



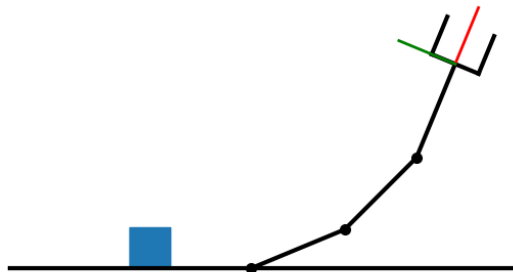
Robotika: Generování cest a trajektorií

Vladimír Petřík

vladimir.petrík@cvut.cz

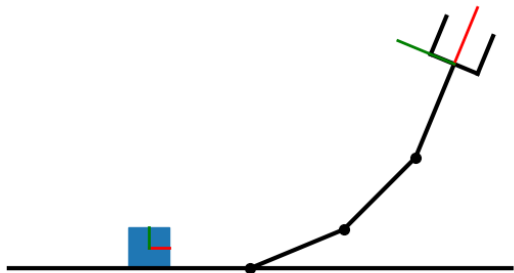
10.11.2024

Motivace: úchop kostky



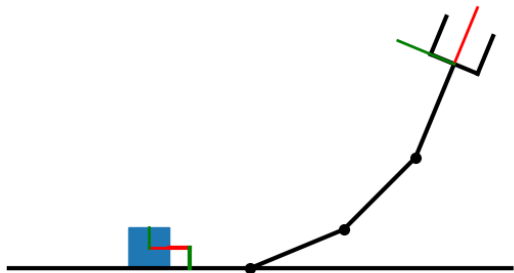
Motivace: úchop kostky

- Detekce, kde je kostka v $SE(2)$, $SE(3)$



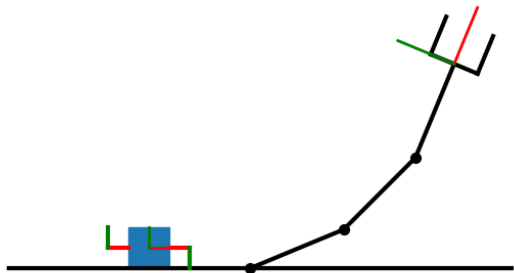
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla



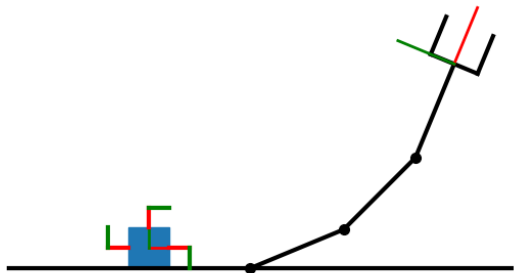
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla



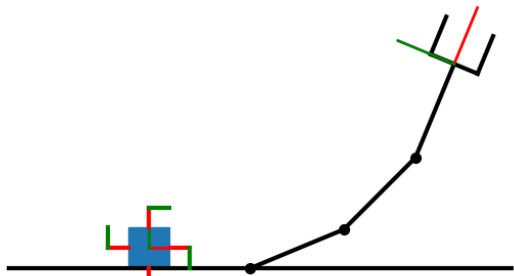
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla



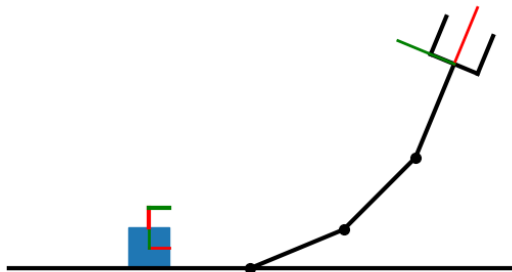
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla



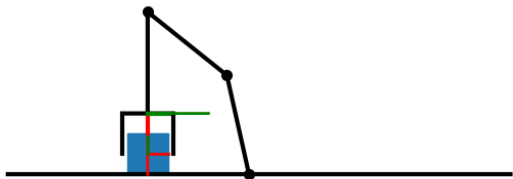
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla
- ▶ Řešení IK (vybrat jedno z řešení, jak?)



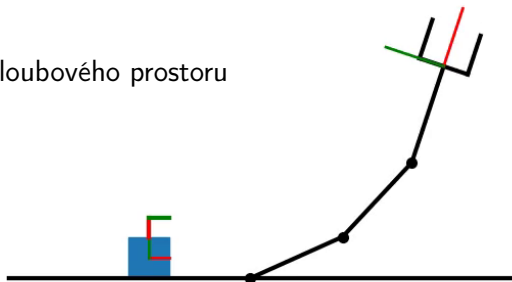
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla
- ▶ Řešení IK (vybrat jedno z řešení, jak?)
- ▶ Odeslání robota do vybrané konfigurace kloubového prostoru
- ▶ Jaký pohyb bude robot vykonávat?



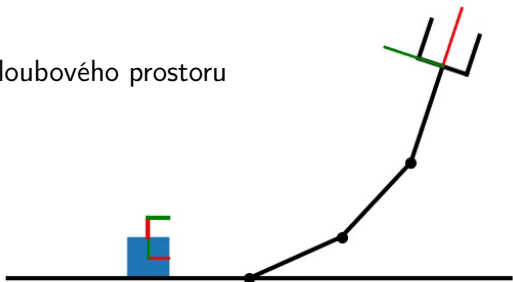
Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla
- ▶ Řešení IK (vybrat jedno z řešení, jak?)
- ▶ Odeslání robota do vybrané konfigurace kloubového prostoru
- ▶ Jaký pohyb bude robot vykonávat?
 - ▶ závisí na robotovi
 - ▶ lineární interpolace v prostoru kloubů je běžná



Motivace: úchop kostky

- ▶ Detekce, kde je kostka v $SE(2)$, $SE(3)$
- ▶ Definování úchytu(ů) vzhledem ke kostce
- ▶ Výpočet polohy chapadla
- ▶ Řešení IK (vybrat jedno z řešení, jak?)
- ▶ Odeslání robota do vybrané konfigurace kloubového prostoru
- ▶ Jaký pohyb bude robot vykonávat?
 - ▶ závisí na robotovi
 - ▶ lineární interpolace v prostoru kloubů je běžná
 - ▶ co je pohyb?



- ▶ Cesta (path)
 - ▶ Geometrický popis (sekvence konfigurací)
 - ▶ Bez časových značek, dynamiky a omezení na kontrolní signál
 - ▶ $q(s) \in \mathcal{C}_{\text{free}}, s \in [0, 1]$
 - ▶ Hlavním předpokladem je, že následným zpracováním je možné získat trajektorii

- ▶ Cesta (path)
 - ▶ Geometrický popis (sekvence konfigurací)
 - ▶ Bez časových značek, dynamiky a omezení na kontrolní signál
 - ▶ $q(s) \in \mathcal{C}_{\text{free}}, s \in [0, 1]$
 - ▶ Hlavním předpokladem je, že následným zpracováním je možné získat trajektorii
- ▶ Trajektorie (trajectory)
 - ▶ Popisuje konfiguraci robota v čase
 - ▶ $q(t) \in \mathcal{C}_{\text{free}}, t \in [0, T]$



Pohyb

- ▶ Cesta (path)
 - ▶ Geometrický popis (sekvence konfigurací)
 - ▶ Bez časových značek, dynamiky a omezení na kontrolní signál
 - ▶ $\mathbf{q}(s) \in \mathcal{C}_{\text{free}}, s \in [0, 1]$
 - ▶ Hlavním předpokladem je, že následným zpracováním je možné získat trajektorii
- ▶ Trajektorie (trajectory)
 - ▶ Popisuje konfiguraci robota v čase
 - ▶ $\mathbf{q}(t) \in \mathcal{C}_{\text{free}}, t \in [0, T]$



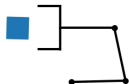
Cesta k uchopení

- ▶ Zaměříme se nejprve na **cestu**
- ▶ Je uchopovací cesta bezpečná? Záleží na počáteční konfiguraci.



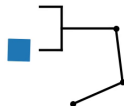
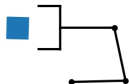
Cesta k uchopení

- ▶ Zaměříme se nejprve na **cestu**
- ▶ Je uchopovací cesta bezpečná? Záleží na počáteční konfiguraci.



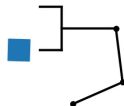
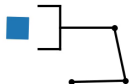
Cesta k uchopení

- ▶ Zaměříme se nejprve na **cestu**
- ▶ Je uchopovací cesta bezpečná? Záleží na počáteční konfiguraci.



Cesta k uchopení

- ▶ Zaměříme se nejprve na **cestu**
- ▶ Je uchopovací cesta bezpečná? Záleží na počáteční konfiguraci.



Předsunutá úchopová pozice

- ▶ Můžeme definovat pozici před uchopením
 - ▶ např. 5 cm od objektu, s ohledem na úchyt.
 - ▶ jak definovat vzdálenost 5 cm? Podle konstrukce úchytu.
 - ▶ stanovíme orientaci rukojeti tak, aby osa x směřovala k objektu.
 - ▶ orientace chapadla tak, aby osa x směřovala ven z chapadla.



Předsunutá úchopová pozice

- ▶ Můžeme definovat pozici před uchopením
 - ▶ např. 5 cm od objektu, s ohledem na úchyt.
 - ▶ jak definovat vzdálenost 5 cm? Podle konstrukce úchytu.
 - ▶ stanovíme orientaci rukojeti tak, aby osa x směřovala k objektu.
 - ▶ orientace chapadla tak, aby osa x směřovala ven z chapadla.
 - ▶ uchopovací pozice T_{RH}
 - ▶ pokud se chapadlo T_{RG} rovná T_{RH} , objekt je uchopen
 - ▶ pozice před uchopením $T_{RP} = T_{RH}T_x(-\delta_{\text{pre_grasp}})$



Předsunutá úchopová pozice

- ▶ Můžeme definovat pozici před uchopením
 - ▶ např. 5 cm od objektu, s ohledem na úchyt.
 - ▶ jak definovat vzdálenost 5 cm? Podle konstrukce úchytu.
 - ▶ stanovíme orientaci rukojeti tak, aby osa x směřovala k objektu.
 - ▶ orientace chapadla tak, aby osa x směřovala ven z chapadla.
 - ▶ uchopovací pozice T_{RH}
 - ▶ pokud se chapadlo T_{RG} rovná T_{RH} , objekt je uchopen
 - ▶ pozice před uchopením $T_{RP} = T_{RH}T_x(-\delta_{\text{pre_grasp}})$
- ▶ Je cesta z předuchovení do uchopení bezpečná, pokud je $\delta_{\text{pre_grasp}}$ velká?

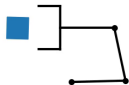


Předsunutá úchopová pozice

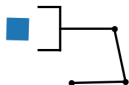
- ▶ Můžeme definovat pozici před uchopením
 - ▶ např. 5 cm od objektu, s ohledem na úchyt.
 - ▶ jak definovat vzdálenost 5 cm? Podle konstrukce úchytu.
 - ▶ stanovíme orientaci rukojeti tak, aby osa x směřovala k objektu.
 - ▶ orientace chapadla tak, aby osa x směřovala ven z chapadla.
 - ▶ uchopovací pozice T_{RH}
 - ▶ pokud se chapadlo T_{RG} rovná T_{RH} , objekt je uchopen
 - ▶ pozice před uchopením $T_{RP} = T_{RH}T_x(-\delta_{\text{pre_grasp}})$
- ▶ Je cesta z předuchovení do uchopení bezpečná, pokud je $\delta_{\text{pre_grasp}}$ velká?
- ▶ Je cesta z předuchovení do uchopení bezpečná, pokud je $\delta_{\text{pre_grasp}}$ malá?



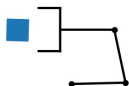
Předsunutá úchopová pozice



Předsunutá úchopová pozice



Předsunutá úchopová pozice



Interpolace v kloubovém prostoru

- ▶ Nazývá se také přímá cesta, cesta z bodu do bodu
- ▶ Start $\mathbf{q}_{\text{start}}$
- ▶ Cíl $\mathbf{q}_{\text{goal}}$
- ▶ $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}), \quad s \in [0, 1]$
- ▶ Snadný výpočet, dobře definovaný

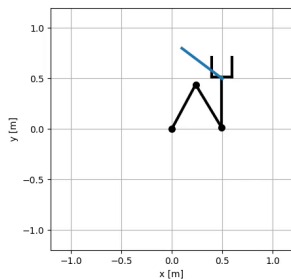


Interpolace v kloubovém prostoru

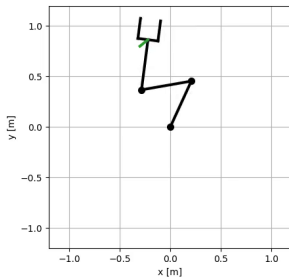
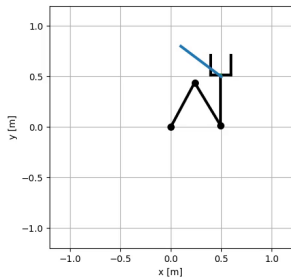
- ▶ Nazývá se také přímá cesta, cesta z bodu do bodu
- ▶ Start $\mathbf{q}_{\text{start}}$
- ▶ Cíl $\mathbf{q}_{\text{goal}}$
- ▶ $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}), \quad s \in [0, 1]$
- ▶ Snadný výpočet, dobře definovaný
- ▶ Jaký je pohyb chapadla?
 - ▶ pravděpodobně není přímočarý (pro otočné klouby)
 - ▶ kombinace kruhových drah (pro otočné klouby)



Interpolace v kloubovém prostoru



Interpolace v kloubovém prostoru



Interpolace v $SE(2)$ a $SE(3)$

- ▶ Přímá cesta v prostoru úloh
 - ▶ pozice $\mathbf{t}(s) = \mathbf{t}_{\text{start}} + s(\mathbf{t}_{\text{goal}} - \mathbf{t}_{\text{start}})$, $s \in [0, 1]$



Interpolace v $SE(2)$ a $SE(3)$

- ▶ Přímá cesta v prostoru úloh

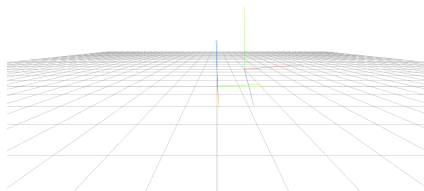
- ▶ pozice $\mathbf{t}(s) = \mathbf{t}_{\text{start}} + s(\mathbf{t}_{\text{goal}} - \mathbf{t}_{\text{start}})$, $s \in [0, 1]$
- ▶ rotace $R(s) = R_{\text{start}} \exp(s \log(R_{\text{start}}^{-1} R_{\text{goal}}))$, $s \in [0, 1]$



Interpolace v $SE(2)$ a $SE(3)$

► Přímá cesta v prostoru úloh

- pozice $\mathbf{t}(s) = \mathbf{t}_{\text{start}} + s(\mathbf{t}_{\text{goal}} - \mathbf{t}_{\text{start}})$, $s \in [0, 1]$
- rotace $R(s) = R_{\text{start}} \exp(s \log(R_{\text{start}}^{-1} R_{\text{goal}}))$, $s \in [0, 1]$



Cesta v kloubovém prostoru z cesty v úkolovém prostoru

- ▶ Výpočet $\mathbf{q}(s)$ z $T_{RG}(s)$
- ▶ Vyřešit IK pro každé s a vybrat první řešení IK?



Cesta v kloubovém prostoru z cesty v úkolovém prostoru

- ▶ Výpočet $\mathbf{q}(s)$ z $T_{RG}(s)$
- ▶ Vyřešit IK pro každé s a vybrat první řešení IK?
 - ▶ nedefinovali jsme, co je *první* řešení IK
 - ▶ použijeme nejbližší řešení IK
 - ▶ může se stát, že nejbližší řešení není *dostatečně blízké*?



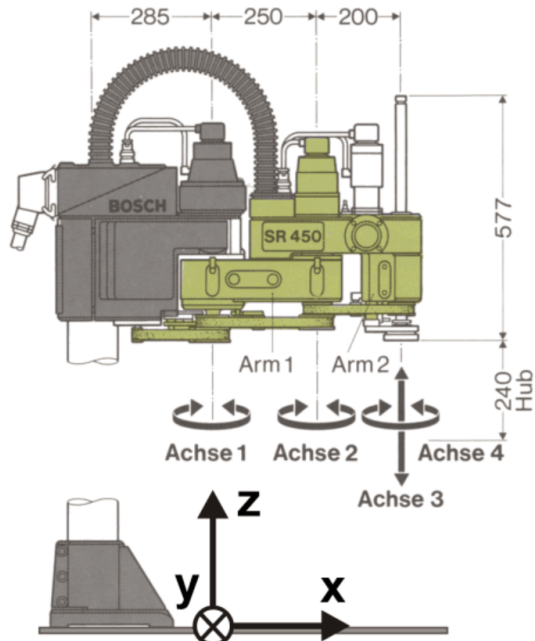
Cesta v kloubovém prostoru z cesty v úkolovém prostoru

- ▶ Výpočet $\mathbf{q}(s)$ z $T_{RG}(s)$
- ▶ Vyřešit IK pro každé s a vybrat první řešení IK?
 - ▶ nedefinovali jsme, co je *první* řešení IK
 - ▶ použijeme nejbližší řešení IK
 - ▶ může se stát, že nejbližší řešení není *dostatečně blízké*? **ano**, podívejme se na příklad

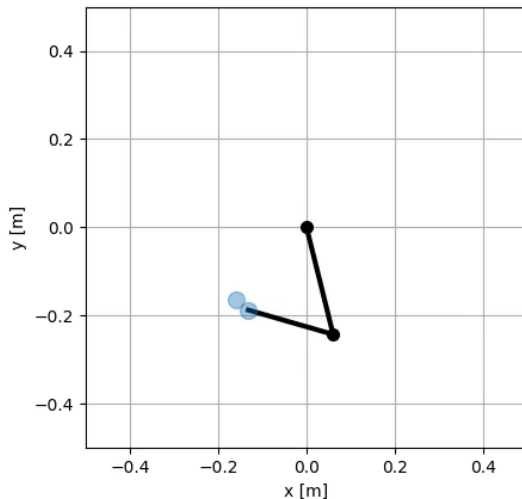


SCARA robot

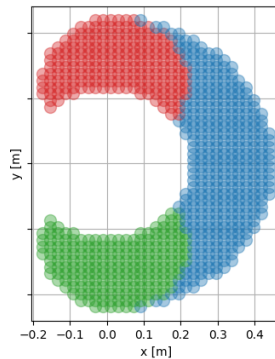
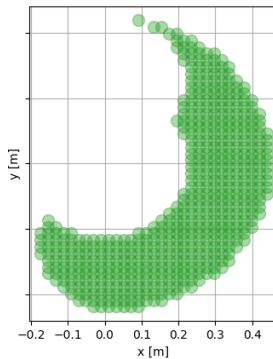
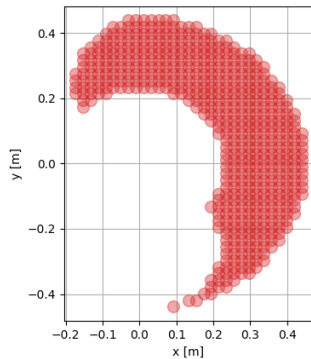
- ▶ Analýza kinematiky SCARA
- ▶ Struktura RRPR
- ▶ Zabránění samokolizím pomocí omezení pohybu kloubů
 - ▶ $\pm 85^\circ$
 - ▶ $\pm 120^\circ$
 - ▶ $(-330 \text{ mm}, 5 \text{ mm})$
 - ▶ $(-20^\circ, 1080^\circ)$
- ▶ Výpočet FK a IK v rovině xy



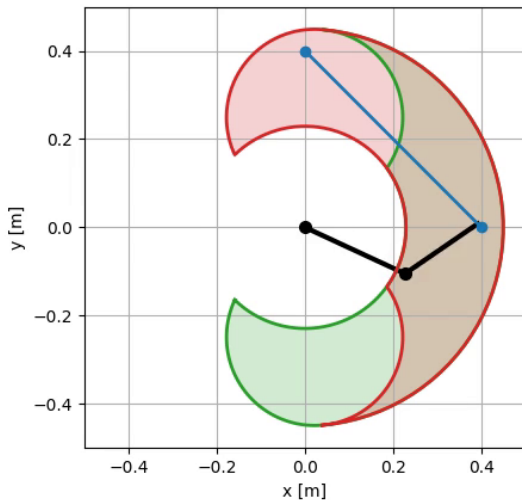
SCARA robot pracovní prostor



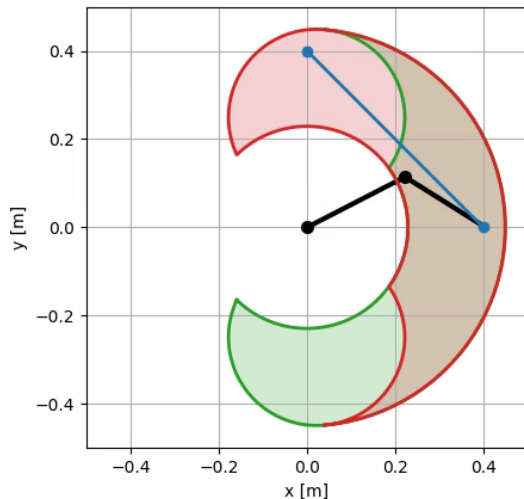
SCARA robot IK



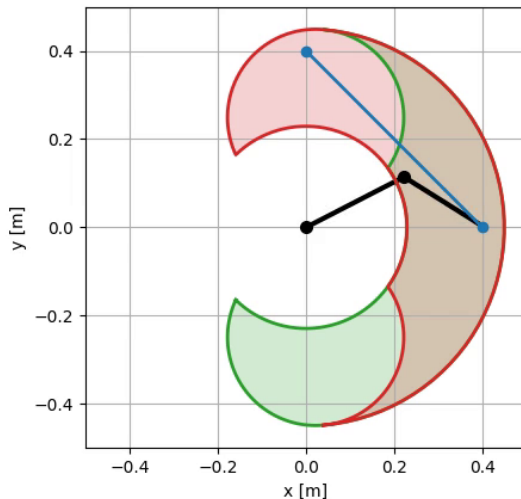
Interpolace v prostoru úloh



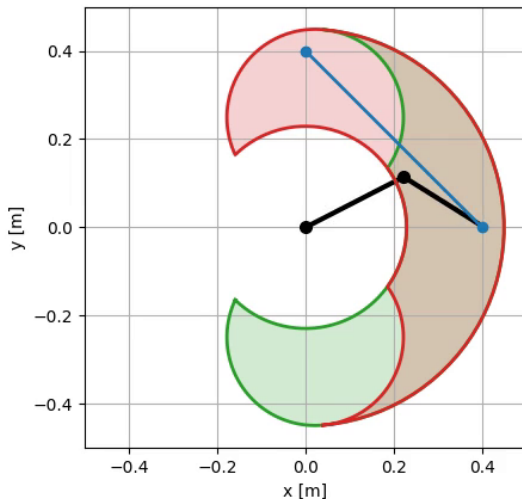
Interpolace v prostoru úloh



Interpolace v prostoru úloh



Interpolace v prostoru úloh

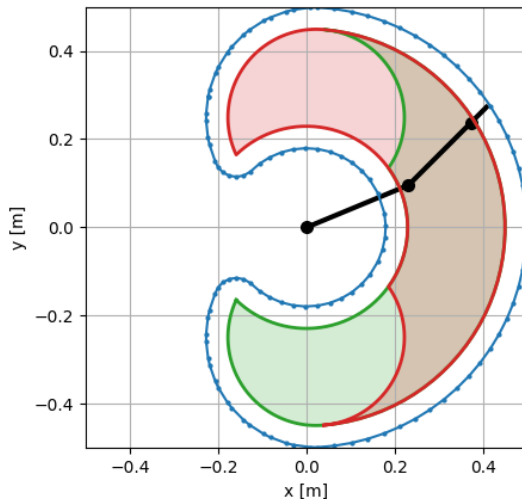


Interpolace v prostoru úloh

- ▶ Ne všechna řešení IK jsou dostupná všude
- ▶ Musíme vyřešit skoky v konfiguračním prostoru
- ▶ Pro změnu konfigurace musíme projít přes singularitu
- ▶ Interpolace v prostoru úloh může být použita pro cestu z před-uchopení do uchopení



SCARA efekt posledního ramene



Trajektorie z cesty

- ▶ Časové škálování $s(t)$, $t \in [0, T]$, $s : [0, T] \rightarrow [0, 1]$
- ▶ Cesta a časové škálování definuje trajektorii $\mathbf{q}(s(t))$
- ▶ Derivace:
 - ▶ rychlost: $\dot{\mathbf{q}} = \frac{d\mathbf{q}}{ds} \dot{s}$
 - ▶ zrychlení: $\ddot{\mathbf{q}} = \frac{d\mathbf{q}}{ds} \ddot{s} + \frac{d^2\mathbf{q}}{ds^2} \dot{s}$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}), \quad s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$



Škálování času přímé cesty

- ▶ Cesta

- ▶ pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$

- ▶ rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

- ▶ zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

- ▶ Časové škálování polynomem 3. řádu

- ▶ $s(t) = a_0 + a_1t + a_2t^2 + a_3t^3$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

► Časové škálování polynomem 3. řádu

- $s(t) = a_0 + a_1t + a_2t^2 + a_3t^3$
- $\dot{s}(t) = a_1 + 2a_2t + 3a_3t^2$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

► Časové škálování polynomem 3. řádu

- $s(t) = a_0 + a_1t + a_2t^2 + a_3t^3$
- $\dot{s}(t) = a_1 + 2a_2t + 3a_3t^2$
- omezení: $s(0) = \dot{s}(0) = 0$, $s(T) = 1$, $\dot{s}(T) = 0$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

► Časové škálování polynomem 3. řádu

- $s(t) = a_0 + a_1t + a_2t^2 + a_3t^3$
- $\dot{s}(t) = a_1 + 2a_2t + 3a_3t^2$
- omezení: $s(0) = \dot{s}(0) = 0$, $s(T) = 1$, $\dot{s}(T) = 0$
- řešení, které splňuje omezení: $a_0 = 0$, $a_1 = 0$, $a_2 = 3/T^2$, $a_3 = -2/T^3$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

► Časové škálování polynomem 3. řádu

- $s(t) = a_0 + a_1t + a_2t^2 + a_3t^3$
- $\dot{s}(t) = a_1 + 2a_2t + 3a_3t^2$
- omezení: $s(0) = \dot{s}(0) = 0$, $s(T) = 1$, $\dot{s}(T) = 0$
- řešení, které splňuje omezení: $a_0 = 0$, $a_1 = 0$, $a_2 = 3/T^2$, $a_3 = -2/T^3$

► Trajektorie

- $\mathbf{q}(t) = \mathbf{q}_{\text{start}} + \left(\frac{3t^2}{T^2} - \frac{2t^3}{T^3} \right) (\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- $\dot{\mathbf{q}}(t) = \left(\frac{6t}{T^2} - \frac{2t^2}{T^3} \right) (\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

► Časové škálování polynomem 3. řádu

- $s(t) = a_0 + a_1t + a_2t^2 + a_3t^3$
- $\dot{s}(t) = a_1 + 2a_2t + 3a_3t^2$
- omezení: $s(0) = \dot{s}(0) = 0$, $s(T) = 1$, $\dot{s}(T) = 0$
- řešení, které splňuje omezení: $a_0 = 0$, $a_1 = 0$, $a_2 = 3/T^2$, $a_3 = -2/T^3$

► Trajektorie

- $\mathbf{q}(t) = \mathbf{q}_{\text{start}} + \left(\frac{3t^2}{T^2} - \frac{2t^3}{T^3}\right)(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- $\dot{\mathbf{q}}(t) = \left(\frac{6t}{T^2} - \frac{2t^2}{T^3}\right)(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- $\ddot{\mathbf{q}}(t) = \left(\frac{6}{T^2} - \frac{12t}{T^3}\right)(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$



Škálování času přímé cesty

► Cesta

- pozice: $\mathbf{q}(s) = \mathbf{q}_{\text{start}} + s(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$, $s \in [0, 1]$
- rychlost: $\dot{\mathbf{q}} = \dot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- zrychlení: $\ddot{\mathbf{q}} = \ddot{s}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

► Časové škálování polynomem 3. řádu

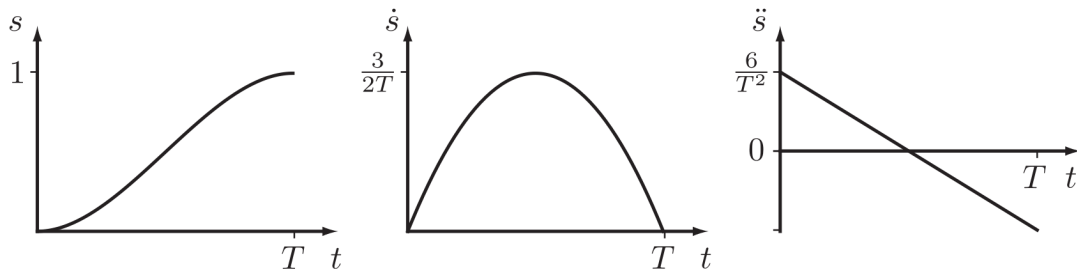
- $s(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$
- $\dot{s}(t) = a_1 + 2a_2 t + 3a_3 t^2$
- omezení: $s(0) = \dot{s}(0) = 0$, $s(T) = 1$, $\dot{s}(T) = 0$
- řešení, které splňuje omezení: $a_0 = 0$, $a_1 = 0$, $a_2 = 3/T^2$, $a_3 = -2/T^3$

► Trajektorie

- $\mathbf{q}(t) = \mathbf{q}_{\text{start}} + \left(\frac{3t^2}{T^2} - \frac{2t^3}{T^3} \right) (\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- $\dot{\mathbf{q}}(t) = \left(\frac{6t}{T^2} - \frac{2t^2}{T^3} \right) (\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- $\ddot{\mathbf{q}}(t) = \left(\frac{6}{T^2} - \frac{12t}{T^3} \right) (\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$



Časové škálování polynomem 3. řádu



Škálování času přímé cesty

- ▶ Maximální rychlosti v kloubech:

- ▶ $t = T/2$

- ▶ $\dot{\mathbf{q}}_{\max} = \frac{3}{2T}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$

- ▶ Maximální kloubové zrychlení:

- ▶ $t = 0$ a $t = T$

- ▶ $\ddot{\mathbf{q}}_{\max} = \left\| \frac{6}{T^2}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}) \right\|$

- ▶ $\ddot{\mathbf{q}}_{\min} = - \left\| \frac{6}{T^2}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}) \right\|$

- ▶ Jak tyto informace použít?



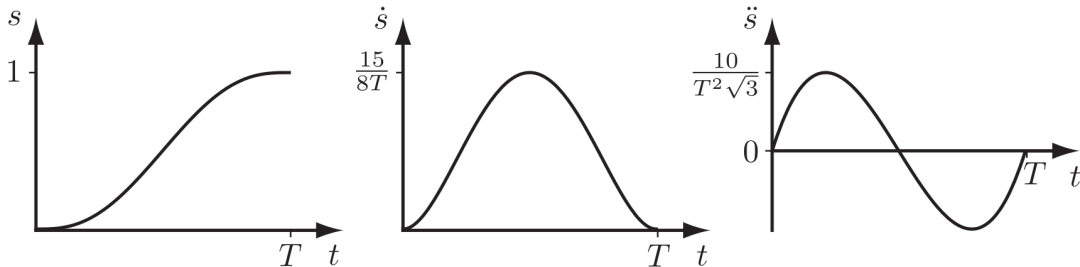
Škálování času přímé cesty

- ▶ Maximální rychlosti v kloubech:
 - ▶ $t = T/2$
 - ▶ $\dot{\mathbf{q}}_{\max} = \frac{3}{2T}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}})$
- ▶ Maximální kloubové zrychlení:
 - ▶ $t = 0$ a $t = T$
 - ▶ $\ddot{\mathbf{q}}_{\max} = \left\| \frac{6}{T^2}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}) \right\|$
 - ▶ $\ddot{\mathbf{q}}_{\min} = - \left\| \frac{6}{T^2}(\mathbf{q}_{\text{goal}} - \mathbf{q}_{\text{start}}) \right\|$
- ▶ Jak tyto informace použít?
 - ▶ zkontrolovat, zda je požadovaný pohyb T proveditelný vzhledem k limitům rychlosti/zrychlení
 - ▶ nalezení minimálního T tak, aby byla splněna omezení rychlosti a zrychlení



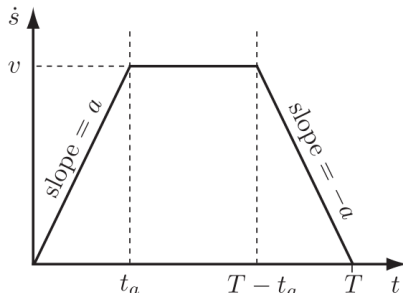
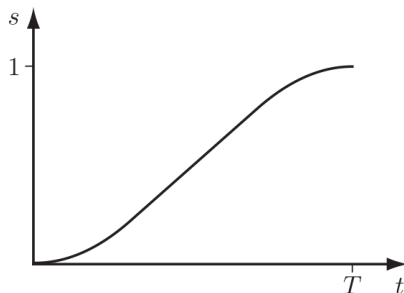
Polynom 5. řádu

- ▶ Polynom 3. řádu nevynucuje nulové zrychlení na začátku a na konci
 - ▶ nekonečný ryv [jerk] (derivace zrychlení)
 - ▶ může způsobit vibrace
- ▶ Můžeme použít polynom 5. řádu



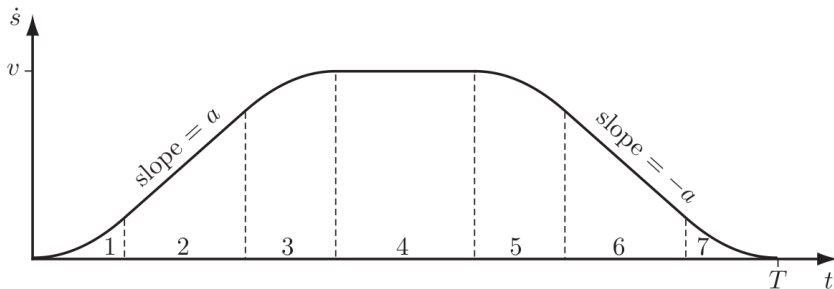
Trapezoidální časové škálování

- ▶ Fáze konstantního zrychlení
- ▶ Fáze konstantní rychlosti
- ▶ Fáze konstantního zpomalení
- ▶ Není hladký, ale je to nejrychlejší možný přímočarý pohyb



Časové škálování pomocí S-křivky

- ▶ Trapézové pohyby způsobují nespojité skoky ve zrychlení
- ▶ S-křivka jej vyhlazuje, aby se zabránilo vibracím
 - ▶ konstantní ryv, konstantní zrychlení, konstantní ryv, konstantní rychlost, konstantní ryv, konstantní zpomalení, konstantní ryv



TAMP: Plánování úloh a pohybu

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



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)**



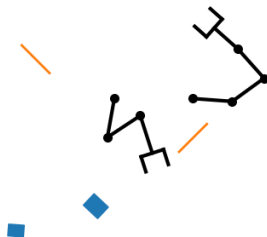
Motivace

- ▶ 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)



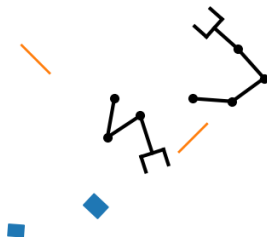
Konfigurační prostor

- ▶ Více chapadel připojených k robotům
- ▶ Povrchy v prostředí, na které lze objekt umístit
- ▶ Více objektů
 - ▶ více úchytů na objekt
 - ▶ více povrchů pro kontakt



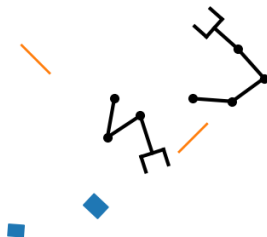
Konfigurační prostor

- ▶ 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



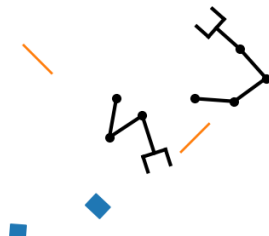
Konfigurační prostor

- ▶ 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



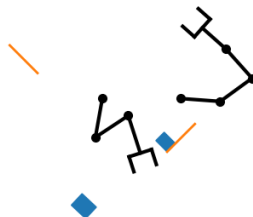
Omezení

- ▶ Objekt je umístěn nebo uchopen, tj. nemůže létat



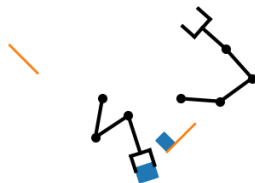
Omezení

- ▶ 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

- ▶ 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

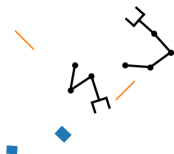


Stav (state)

- ▶ 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

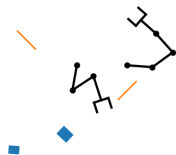


Vzorkování ze stavů



Vzorek z $\mathcal{C}$

Vzorkování ze stavů

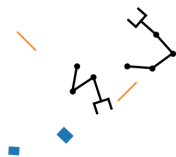


Vzorek z $\mathcal{C}$



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

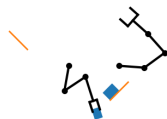
Vzorkování ze stavů



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



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



Identita na umístění

Interpolace na přechodu

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



Identita na umístění



Identita na uchopení

Interpolace na přechodu

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



Uchopení

Interpolace na přechodu

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



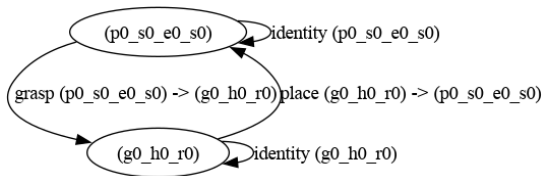
Uchopení



Umístění

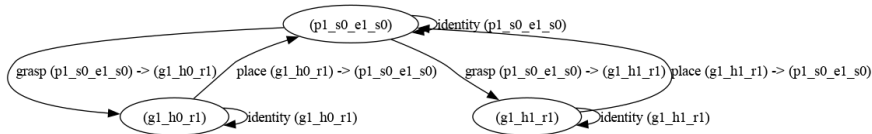
Graf omezení

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



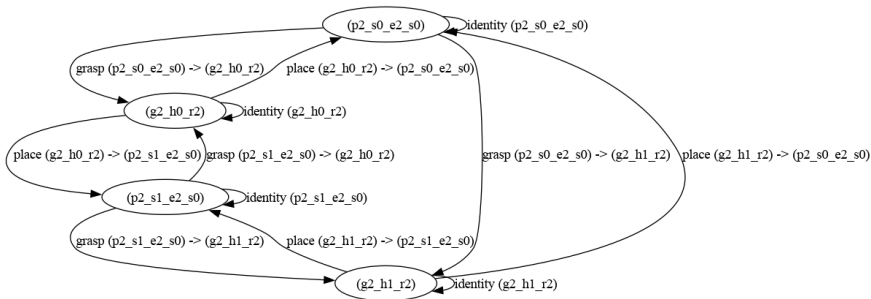
Graf omezení

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



Graf omezení

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

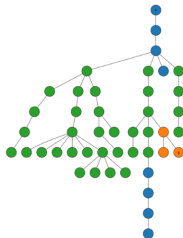
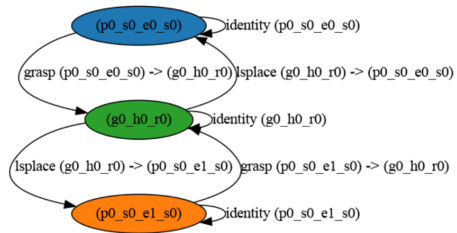


RRT na grafu omezení

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



RRT na grafu omezení

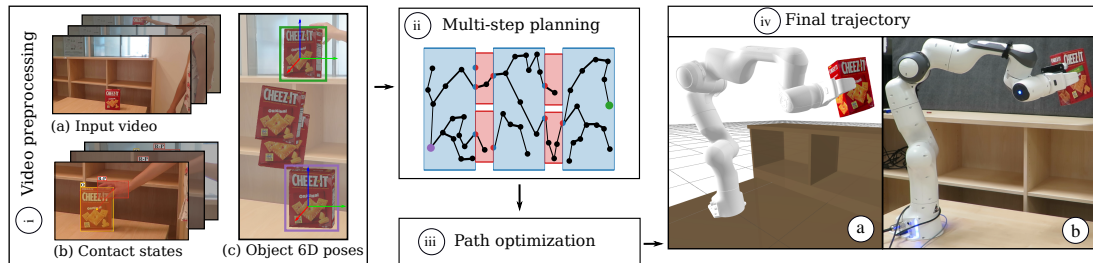


- ▶ Konfigurační prostor pro TAMP je komplexní
 - ▶ diskrétní množina stavů
 - ▶ spojitý pohyb
 - ▶ zakódovaný pomocí grafu omezení, který nám umožňuje použít RRT

- ▶ 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)

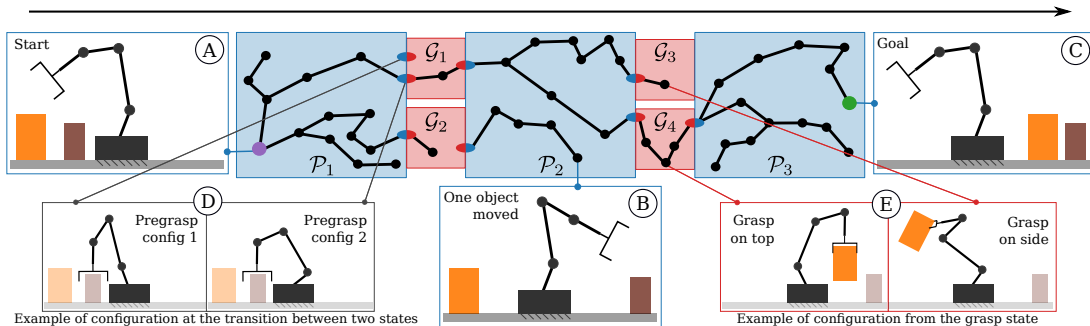
- ▶ Konfigurační prostor pro TAMP je komplexní
 - ▶ diskrétní množina stavů
 - ▶ spojitý pohyb
 - ▶ zakódovaný pomocí grafu omezení, který nám umožňuje použít RRT
- ▶ Obvykle se v průmyslu nepoužívá
 - ▶ posloupnost úkolů v prostoru je pevně zakódována programátory
 - ▶ plánovač je použit pouze pro pohyb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

TAMP řízený videem



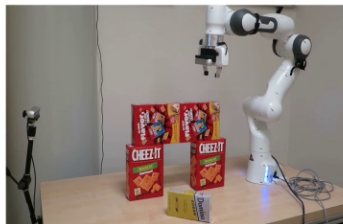
TAMP řízený videem

Time



Multi-Contact Task and Motion Planning Guided by Video Demonstration

Kateryna Zorina ♣ David Kovar ♣ Florent Lamiraux ◇ Nicolas Mansard ◇
Justin Carpentier ♥ Josef Sivic ♣ Vladimír Petřík ♣



♣ CIIRC, Czech Technical University in Prague
◇ LAAS-CNRS, Université de Toulouse, CNRS, Toulouse
♥ INRIA, Paris

Shrnutí

- ▶ Cesta/trajektorie
- ▶ Generování cesty k uchopení
- ▶ Interpolace v kloubovém prostoru a prostoru úloh
- ▶ Parametrizace časového škálování
- ▶ TAMP



Cvičení (KN)

- ▶ Poslední oraganizované cvičení
- ▶ Implementace PRM
- ▶ Domácí úkol: implementace RRT a Random shortcut [dobrovolné]

