές

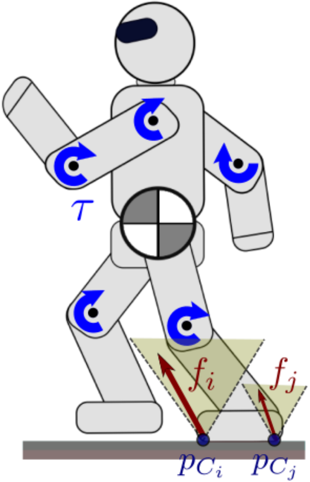
Κωνσταντίνος Χατζηλυγερούδης - costashatz@upatras.gr

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

13 Μαρτίου 2023

Template made by **Παναγιώτης Παπαγιαννόπουλος**

Οι δυνάμεις που είναι;



Δυναμική Ανάλυση Ρομποτικών Συστημάτων (1)

$$M(\theta)\ddot{\theta} + C_g(\theta, \dot{\theta}) = \tau + J^T(\theta)\mathcal{F}$$
$$\text{όπου } C_g(\theta, \dot{\theta}) = C(\theta, \dot{\theta}) + g(\theta)$$

όπου M είναι το λεγόμενο mass matrix, C_g είναι το άθροισμα των βαρυτικών ($g(\theta)$), φυγοκεντρικών και Coriolis δυνάμεων, τ είναι ροπές που βάζουμε ή δέχονται οι αρθρώσεις, J είναι όλοι οι Ιακωβιανοί πίνακες (ένας για κάθε επαφή/contact), και $\mathcal{F}$ είναι όλες οι δυνάμεις που ασκούνται στις επαφές.

Δυναμική Ανάλυση Ρομποτικών Συστημάτων (2)

Όταν δεν έχουμε επαφές και πολύ γρήγορες κινήσεις, μπορούμε να γράφουμε:

$$M(\theta)\ddot{\theta} + C_g(\theta, \dot{\theta}) = \tau$$

Όταν δεν έχουμε επαφές και πολύ γρήγορες κινήσεις, μπορούμε να γράψουμε:

$$M(\theta)\ddot{\theta} + C_g(\theta, \dot{\theta}) = \tau$$

Ευθύ Δυναμικό Πρόβλημα (Forward Dynamics) είναι να βρούμε τα $\ddot{\theta}$ αν γνωρίζουμε τα $(\theta, \dot{\theta}, \tau)$:

$$\ddot{\theta} = M^{-1}(\theta)(\tau - C_g(\theta, \dot{\theta}))$$

Όταν δεν έχουμε επαφές και πολύ γρήγορες κινήσεις, μπορούμε να γράφουμε:

$$M(\theta)\ddot{\theta} + C_g(\theta, \dot{\theta}) = \tau$$

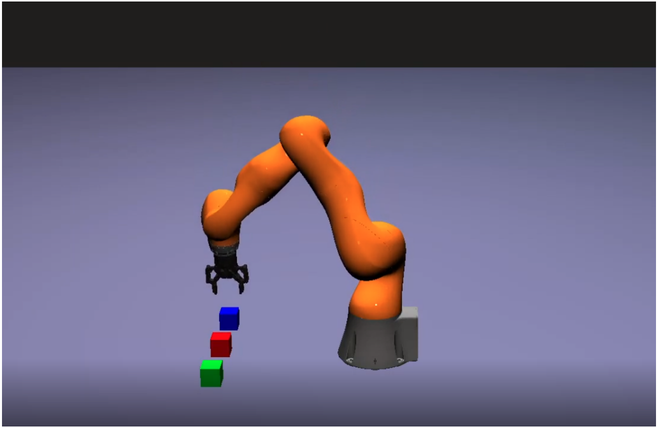
Ευθύ Δυναμικό Πρόβλημα (Forward Dynamics) είναι να βρούμε τα $\ddot{\theta}$ αν γνωρίζουμε τα $(\theta, \dot{\theta}, \tau)$:

$$\ddot{\theta} = M^{-1}(\theta)(\tau - C_g(\theta, \dot{\theta}))$$

Αντίστροφο Δυναμικό Πρόβλημα (Inverse Dynamics) είναι να βρούμε τα τ αν γνωρίζουμε τα $(\theta, \dot{\theta}, \ddot{\theta})$:

$$\tau = M(\theta)\ddot{\theta} + C_g(\theta, \dot{\theta})$$

Ας κινήσουμε κάτι!



Έλεγχος μίας άρθρωσης με ταχύτητες (1)

- Έστω ότι έχουμε ορίσει μία τροχιά $\theta_d(t)$
- Για παράδειγμα, $\theta_d(t) = \alpha \sin(t)$
- Ο πιο εύκολος τρόπος είναι να δίνουμε στην άρθρωση ταχύτητες στόχους, π.χ.:
$$\dot{\theta}(t) = \dot{\theta}_d(t) = \alpha \cos(t)$$
- **Feedforward** ή **open-loop** ελεγκτής (controller)
- ΔΕΝ υπάρχει καμία ανατροφοδότηση από αισθητήρες

Έλεγχος μίας άρθρωσης με ταχύτητες (2)

- Πώς μπορούμε να εκμεταλλευτούμε τους αισθητήρες; Να έχουμε δηλαδή έλεγχο με ανατροφοδότηση (feedback control);

Έλεγχος μίας άρθρωσης με ταχύτητες (2)

- Πώς μπορούμε να εκμεταλλευτούμε τους αισθητήρες; Να έχουμε δηλαδή έλεγχο με ανατροφοδότηση (feedback control);
- **P-controller:** $\dot{\theta}(t) = K_p(\theta_d(t) - \theta(t)),\ K_p > 0$
- $\theta_e(t) = (\theta_d(t) - \theta(t))$
- **PI-controller:** $\dot{\theta}(t) = K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt,\ K_p, K_i > 0$
- Αν $\dot{\theta}_d(t) =$ constant, τότε ο PI-controller αφαιρεί το σφάλμα σταθερής κατάστασης

Έλεγχος μίας άρθρωσης με ταχύτητες (3)

- Ο έλεγχος με ανατροφοδότηση χρειάζεται σφάλμα για να "ξεκινήσει"

Έλεγχος μίας άρθρωσης με ταχύτητες (3)

- Ο έλεγχος με ανατροφοδότηση χρειάζεται σφάλμα για να "ξεκινήσει"
- Ας συνδυάσουμε open-loop και feedback control:
  $$\dot{\theta}(t) = \dot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt, K_p, K_i > 0$$
- **Feedforward Plus Feedback Control**
- Ο επιθυμητός τρόπος ελέγχου

Τι κάνουμε όταν έχουμε πολλές αρθρώσεις;

Έλεγχος πολλών αρθρώσεων με ταχύτητες

Τι κάνουμε όταν έχουμε πολλές αρθρώσεις;

- Κάθε άρθρωση είναι "ανεξάρτητη"
- Άρα έχουμε n ελεγκτές (όπου n οι βαθμοί ελευθερίας)
- $K_p \in \mathbb{R}^{n \times n}$ ή $K_p \in \mathbb{R}^n$
- $K_i \in \mathbb{R}^{n \times n}$ ή $K_i \in \mathbb{R}^n$
- K_p, K_i έχουν τιμές μόνο στις διαγώνιους ή είναι διανύσματα

Έλεγχος με ταχύτητες σε Task-Space

Αν γνωρίζουμε το προφίλ ταχυτήτων, $\mathcal{V}_d(t)$, **του end-effector;**

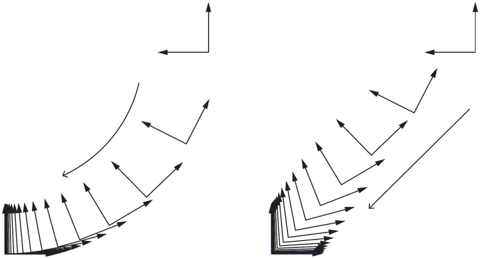
Αν γνωρίζουμε το προφίλ ταχυτήτων, $\mathcal{V}_d(t)$, **του end-effector;**

- $\mathcal{V}_b(t) = [Ad_{T_{wb}}]\mathcal{V}_d(t) + K_p X_e(t) + K_i \int_0^t X_e(t)dt, K_p, K_i > 0$
- $[X_e] = \log(T_{wb}^{-1} T_{wd})$, το σφάλμα στο χώρο των θέσεων/προσανατολισμών
- $\mathcal{V}_b = J_b\dot{\theta} \implies \dot{\theta}(t) = J_b^{\dagger}(\theta(t))\mathcal{V}_b(t)$
- $\dot{\theta} = J_b^{\dagger}\mathcal{V}_b$

Ο ψευδο-αντίστροφος μπορεί να είναι ασταθής γύρω από singularities, στην πράξη χρησιμοποιούμε:

- $\dot{\theta} = \alpha J^T \mathcal{V}$
- Damped Pseudoinverse:
  - $J^\dagger = J^T(JJ^T + \lambda^2 I)^{-1}$, αν $n > m, (J^\dagger J = I)$
  - $J^\dagger = (J^T J + \lambda^2 I)^{-1} J^T$, αν $n < m, (JJ^\dagger = I)$
  - $\lambda \in \mathbb{R}, \lambda > 0$

Διαχωρισμός προσανατολισμού και θέσης (1)



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

Διαχωρισμός προσανατολισμού και θέσης (2)

Συνήθως διαχωρίζουμε τον έλεγχο του προσανατολισμού και της θέσης. Άρα έχουμε:

- $\mathcal{V}_d(t) = [\omega_d, \dot{p_d}]^T$
- $\mathcal{V}_b(t) = \begin{bmatrix} R_{wb}^{-1} R_{wd} & 0 \\ 0 & I \end{bmatrix} \mathcal{V}_d(t) + K_p X_e(t) + K_i \int_0^t X_e(t) dt$
- $X_e = [\log(R_{wb}^{-1} R_{wd}), t_{wd} - t_{wb}]$, το σφάλμα στο χώρο των θέσεων/προσανατολισμών
- $\mathcal{V}_b = J_b \dot{\theta} \implies \dot{\theta}(t) = J_b^{\dagger}(\theta(t))\mathcal{V}_b(t)$
- $\dot{\theta} = J_b^{\dagger} \mathcal{V}_b$

Έλεγχος με ροπές/δυνάμεις

Τι κάνουμε όταν θέλουμε να ελέγχουμε τις ροπές, τ, στις αρθρώσεις;

Τι κάνουμε όταν θέλουμε να ελέγχουμε τις ροπές, τ, στις αρθρώσεις;

- **Feedforward/Open-loop control:**

$$\tau(t) = M(\theta_d(t))\ddot{\theta}_d(t) + C_g(\theta_d(t), \dot{\theta}_d(t))$$

- Αν έχουμε καλές προσεγγίσεις για το M και C_g, έχουμε πολύ καλό έλεγχο

- **Feedback control (PID):**

$$\tau(t) = K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt + K_d\dot{\theta}_e(t), \quad K_p, K_i, K_d > 0$$

- **Feedforward plus feedback linearizing controller (Inverse Dynamics Controller):**

$$\tau(t) = M(\theta_d(t))\Big(\ddot{\theta}_d(t) + K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt + K_d\dot{\theta}_e(t)\Big) + C_g(\theta_d(t), \dot{\theta}_d(t))$$

- **Gravity Compensation controller:**

$$\tau(t) = K_p\theta_e(t) + K_i \int_0^t \theta_e(t)dt + K_d\dot{\theta}_e(t) + g(\theta_d(t)), \quad K_p, K_i, K_d > 0$$

Έλεγχος με ροπές/δυνάμεις σε Task-Space

$$\tau = J^T \mathcal{F}_d$$

όπου $\mathcal{F}_d$ μπορεί να υπολογιστεί:

- Με ανατροφοδότηση (π.χ. PID controller)
- Με feedforward/open-loop controller (εκτός του σκοπού του μαθήματος)
- Με feedforward plus feedback linearizing controller (εκτός του σκοπού του μαθήματος)
- ....

Βιβλιογραφία

Κεφάλαια 6, 8 και 11 από **Modern Robotics: Mechanics, Planning, and Control**, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press. ebook