



Τεχνολογίες Ευφών Συστημάτων και Ρομποτική

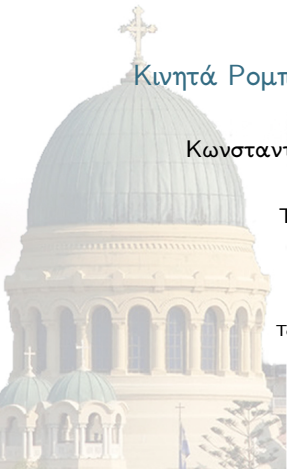
Κινητά Ρομπότ, Σχεδιασμός Κίνησης και Behavior Trees

Κωνσταντίνος Χατζηλυγερούδης - costashatz@upatras.gr

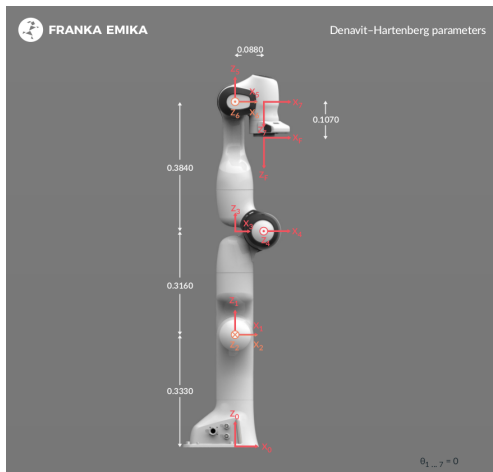
Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

20 Μαρτίου 2023

Template made by Παναγιώτης Παπαγιαννόπουλος



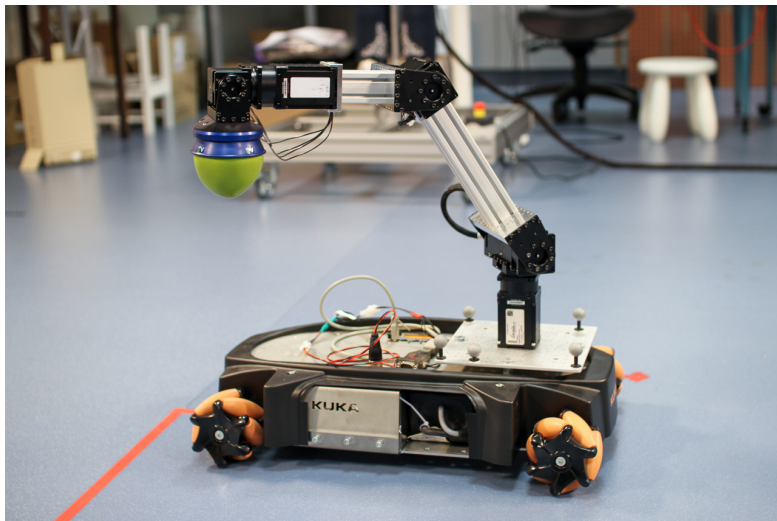
Βραχίονες



Πηγή: <https://petercorke.com/robotics/franka-emika-panda-kinematics-and-singularities/>

- Συνήθως στατικά ρομπότ (fixed to world)
- Ορίζονται πλήρως από τις θέσεις και ταχύτητες των αρθρώσεων
- $x = (q, \dot{q})$ όπου $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$, n βαθμοί ελευθερίας
- Τι γίνεται όμως αν αλλάξουμε την θέση τους στο χώρο;
- Ή αν τα βάλουμε πάνω σε ένα κινητό ρομπότ;

Βραχίονες σε Κινητά Ρομπότ

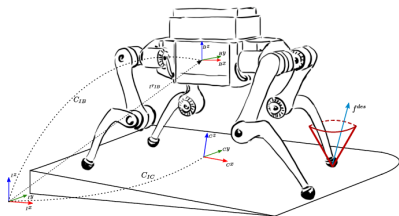
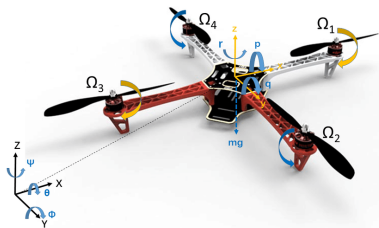


Πηγή: <https://www.resibots.eu/photos.html#kuka-youbot-versaball>

Κινητά Ρομπότ (Mobile Robots) (1)



Κινητά Ρομπότ (Mobile Robots) (2)



- Ορίζονται από τις θέσεις και ταχύτητες των αρθρώσεων
- ΚΑΙ τη θέση και προσανατολισμό του *κεντρικού σώματος* (base link)
- Άρα έχουμε:

$x = (T_{wb}, \dot{T}_{wb}, q, \dot{q})$ όπου $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$, T_{wb} το μητρώο μετασχηματισμού του base link και \dot{T}_{wb} οι ταχύτητες που κινείται το base link

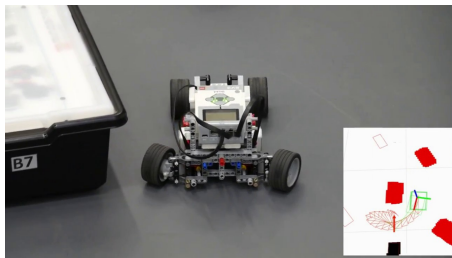
- Συνολικά έχουμε (το πολύ) $n + 6$ βαθμούς ελευθερίας

- 1 Ελεύθερα κινούμενα (free-floating robots)
 - 6 επιπλέον βαθμούς ελευθερίας
 - Ανθρωποειδή, drones, ...
- 2 Κινούμενα σε επίπεδο (planar/classical mobile robots)
 - Ρομπότ με ρόδες!
 - 3 επιπλέον βαθμούς ελευθερίας
 - Αυτοκίνητα, ποδήλατα, omnidirectional, ...
- 3 ...

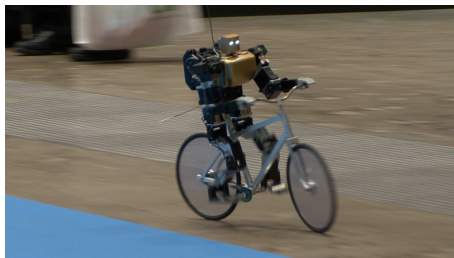
Ελεύθερα Κινούμενα Ρομπότ



Κινούμενα σε επίπεδο Ρομπότ (1)



Κινούμενα σε επίπεδο Ρομπότ (2)

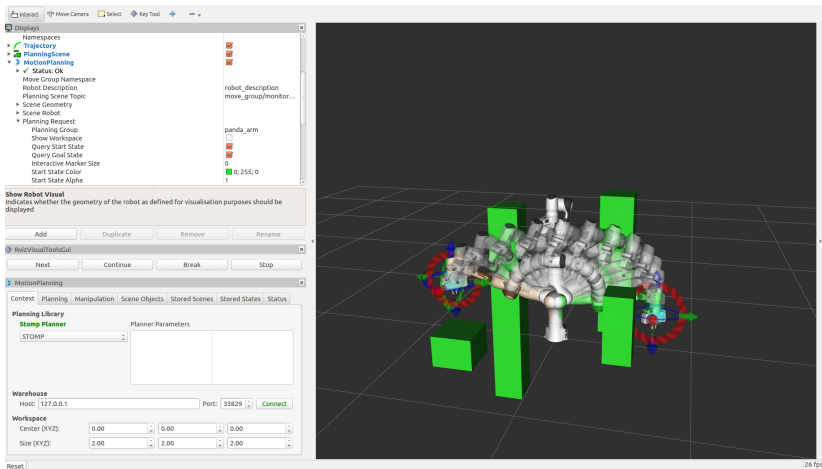


Ορισμός (Γενικό Πρόβλημα Σχεδιασμού Κίνησης)

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).

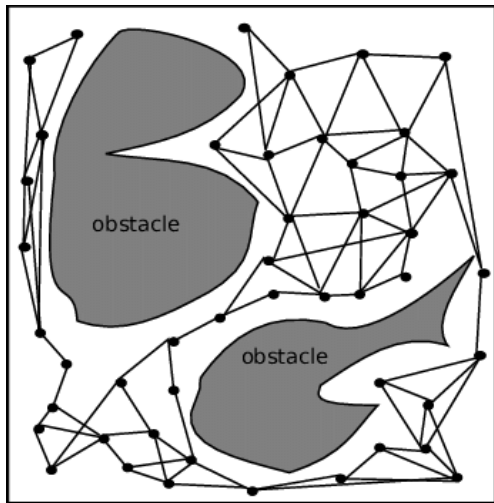
- Υποθέτουμε ότι γνωρίζουμε το ρομπότ και το περιβάλλον
- Το μοντέλο f του ρομπότ μας δίνει τις εξισώσεις κίνησης (συνήθως σε επίπεδο ταχυτήτων)
- Υποθέτουμε ότι έχουμε κάποιον ελεγκτή για να “ακολουθήσει” την κίνηση $x(t)$

Σχεδιασμός Κίνησης (Motion Planning) (3)



- Η αναζήτηση παίζει πολύ σημαντικό ρόλο στους αλγόριθμους σχεδιασμού κίνησης
- Πολλά προβλήματα σχεδιασμού κίνησης μπορούν να λυθούν με αναζήτηση
- Οι περισσότεροι αλγόριθμοι σχεδιασμού κίνησης περιγράφουν το χώρο χρησιμοποιώντας κάποιο δέντρο ή γράφο

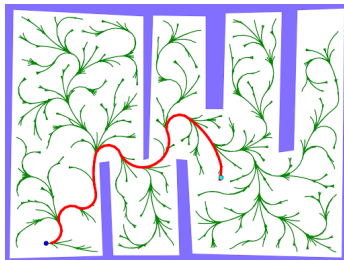
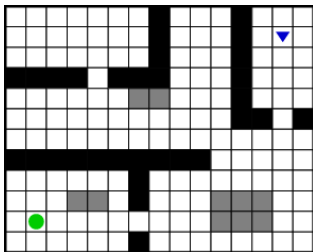
Σχεδιασμός Κίνησης και Αναζήτηση (2)



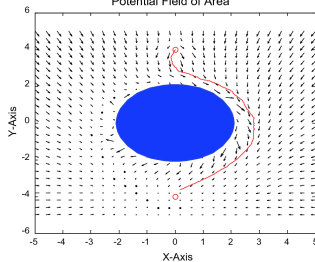
Πηγή: Murray, S., Floyd-Jones, W., Qi, Y., Sorin, D.J. and Konidaris, G.D., 2016, June. Robot Motion Planning on a Chip. In Robotics: Science and Systems.

- Πλήρης
- Grid-based
- Sampling-based
- Virtual Potential Fields
- Μη-γραμμική Βελτιστοποίηση

Κατηγορίες Αλγορίθμων Σχεδιασμού Κίνησης (2)



Potential Field of Area



ORB-SLAM

Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós

{raulmur, josemari, tardos} @unizar.es



Instituto Universitario de Investigación
en Ingeniería de Aragón
Universidad Zaragoza



Universidad
Zaragoza



Universidad
Zaragoza



Instituto Universitario de Investigación
en Ingeniería de Aragón
Universidad Zaragoza

ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras

Raúl Mur-Artal and Juan D. Tardós

raulmur@unizar.es

tardos@unizar.es

Τι γίνεται αν θέλουμε να σχεδιάσουμε ενέργειες σε υψηλότερο επίπεδο (high-level);

- Τι εννοούμε με την έκφραση “υψηλότερο επίπεδο”;

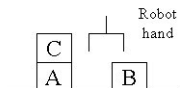
Τι γίνεται αν θέλουμε να σχεδιάσουμε ενέργειες σε υψηλότερο επίπεδο (high-level);

- Τι εννοούμε με την έκφραση “υψηλότερο επίπεδο”;
- Τι είδους αναπαράσταση θα χρησιμοποιήσουμε για τις καταστάσεις;
- STRIPS Planning
- PDDL (Planning Domain Definition Language)

- Οι καταστάσεις αναπαριστώνται σαν μία σειρά από γεγονότα
- Για παράδειγμα:
`holding(A), clear(B), on(B,C), onTable(C)`
- Οι στόχοι αναπαριστώνται με τον ίδιο τρόπο
- Οι ενέργειες έχουν 3 κατηγορίες γεγονότων:
 - Pre-conditions: γεγονότα που πρέπει να ικανοποιούνται για να χρησιμοποιηθεί αυτή η ενέργεια
 - Add: γεγονότα που προστίθενται στην κατάσταση μετά την ενεργοποίηση της ενέργειας
 - Delete: γεγονότα που διαγράφονται από την κατάσταση
- Συνδυάζεται πολύ καλά με *Λογική*

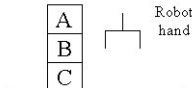
Example : Blocks World

- **STRIPS** : A planning system – Has rules with precondition deletion list and addition list



START

on(B, table)
 on(A, table)
 on(C, A)
 hand empty
 clear(C)
 clear(B)



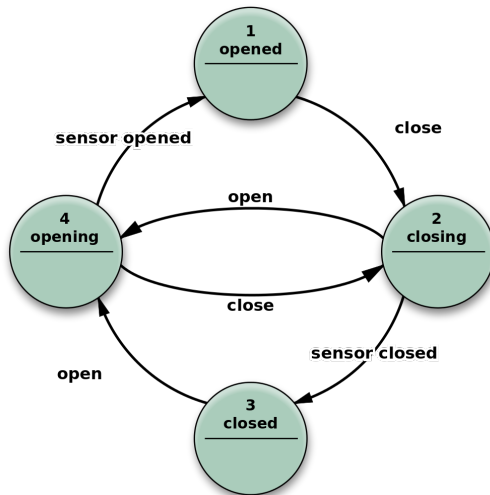
GOAL

on(C, table)
 on(B, C)
 on(A, B)
 hand empty
 clear(A)

- Ο STRIPS planner δεν είναι ο μόνος τρόπος αναπαράστασης
- Η PDDL (Planning Domain Definition Language) δημιουργεί μία κοινή γλώσσα για όλους τους “παραδοσιακούς” αλγορίθμους/αναπαραστάσεις σχεδιασμού ενεργειών
- <https://www.cs.toronto.edu/~sheila/2542/s14/A1/introtopddl2.pdf> για παραπάνω πληροφορίες

- Τα ρομποτικά συστήματα αλληλεπιδρούν με τον πραγματικό κόσμο
- Δεν μπορούμε να έχουμε ένα στατικό πλάνο
- Επανασχεδίαση αν κάτι δεν πάει καλά;
- Τι άλλο μπορούμε να κάνουμε;

Finite State Machines (FSM)



Πηγή: https://en.wikipedia.org/wiki/File:Fsm_Moore_model_door_control.svg

Οι έξυπνοι πράκτορες πρέπει να έχουν:

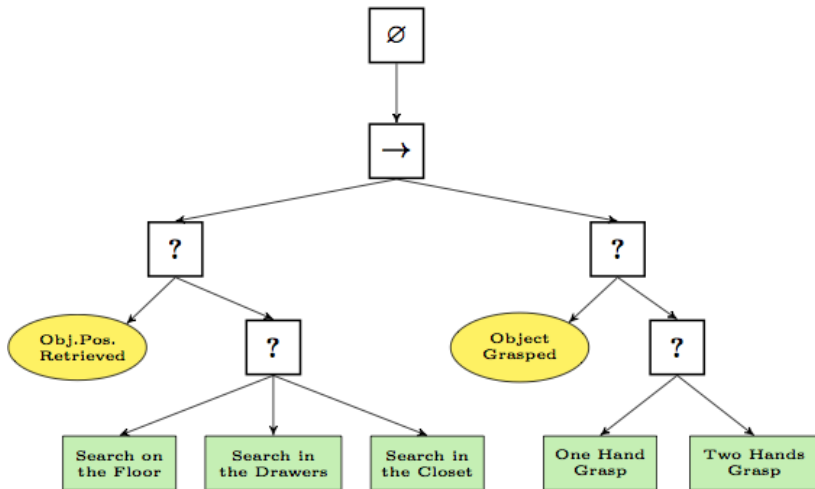
- Αντιδραστικότητα (reactiveness)
- Αρθρωτότητα (modularity)

Οι έξυπνοι πράκτορες πρέπει να έχουν:

- Αντιδραστικότητα (reactiveness)
- Αρθρωτότητα (modularity)

Τα FSMs δεν τα πάνε καθόλου καλά με το 2ο! Τι άλλο μπορούμε να κάνουμε;

Behavior Trees: Δέντρα με πληροφορία χρόνου;



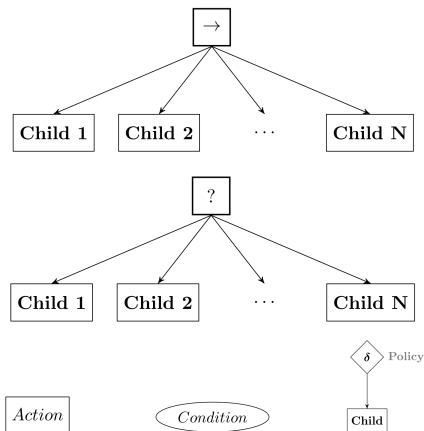
Πηγή: https://upload.wikimedia.org/wikipedia/commons/1/1b/BT_search_and_grasp.png

Σε κάθε “timestep” ένας χτύπος (tick) ξεκινάει από την ρίζα του δέντρου και προχωράει προς τα κάτω.

- Κάθε κόμβος επιστρέφει ένα από τα παρακάτω:
 - Success
 - Failure
 - Running
- Ανάλογα με τον κόμβο (και το αποτέλεσμα του) σταματάμε ή συνεχίζουμε την διαδρομή

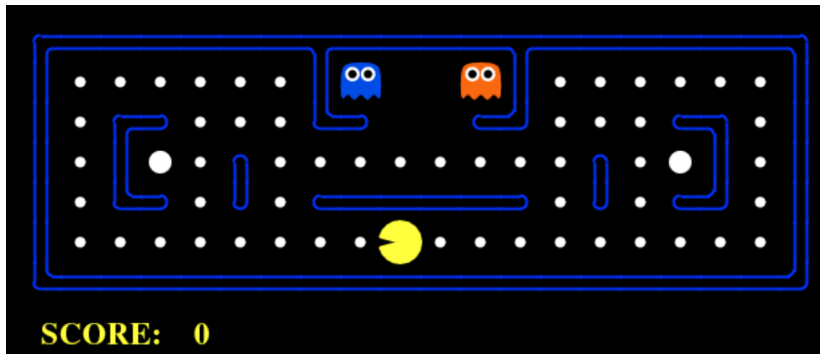
Behavior Trees: Τύποι κόμβων

- Root
- Sequence
- Fallback
- Parallel
- Decorator
- **Action**
- **Condition**



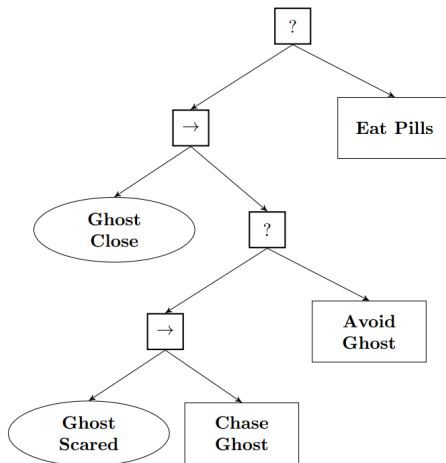
Πηγή: Colledanchise, M. and Ögren, P., 2018. Behavior trees in robotics and AI: An introduction. CRC Press.

Behavior Trees: Παράδειγμα (1)



Πηγή: Colledanchise, M. and Ögren, P., 2018. Behavior trees in robotics and AI: An introduction. CRC Press.

Behavior Trees: Παράδειγμα (2)



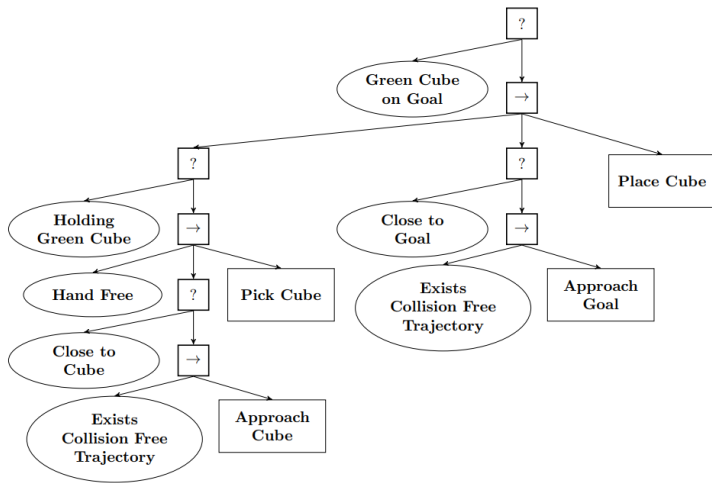
Πηγή: Colledanchise, M. and Ögren, P., 2018. Behavior trees in robotics and AI: An introduction. CRC Press.

Behavior Trees: Παράδειγμα (1)



Πηγή: Colledanchise, M. and Ögren, P., 2018. Behavior trees in robotics and AI: An introduction. CRC Press.

Behavior Trees: Παράδειγμα (2)



Πηγή: Colledanchise, M. and Ögren, P., 2018. Behavior trees in robotics and AI: An introduction. CRC Press.

Κεφάλαια 2, 10 και 13 από **Modern Robotics: Mechanics, Planning, and Control**, Kevin M. Lynch and Frank C. Park, 2017, Cambridge University Press. ebook

- Finite State Machines (Wikipedia)
- STRIPS (Wikipedia)
- PDDL
- Colledanchise, M. and Ögren, P., 2018. Behavior trees in robotics and AI: An introduction. CRC Press.