



# Robotika: Plánování úloh a pohybu

Vladimír Petřík

[vladimir.petrik@cvut.cz](mailto:vladimir.petrik@cvut.cz)

13.11.2023

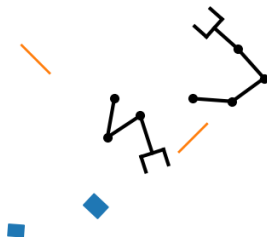
# Motivace

- ▶ Víme, jak naplánovat pohyb robota v konfiguračním prostoru robota
  - ▶ ručně definovat úchyt na objektu
  - ▶ vypočítat úchop a předúchop pro detekovanou polohu objektu
  - ▶ naplánovat pohyb pro předuchopení
  - ▶ interpolovat pro uchopení, uchopení
  - ▶ interpolovat na předuchopení
  - ▶ naplánovat pohyb pro předumístění, umístění, uvolnění chapadla, předumístění
- ▶ Co když máme mnoho úchytů? Mnoho objektů?
- ▶ **Plánování manipulačních úloh a pohybu (Task and Motion Planning, TAMP)**
  - ▶ simultánní plánování úloh a pohybů
  - ▶ úloha je daná posloupností úchopů a umístění (diskrétní prostor)
  - ▶ pohyb je posloupnost konfigurací robota (spojitý prostor)
  - ▶ Přístup Humanoid Path Planning (HPP)



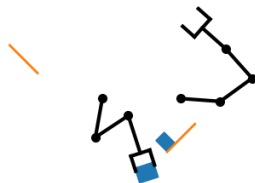
# Konfigurační prostor

- ▶ Více chapadel připojených k robotům
- ▶ Povrchy v prostředí, na které lze objekt umístit
- ▶ Více objektů
  - ▶ více úchytů na objekt
  - ▶ více povrchů pro kontakt
- ▶ Konfigurační prostor je množina všech konfigurací všech objektů a robotů
  - ▶  $\mathcal{C} = \mathbb{R}^{N_1} \times \mathbb{R}^{N_2} \dots \times SE(3)^M$
  - ▶  $N_i$  DoF  $i$ -tého robota
  - ▶  $M$  počet objektů
  - ▶ nicméně, ne všechny konfigurace jsou přípustné
  - ▶ omezení (constraints) jsou použita k definici přípustných konfigurací



# Omezení

- ▶ Objekt je umístěn nebo uchopen, tj. nemůže létat
- ▶ Omezení pro umístění
  - ▶ objekt leží na povrchu
  - ▶ numerické omezení
  - ▶ **povrch objektu** je umístěn na **povrch prostředí**
- ▶ Omezení pro uchopení
  - ▶ objekt je uchopen chapadlem
  - ▶ numerické omezení
  - ▶ **s.s. úchopu** se rovná **s.s. chapadla**

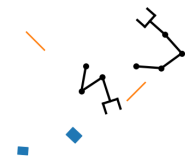


## Stav (state)

- ▶ Stav je soubor omezení
- ▶ Manifold přípustných konfigurací v konfiguračním prostoru
- ▶ Například jeden stav může být definován omezením obou objektů
  - ▶ objekt  $O_1$  je umístěn na povrchu  $E_1$  prostřednictvím povrchu objektu  $S_1$
  - ▶ objekt  $O_2$  je uchopen chapadlem  $G_1$  prostřednictvím úchopu  $H_1$
- ▶ Jak vzorkovat konfiguraci ze stavu?
  - ▶ vzorek ze  $\mathcal{C}$
  - ▶ geometrická projekce pro splnění všech omezení
  - ▶ numerická optimalizace (Newton-Raphson) ke splnění všech omezení



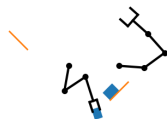
# Vzorkování ze stavů



Vzorek z  $\mathcal{C}$



- ▶ Projekce do stavu:
  - ▶  $O_1$  umístěn na  $E_1$  přes  $S_1$
  - ▶  $O_2$  umístěn na  $E_2$  přes  $S_1$



- ▶ Projekce do stavu:
  - ▶  $O_1$  uchopen  $G_1$  přes  $H_1$
  - ▶  $O_2$  umístěn na  $E_2$  přes  $S_1$

# Přechody (Transitions)

- ▶ Přechod definuje pohyb mezi dvěma stavy
  - ▶ přechod *identita* umožňuje pohyb robota uvnitř stavu
  - ▶ přechod *umístění* umožňuje přesunout objekt z chapadla na povrch
  - ▶ přechod *uchopení* umožňuje přesunout objekt z povrchu do uchopovače
- ▶ Vzorkování z přechodů vs. vzorkování ze stavů
  - ▶ přechod respektuje omezení z dané konfigurace
  - ▶ např. *identita* na stavu, kde objekt leží, nepřesune objekt (vzorkování na stavu může objekt přesunout)
  - ▶ přechod *uchopení* je specifikován tak, aby se přesunul přes předuchopení
  - ▶ přechod *umístění* je specifikován tak, aby se pohyboval přes předumístění



## Interpolace na přechodu

- ▶ Interpolace mezi dvěma konfiguracemi, ale respektuje omezení stavů/přechodu



Uchopení

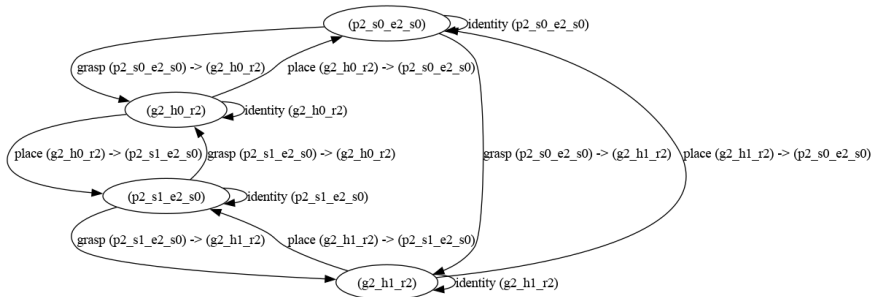


Umístění



# Graf omezení

- ▶ Definuje všechny možné přechody mezi existujícími stavy
- ▶ Příklad: jedno rameno, jeden objekt

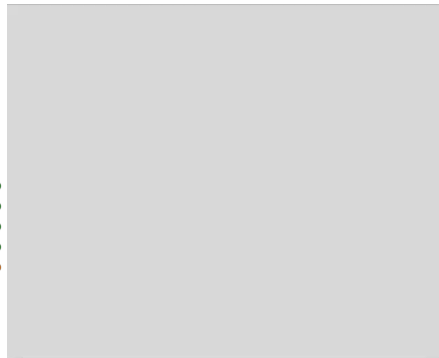
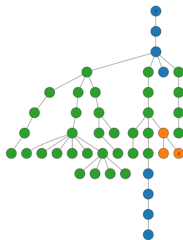
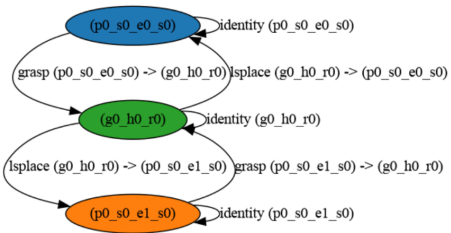


# RRT na grafu omezení

- ▶ Náhodný vzorek  $q_{\text{rand}}$ 
  - ▶ vybrat náhodný přechod
  - ▶ vybrat náhodnou existující konfiguraci z počátku přechodu
  - ▶ vybrat náhodnou konfiguraci z cíle přechodu, který je dosažitelný z počátku
- ▶ Nejbližší soused  $q_{\text{tree}}$ 
  - ▶ uzel, který je nejbližší k  $q_{\text{rand}}$  přes interpolaci na přechodu
- ▶ Lokální plánovač používá interpolaci na přechodu



# RRT na grafu omezení



# Závěr

- ▶ Konfigurační prostor pro TAMP je komplexní
  - ▶ diskrétní množina stavů
  - ▶ spojitý pohyb
  - ▶ zakódovaný pomocí grafu omezení, který nám umožňuje použít RRT
- ▶ Obvykle se v průmyslu nepoužívá
  - ▶ posloupnost úkolů v prostoru je pevně zakódována programátory
  - ▶ plánovač je použit pouze pro pohyb (pokud nemůže být pevně zakódován)
- ▶ Jak se vyhnout tvrdému kódování? Video demonstrace



- ▶ Konzultace k závěrečnému projektu
- ▶ Závěrečný projekt je nyní popsán na webové stránce kurzu
- ▶ Nové rozhraní pro robota Bosch [nepovinné]
  - ▶ opraveno FK, IK
  - ▶ můžete si jej nainstalovat na svůj počítač, abyste mohli používat FK a IK offline pro ladění

