Τεχνολογίες Ευφυών Συστημάτων και Ρομποτική

Optimal Control και Trajectory Optimization

Κωνσταντίνος Χατζηλυγερούδης - costashatz@upatras.gr

Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

27 Μαρτίου 2023

Template made by **Παναγιώτης Παπαγιαννόπουλος**

Σχεδιασμός Κίνησης (Motion Planning) (1)

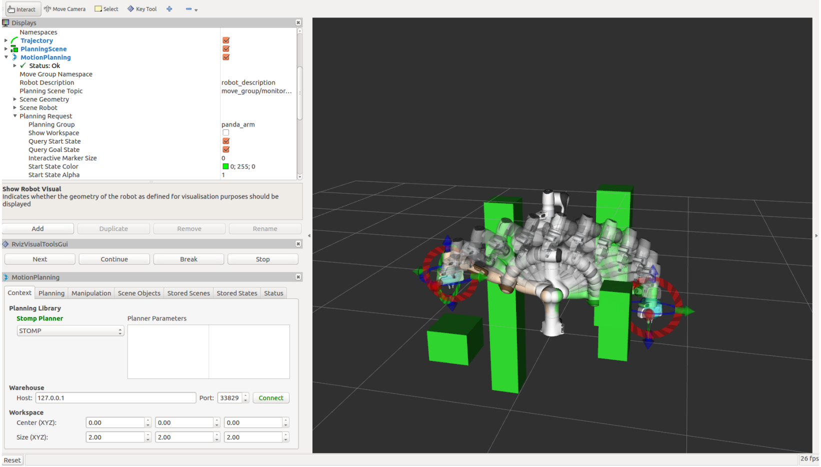
Ορισμός (Γενικό Πρόβλημα Σχεδιασμού Κίνησης)

Έστω μία αρχική κατάσταση $x(0) = x_{start}$ *και μία κατάσταση στόχος* x_{goal}, *θέλουμε να βρούμε τον χρόνο* T *και τις ενέργειες* $u : [0, T] \rightarrow \mathcal{U}$ *έτσι ώστε ακολουθώντας το μοντέλο του ρομπότ* $\dot{x} = f(x, u)$ *να φτάσουμε σε* $x(T) = x_{goal}$ *και* $\forall t \in [0, T]$ *η κατάσταση* $x(t)$ *είναι έγκυρη κατάσταση (καμία σύγκρουση ή εσφαλμένες θέσεις των αρθρώσεων).*

Σχεδιασμός Κίνησης (Motion Planning) (2)

- Υποθέτουμε ότι γνωρίζουμε το ρομπότ και το περιβάλλον
- Το μοντέλο f του ρομπότ μας δίνει τις εξισώσεις κίνησης (συνήθως σε επίπεδο ταχυτήτων)
- Υποθέτουμε ότι έχουμε κάποιον ελεγκτή για να "ακολουθήσει" την κίνηση $x(t)$

Σχεδιασμός Κίνησης (Motion Planning) (3)



Ας το δούμε σαν βελτιστοποίηση

$$\begin{aligned}
\underset{t_0, t_f, \boldsymbol{x}, \boldsymbol{u}}{\text{argmin}} \quad & \ell_f(t_0, t_f, \boldsymbol{x}(t_0), \boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \ell(t, \boldsymbol{x}(t), \boldsymbol{u}(t))dt \\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \forall t \in [t_0, t_f] \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_{init} \\
& \ldots
\end{aligned}$$

Ας το δούμε σαν βελτιστοποίηση

$$
\begin{aligned}
\underset{t_0, t_f, \boldsymbol{x}, \boldsymbol{u}}{\text{argmin}} \quad & \ell_f(t_0, t_f, \boldsymbol{x}(t_0), \boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \ell(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) dt \\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \forall t \in [t_0, t_f] \\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_{init} \\
& \ldots
\end{aligned}
$$

Δεν λύνεται στην γενική περίπτωση!

Practical Trajectory Optimization (Direct Collocation)

$$
\begin{aligned}
\underset{\boldsymbol{x}_k,\boldsymbol{u}_k}{\text{argmin}} \quad & \ell_f(\boldsymbol{x}_N) + \sum_{k=1}^{N-1} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k) \\
\text{s.t.} \quad & \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f(\boldsymbol{x}_k, \boldsymbol{u}_k)dt, k = 1, \dots, N-1 \\
& \boldsymbol{x}_1 = \boldsymbol{x}_{init} \\
& \dots
\end{aligned}
$$

Double Integrator (1)

Fig. 1 *Illustration of the boundary conditions for the simple block-move example.*

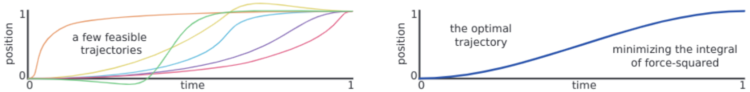


Fig. 2 *Comparison of feasible (left) and optimal (right) trajectories for the simple block-move example.*

Πηγή: Kelly, M., 2017. An introduction to trajectory optimization: How to do your own direct collocation. SIAM Review, 59(4), pp.849-904.

Double Integrator (2)

- Κατάσταση: $\boldsymbol{x} = [p, \dot{p}] \in \mathbb{R}^2$
- Ενέργειες: $\boldsymbol{u} = [\ddot{p}] \in \mathbb{R}$
- Δυναμικό Μοντέλο: $\dot{\boldsymbol{x}}(t) = [\dot{p}(t), \boldsymbol{u}(t)]$
- Τελικό Κόστος: $\ell_f(\boldsymbol{x}(t_f)) = 0$
- Κόστος: $\ell(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \ddot{p}(t)^2$
- Αρχική κατάσταση: $\boldsymbol{x}_{init} = [0, 0]$
- Τελική κατάσταση: $\boldsymbol{x}_{final} = [1, 0]$

Double Integrator (3)

$N = 3$
$dt = 0.5$

$$\begin{aligned} \underset{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{u}_1,\boldsymbol{u}_2,\boldsymbol{u}_3}{\operatorname{argmin}} \quad & \sum_{k=1}^{N-1} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \text{s.t.} \quad & \boldsymbol{x}_2 = \boldsymbol{x}_1 + f(\boldsymbol{x}_1, \boldsymbol{u}_1)dt \\ & \boldsymbol{x}_3 = \boldsymbol{x}_2 + f(\boldsymbol{x}_2, \boldsymbol{u}_2)dt \\ & \boldsymbol{x}_1 = \boldsymbol{x}_{init} \\ & \boldsymbol{x}_3 = \boldsymbol{x}_{final} \end{aligned}$$

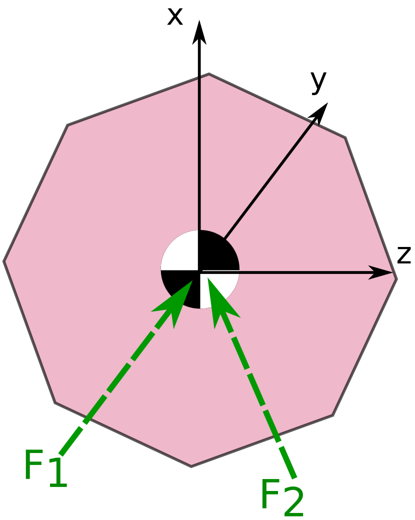
Double Integrator (3)

$N = 3$
$dt = 0.5$

$$
\begin{aligned}
\underset{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{u}_1,\boldsymbol{u}_2,\boldsymbol{u}_3}{\text{argmin}} \quad & \sum_{k=1}^{N-1} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k) \\
\text{s.t.} \quad & \boldsymbol{x}_2 = \boldsymbol{x}_1 + f(\boldsymbol{x}_1, \boldsymbol{u}_1)dt \\
& \boldsymbol{x}_3 = \boldsymbol{x}_2 + f(\boldsymbol{x}_2, \boldsymbol{u}_2)dt \\
& \boldsymbol{x}_1 = \boldsymbol{x}_{init} \\
& \boldsymbol{x}_3 = \boldsymbol{x}_{final}
\end{aligned}
$$

Προφανώς $N > 10$ σε ρεαλιστικές εφαρμογές!

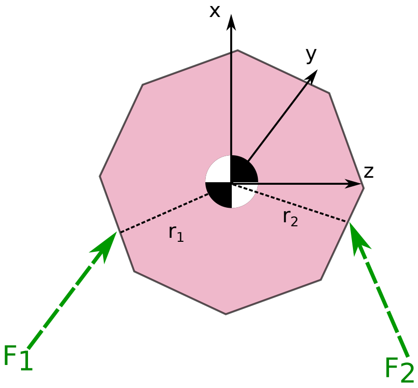
Δυναμικά Στερεού Σώματος (1)



Δυναμικά Στερεού Σώματος (2)

- Συνολική δύναμη: $\boldsymbol{F} = \sum_i \boldsymbol{F}_i$
- Εξίσωση: $\boldsymbol{F} = m\boldsymbol{a}$
- Euler Integration:
  - $\boldsymbol{a} = \ddot{\boldsymbol{x}} = \frac{\boldsymbol{F}}{m}$
  - $\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}} + \ddot{\boldsymbol{x}}dt$
  - $\boldsymbol{x} = \boldsymbol{x} + \dot{\boldsymbol{x}}dt$

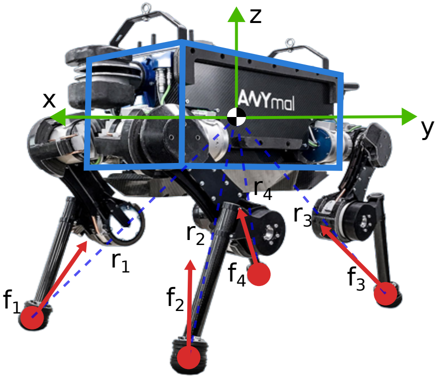
Δυναμικά Στερεού Σώματος (3)



Δυναμικά Στερεού Σώματος (4)

- Συνολική δύναμη: $\boldsymbol{F} = \sum_i \boldsymbol{F}_i$
- Συνολική ροπή: $\boldsymbol{\tau} = \sum_i (r_i \times \boldsymbol{F_i})$
- Εξίσωση (δύναμη): $\boldsymbol{F} = m\boldsymbol{a}$
- Εξίσωση (ροπή): $\boldsymbol{\tau} = \boldsymbol{I}_{cm}\boldsymbol{\alpha} + \boldsymbol{\omega} \times \boldsymbol{I}_{cm}\boldsymbol{\omega}$
- Euler Integration:
  - $\boldsymbol{a} = \ddot{\boldsymbol{x}} = \frac{\boldsymbol{F}}{m}$
  - $\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}} + \ddot{\boldsymbol{x}}dt$
  - $\boldsymbol{x} = \boldsymbol{x} + \dot{\boldsymbol{x}}dt$
  - $\alpha = \dot{\omega} = \boldsymbol{I}_{cm}^{-1}(R^T\boldsymbol{\tau} - \omega \times \boldsymbol{I}_{cm}\omega)$
  - $\omega = \dot{\omega} + \dot{\omega}dt$
  - $R = R\exp(\omega dt)$

Γιατί είναι σημαντικό;



Βιβλιογραφία

Κεφάλαια 3, 8, 9 και 10 από **Modern Robotics: Mechanics, Planning, and Control**, *Kevin M. Lynch and Frank C. Park*, 2017, Cambridge University Press. ebook

- An introduction to trajectory optimization: How to do your own direct collocation
- Trajectory Optimization Tutorials
- MIT Underactuated Course