



Robotika: Řízení robotů

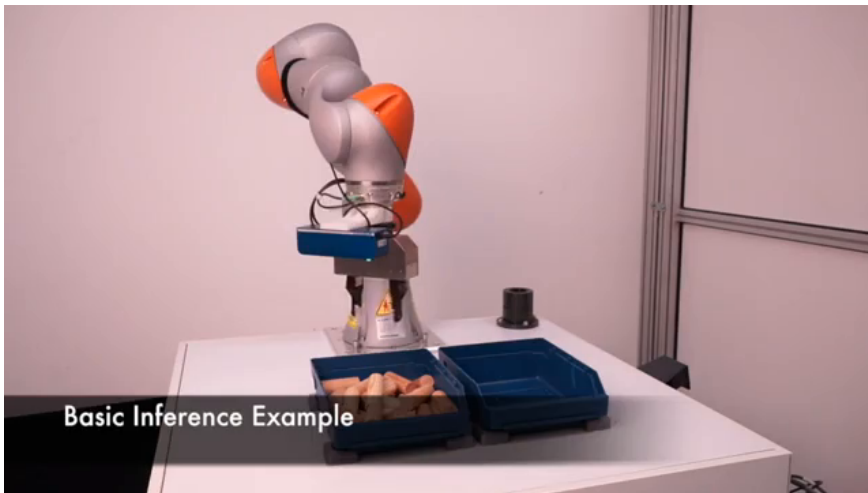
Vladimír Petřík

vladimir.petrík@cvut.cz

01.12.2025



¹Zdroj videa: https://www.youtube.com/watch?v=E_VrZvpvH5E



²Zdroj videa: <https://www.youtube.com/watch?v=dCChL1UiZy8>



³Zdroj videa: <https://www.youtube.com/watch?v=sR6oF2PpVno>

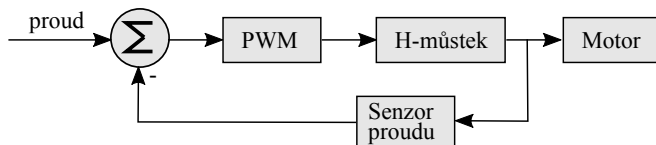
Cíle řízení

- ▶ Řízení pohybu
- ▶ Řízení síly
- ▶ Hybridní řízení



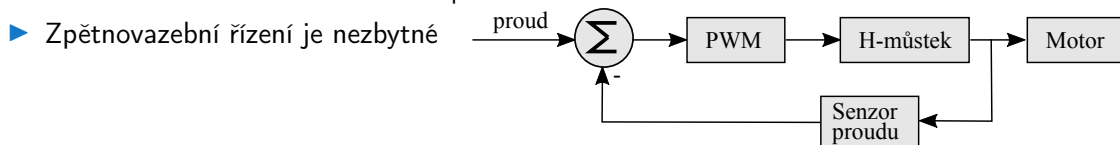
Jak motor ovládá kloub?

- ▶ Pro kvalitní motory platí vztah: $\tau_{\text{motor}} = k_T i$
 - ▶ τ_{motor} je moment vyvinutý motorem
 - ▶ k_T je momentová konstanta motoru (motor torque constant)
 - ▶ i je ovládaný proud
- ▶ Pokud můžeme ovládat proud, můžeme ovládat přímo moment/sílu motoru?



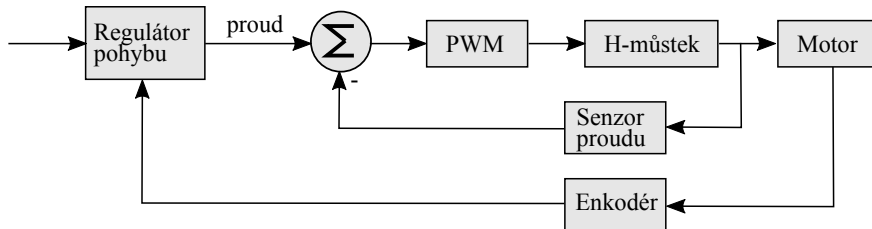
Jak motor ovládá kloub?

- ▶ Pro kvalitní motory platí vztah: $\tau_{\text{motor}} = k_T i$
 - ▶ τ_{motor} je moment vyvinutý motorem
 - ▶ k_T je momentová konstanta motoru (motor torque constant)
 - ▶ i je ovládaný proud
- ▶ Pokud můžeme ovládat proud, můžeme ovládat přímo moment/sílu motoru?
- ▶ Většinou **NE**
 - ▶ Většina robotů je poháněná malými motory, které jsou zpřevodované
 - ▶ Dynamické vlastnosti převodů jsou těžko simulovatelné (tření, vibrace)
 - ▶ Jednoduchý vztah mezi proudem a momentem pak neexistuje
 - ▶ moment roste monotónně s proudem



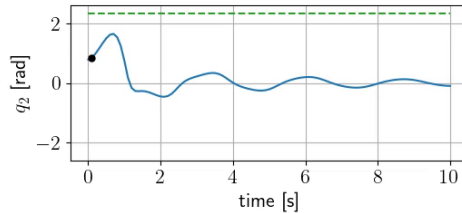
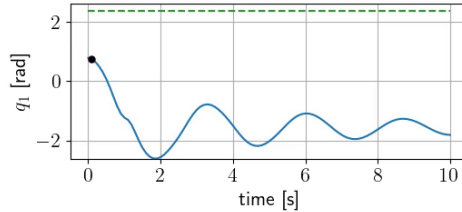
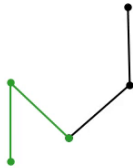
Zpětná vazba polohy

- ▶ Potřebujeme senzor: rotační enkodér
 - ▶ relativní rychlost $\dot{q}(t)$
 - ▶ absolutní poloha $q(t)$
 - ▶ levný, odolný a přesný senzor
- ▶ Zpětnovazební řízení
 - ▶ vstup: očekávaná/referenční trajektorie $q_r(t)$
 - ▶ kontrolní signál: proud motoru i
 - ▶ regulační odchylka: $q_e(t) = q_r(t) - q(t)$
 - ▶ cíl: nulová regulační odchylka



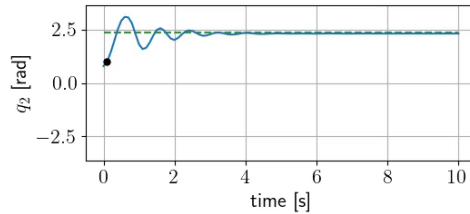
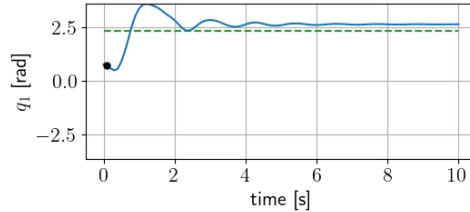
Řízení na polohu - nulový proud

► $i = 0$



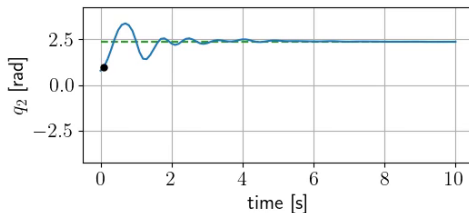
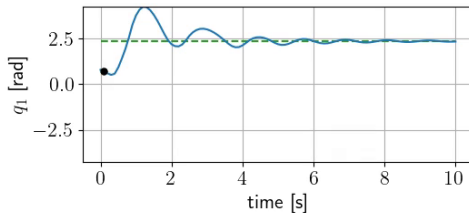
Řízení na polohu - P regulátor

► $\dot{q} = K_p q_e$



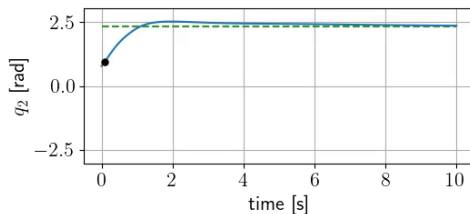
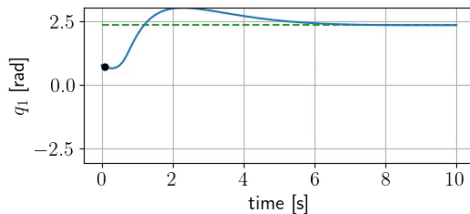
Řízení na polohu - PI regulátor

► $\dot{\mathbf{q}} = K_p \mathbf{q}_e + K_i \int \mathbf{q}_e$



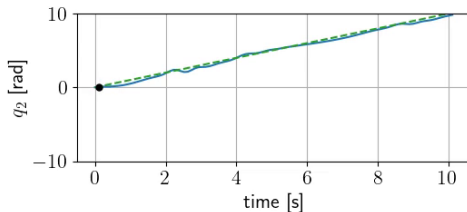
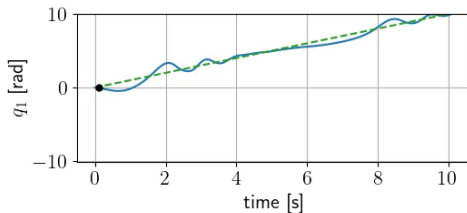
Řízení na polohu - PID regulátor

► $\dot{\mathbf{q}} = K_p \mathbf{q}_e + K_d \dot{\mathbf{q}}_e + K_i \int \mathbf{q}_e$



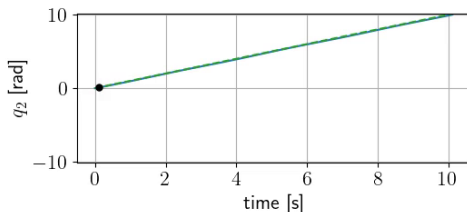
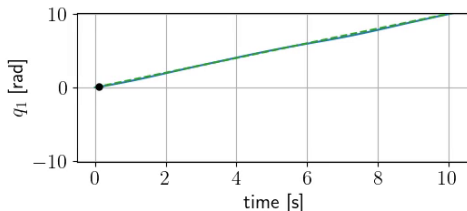
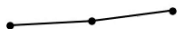
Řízení na polohu - sledování trajektorie P

► $\dot{\mathbf{q}} = K_p \mathbf{q}_e(t) = K_p (\mathbf{q}_r(t) - \mathbf{q}(t))$



Řízení na polohu - sledování trajektorie PID

► $i = K_p q_e(t) + K_d (\dot{q}_r(t) - \dot{q}(t)) + K_i \int q_e(t)$



- Integrovaná složka nezajistí nulovou regulační odchylku pro libovolnou trajektorii

PID řízení na polohu

► PID

- čeká na chybu, kterou pak kompenzuje
- díky I-složce dosáhneme nulovou regulační odchylku pro jednu polohu
- ne však pro libovolnou trajektorii



PID řízení na polohu

- ▶ PID
 - ▶ čeká na chybu, kterou pak kompenzuje
 - ▶ díky I-složce dosáhneme nulovou regulační odchylku pro jednu polohu
 - ▶ ne však pro libovolnou trajektorii
- ▶ Přidání modelu do kontrolní smyčky
 - ▶ lze dosáhnout lepších regulačních vlastností
 - ▶ pokud ovládáme proud, musíme identifikovat model
 - ▶ pokud ovládáme momenty, můžeme použít fyzikální model
 - ▶ většina průmyslových robotů neumí řídit momenty/síly
 - ▶ výrobce pro nás nastaví regulátor polohy
 - ▶ nelze použít pro spolupráci člověk/robot



Řízení momentů/sil

- ▶ Roboti řízení na momenty jsou stále častější
- ▶ Jak ale dosáhnout řízení momentů?
 - ▶ nepoužívat velké převody (vztah proudu a momentu pak bude jednodušší)
 - ▶ Barrett WAM (kabelový přenos sil/momentů)
 - ▶ Berkeley Blue (moderní motory v kloubech)



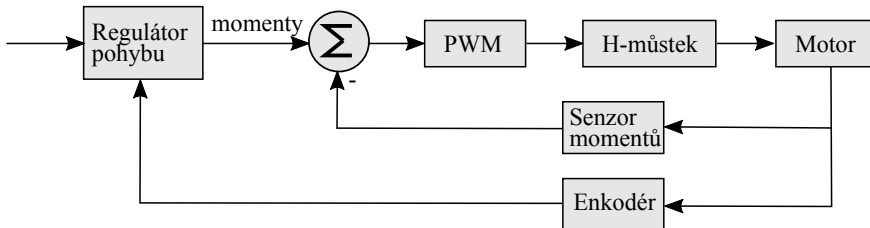
Řízení momentů/sil

- ▶ Roboti řízení na momenty jsou stále častější
- ▶ Jak ale dosáhnout řízení momentů?
 - ▶ nepoužívat velké převody (vztah proudu a momentu pak bude jednodušší)
 - ▶ Barrett WAM (kabelový přenos sil/momentů)
 - ▶ Berkeley Blue (moderní motory v kloubech)
 - ▶ dodatečný senzor na měření momentu
 - ▶ Kuka IIWA
 - ▶ Franka Panda Emika
 - ▶ Baxter



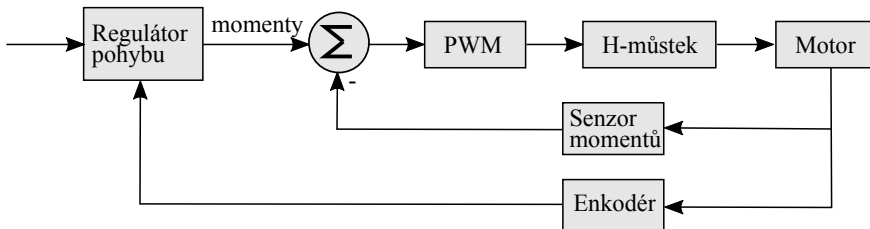
Řízení na polohu při silové zpětné vazbě

- ▶ Můžeme použít PID řízení stejně jako při řízení proudu
 - ▶ vstup: očekávaná/referenční trajektorie $\mathbf{q}_r(t)$
 - ▶ kontrolní signál: momenty motorů $\boldsymbol{\tau}$
 - ▶ regulační odchylka: $\mathbf{q}_e(t) = \mathbf{q}_r(t) - \mathbf{q}(t)$
 - ▶ cíl: nulová regulační odchylka



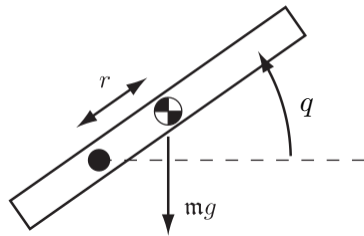
Řízení na polohu při silové zpětné vazbě

- ▶ Můžeme použít PID řízení stejně jako při řízení proudu
 - ▶ vstup: očekávaná/referenční trajektorie $\mathbf{q}_r(t)$
 - ▶ kontrolní signál: momenty motorů $\boldsymbol{\tau}$
 - ▶ regulační odchylka: $\mathbf{q}_e(t) = \mathbf{q}_r(t) - \mathbf{q}(t)$
 - ▶ cíl: nulová regulační odchylka
- ▶ Můžeme ale využít i dynamický model robota
 - ▶ dopředné řízení
 - ▶ reagujeme na chybu předtím, než nastane



Dynamický model - 1D⁴

- ▶ $\tau = M\ddot{q} + mgr \cos q$
 - ▶ M - setrvačnost kolem osy rotace
 - ▶ m - hmotnost
 - ▶ g - gravitační zrychlení
 - ▶ r - vzdálenost od osy rotace k těžišti
- ▶ Diferenciální rovnice pohybu: jaký moment musíme vyvinout, abychom docílili daného zrychlení
- ▶ Žádné tření - často ho aproximujeme pomocí $b\dot{q}$
- ▶ Rovnice se často vyjadřuje ve tvaru: $\tau = M\ddot{q} + h(q, \dot{q})$
- ▶ Funkce $h(\cdot)$ obsahuje vše, co nezávisí na zrychlení



⁴Obrázek z Lynch&Park: Modern Robotics

Dynamický model pro n stupňů volnosti

- ▶ $\tau = M(q)\ddot{q} + h(q, \dot{q})$
 - ▶ M - je matice $n \times n$ a závisí na konfiguraci robota
 - ▶ zrychlení jednoho kloubu závisí na pozici, rychlosti a momentu jiných kloubů
- ▶ Jak spočítat hodnoty parametrů rovnice pohybu?



Dynamický model pro n stupňů volnosti

- ▶ $\tau = M(q)\ddot{q} + h(q, \dot{q})$
 - ▶ M - je matice $n \times n$ a závisí na konfiguraci robota
 - ▶ zrychlení jednoho kloubu závisí na pozici, rychlosti a momentu jiných kloubů
- ▶ Jak spočítat hodnoty parametrů rovnice pohybu?
 - ▶ **Musíme znát dynamické vlastnosti jednotlivých ramen**
 - ▶ Newton-Eulerova formulace - efektivní numerický rekurzivní výpočet
 - ▶ Lagrangova formulace dynamiky - efektivní pro symbolické výpočty



Dynamické úlohy

- ▶ Přímá dynamická úloha
 - ▶ Cíl: najdi zrychlení $\ddot{\mathbf{q}}$
 - ▶ Vstup: $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}$
 - ▶ $\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q}) (\boldsymbol{\tau} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}))$
 - ▶ *Jak se bude robot pohybovat, pokud aplikujeme dané momenty?*



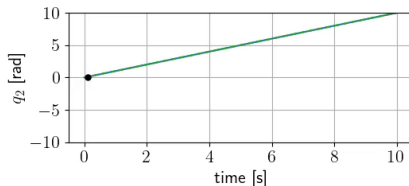
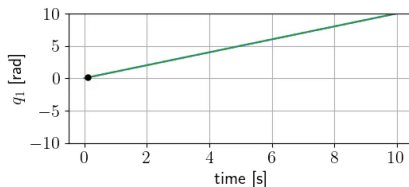
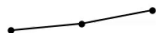
Dynamické úlohy

- ▶ Přímá dynamická úloha
 - ▶ Cíl: najdi zrychlení $\ddot{\mathbf{q}}$
 - ▶ Vstup: $\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}$
 - ▶ $\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q}) (\boldsymbol{\tau} - \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}))$
 - ▶ *Jak se bude robot pohybovat, pokud aplikujeme dané momenty?*
- ▶ Inverzní dynamická úloha
 - ▶ Cíl: najdi momenty $\boldsymbol{\tau}$
 - ▶ Vstup: $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$
 - ▶ $\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$
 - ▶ *Jaké momenty musíme aplikovat, abychom docílili danou trajektorii?*



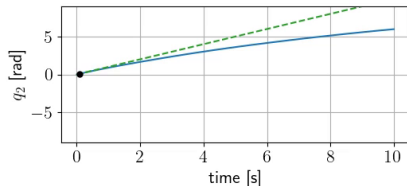
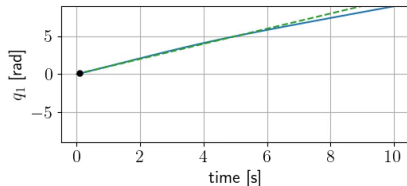
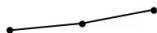
Jak můžeme využít dynamický model pro řízení?

- ▶ Dopředné řízení pro sledování trajektorie ($q, \dot{q}, \ddot{q}$)
- ▶ Vyřešíme inverzní dynamickou úlohu $\tau(t) = \hat{M}(q(t))\ddot{q}(t) + \hat{h}(q(t), \dot{q}(t))$
 - ▶ $\hat{M}(\cdot), \hat{h}$ jsou odhady skutečných parametrů
 - ▶ aplikováním momentů bude robot sledovat požadovanou trajektorii
 - ▶ pokud jsou naše odhady přesné
 - ▶ pokud je počáteční stav přesný



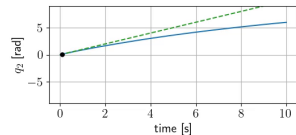
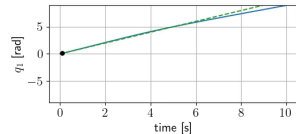
Jak můžeme využít dynamický model pro řízení?

- ▶ Dopředné řízení pro sledování trajektorie ($q, \dot{q}, \ddot{q}$)
- ▶ Vyřešíme inverzní dynamickou úlohu $\tau(t) = \hat{M}(q(t))\ddot{q}(t) + \hat{h}(q(t), \dot{q}(t))$
 - ▶ $\hat{M}(\cdot), \hat{h}$ jsou odhady skutečných parametrů
 - ▶ aplikováním momentů bude robot sledovat požadovanou trajektorii
 - ▶ pokud jsou naše odhady přesné, **chyba v aproximaci tření** $b = 2$ vs $b = 1.9$
 - ▶ pokud je počáteční stav přesný



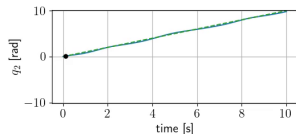
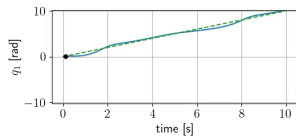
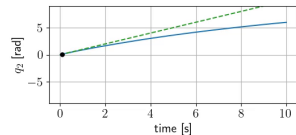
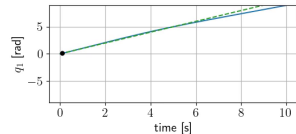
Dopředné × zpětnovazební řízení

- ▶ Dopředné řízení
 - ▶ model není přesný
 - ▶ dopředné řízení nereaguje na šum z prostředí
 - ▶ nečeká na chybu (generuje kontrolní signál s předstihem)



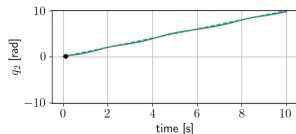
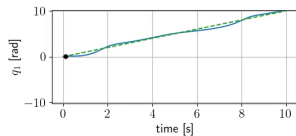
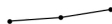
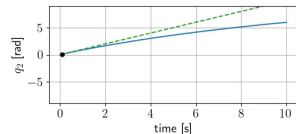
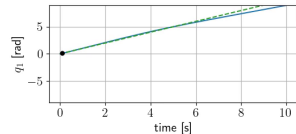
Dopředné × zpětnovazební řízení

- ▶ Dopředné řízení
 - ▶ model není přesný
 - ▶ dopředné řízení nereaguje na šum z prostředí
 - ▶ nečeká na chybu (generuje kontrolní signál s předstihem)
- ▶ Zpětnovazební řízení
 - ▶ začne reagovat až když vznikne chyba (požadovaný stav je jinde než současný)
 - ▶ reaguje na šum z prostředí



Dopředné × zpětnovazební řízení

- ▶ Dopředné řízení
 - ▶ model není přesný
 - ▶ dopředné řízení nereaguje na šum z prostředí
 - ▶ nečeká na chybu (generuje kontrolní signál s předstihem)
- ▶ Zpětnovazební řízení
 - ▶ začne reagovat až když vznikne chyba (požadovaný stav je jinde než současný)
 - ▶ reaguje na šum z prostředí
- ▶ Kombinace dopředného a zpětnovazebního řízení
 - ▶ dopředné pro včasnou kompenzaci (nelineární) dynamiky
 - ▶ zpětnovazební pro kompenzaci nemodelovaných jevů/nepřesností



Kombinace dopředného a zpětnovazebního řízení

- ▶ $\tau = \hat{M}(\mathbf{q}) (\ddot{\mathbf{q}}_r + K_p \mathbf{q}_e + K_d \dot{\mathbf{q}}_e) + \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$
 - ▶ $\mathbf{q}_e = \mathbf{q}_r - \mathbf{q}$
 - ▶ kompenzace nelineární dynamiky pomocí $\hat{\mathbf{h}}$
 - ▶ lineární dopředná složka $\hat{M}(\mathbf{q}) (\ddot{\mathbf{q}}_r)$
 - ▶ lineární zpětnovazební složka $\hat{M}(\mathbf{q}) (K_p \mathbf{q}_e + K_d \dot{\mathbf{q}}_e)$
 - ▶ K_p, K_d jsou často ve tvaru $k_p I, k_d I$
 - ▶ integrační složka je často vynechána

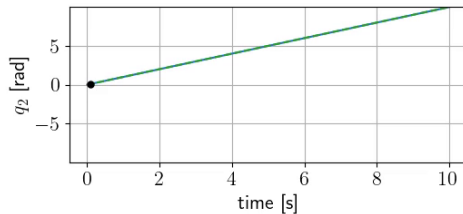
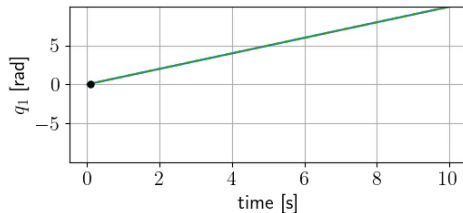
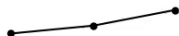


Kombinace dopředného a zpětnovazebního řízení

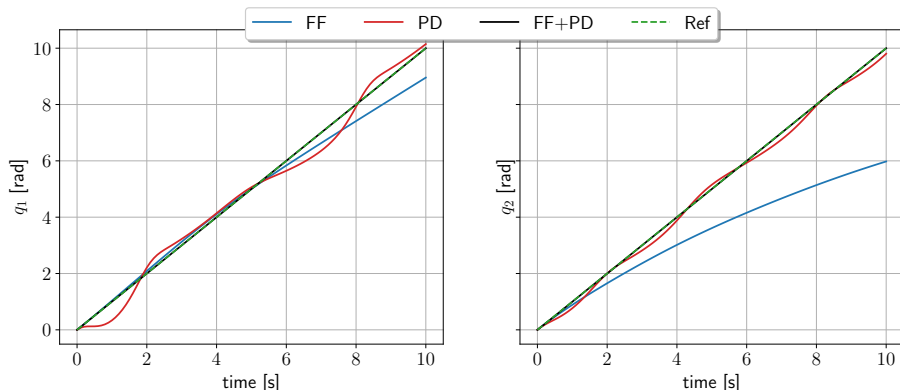
- ▶ $\tau = \hat{M}(\mathbf{q}) (\ddot{\mathbf{q}}_r + K_p \mathbf{q}_e + K_d \dot{\mathbf{q}}_e) + \hat{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}})$
 - ▶ $\mathbf{q}_e = \mathbf{q}_r - \mathbf{q}$
 - ▶ kompenzace nelineární dynamiky pomocí $\hat{\mathbf{h}}$
 - ▶ lineární dopředná složka $\hat{M}(\mathbf{q}) (\ddot{\mathbf{q}}_r)$
 - ▶ lineární zpětnovazební složka $\hat{M}(\mathbf{q}) (K_p \mathbf{q}_e + K_d \dot{\mathbf{q}}_e)$
 - ▶ K_p, K_d jsou často ve tvaru $k_p I, k_d I$
 - ▶ integrační složka je často vynechána
- ▶ Regulátor znám jako:
 - ▶ Feedforward plus feedback linearizing controller
 - ▶ Inverse dynamics controller
 - ▶ Computed torque controller



Kombinace dopředného a zpětnovazebního řízení

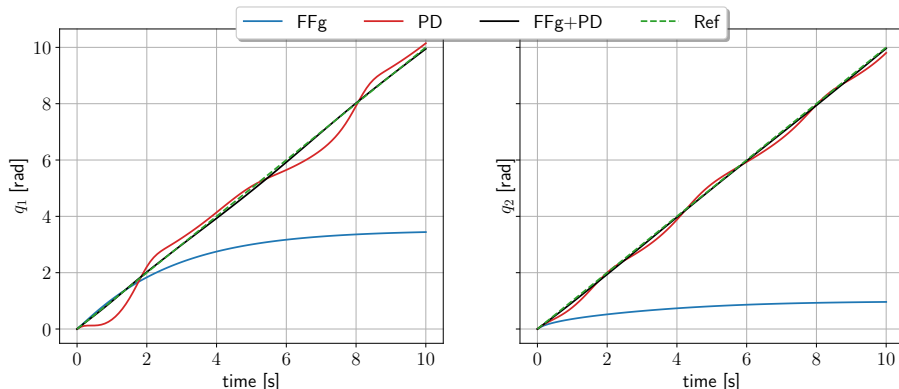


Kombinace dopředného a zpětnovazebního řízení



Zanedbání dynamiky vyšších řádů

- ▶ Často neznáme (všechny) dynamické parametry
- ▶ Robot se pohybuje pomalu a můžeme zanedbat dynamiku vyšších řádů
- ▶ $\tau = K_p q_e + K_d \dot{q}_e + \hat{g}(q)$
 - ▶ kompenzace gravitace $\hat{g}(q)$

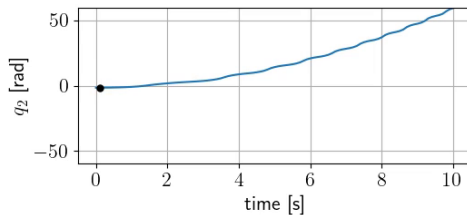
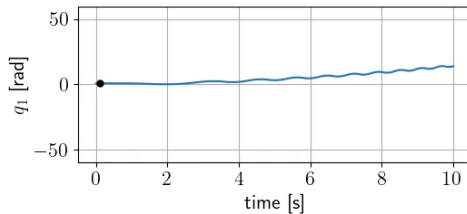


Řízení síly

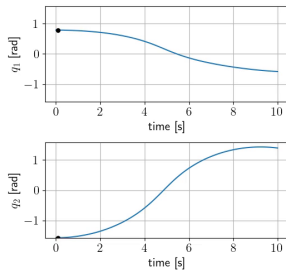
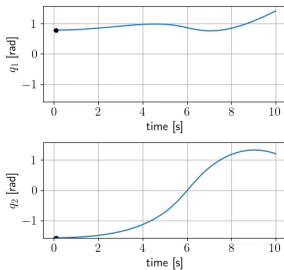
- ▶ Cílem je aplikovat sílu pomocí chapadla
 - ▶ tlačit do stolu
 - ▶ zavírat prsty chapadla
- ▶ Během aplikování síly se pohybujeme málo nebo vůbec → zanedbáme dynamiku vyšších řádů
- ▶ $\tau = \hat{g}(q) + J^T(q)\mathcal{F}_r$
 - ▶ J - jakobián
 - ▶ $\mathcal{F}_r$ - požadovaná síla/moment chapadla
- ▶ Co se stane s objektem na který aplikujeme konstantní sílu?



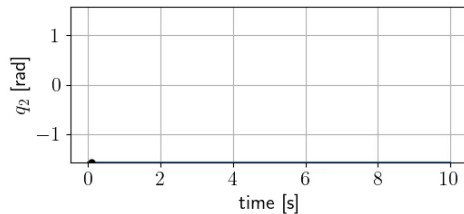
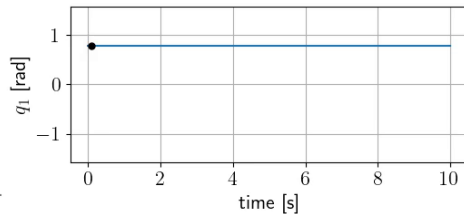
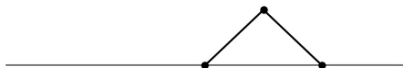
Konstantní moment chapadla



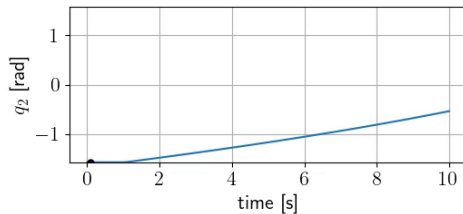
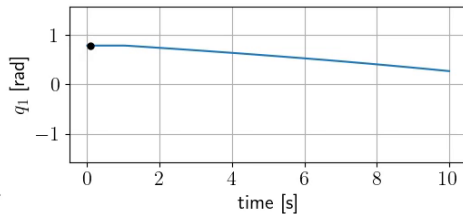
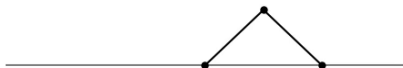
Konstantní síla chapadla



Tlačení proti stolu

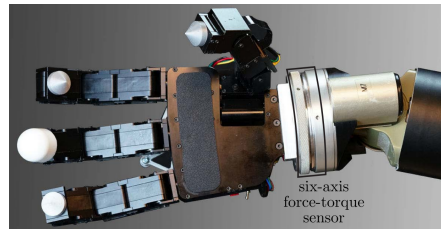


Tlačení proti stolu - šum

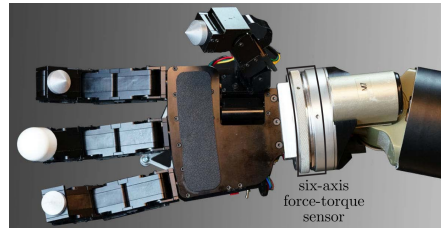


Dodatečný senzor

- ▶ Předchozí model vyžaduje
 - ▶ přesný model pro kompenzaci gravitace
 - ▶ přesný interní regulátor pro dosažení momentů
- ▶ Můžeme ale použít další zpětnou vazbu
- ▶ Externí senzor sil a momentů namontován mezi chapadlo a robota
- ▶ $\tau = \hat{g}(q) + J^T(q) (\mathcal{F}_r + K_{fp} \mathcal{F}_e)$



Dodatečný senzor



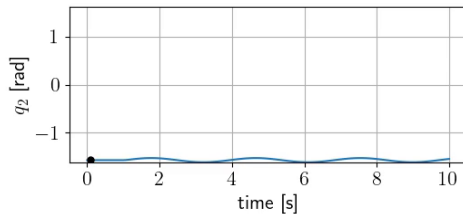
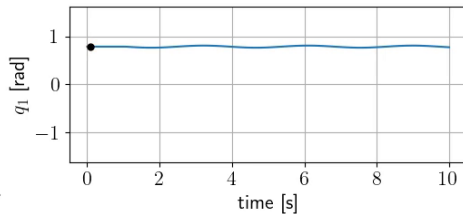
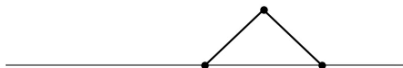
- ▶ Předchozí model vyžaduje
 - ▶ přesný model pro kompenzaci gravitace
 - ▶ přesný interní regulátor pro dosažení momentů
- ▶ Můžeme ale použít další zpětnou vazbu
- ▶ Externí senzor sil a momentů namontován mezi chapadlo a robota
- ▶ $\tau = \hat{g}(\mathbf{q}) + J^T(\mathbf{q}) (\mathcal{F}_r + K_{fp}\mathcal{F}_e)$
- ▶ Pomůže nám dodatečný senzor před posunem způsobeným šumem?

Hybridní řízení

- ▶ Některé stupně volnosti budeme řídit na polohu, některé na sílu
- ▶ Základní mechanické omezení: nemůžeme ovládat sílu i pohyb ve stejné ose zároveň
- ▶ $\tau = \hat{g}(q) + J^\top(q) (PK_p X_e + (I - P)\mathcal{F}_r)$
 - ▶ P - matice určující stupně volnosti ovládané na pohyb
 - ▶ např. $P_{2D} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
 - ▶ X_e - vzdálenost aktuální a referenční polohy v Kartézském prostoru
 - ▶ pozor na rotace, nemůžeme použít jenom odečítání
 - ▶ $D = X^{-1}X_r$
 - ▶ chyba polohy je v posledním sloupci matice D
 - ▶ chyba rotace se získá pomocí Rodriguesova vzorce z rotační matice transformace D
- ▶ Řízení se může rozšířit o PID, dodatečné znalosti kontaktů, externí senzor, ...

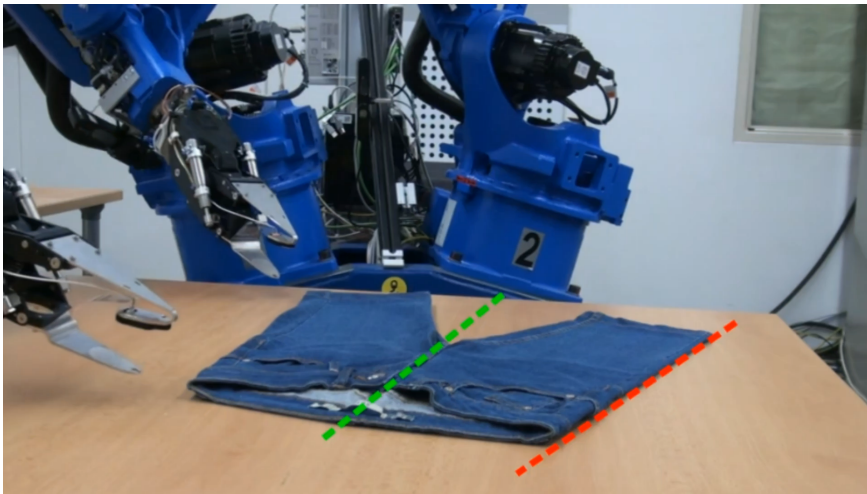


Hybridní řízení



Kontakt bez silového řízení

- ▶ Co když nemáme silové řízení a chceme být v kontaktu?
- ▶ Pasivní poddajnost - pružina



Optimální řízení (MPC)

- Řešení optimálního řízení

$$\begin{aligned} \arg \min_{\substack{\mathbf{u}_0, \dots, \mathbf{u}_{M-1} \\ \mathbf{x}_1, \dots, \mathbf{x}_M}} \quad & \sum_{i=0}^{M-1} l_i(\mathbf{x}_i, \mathbf{u}_i) + l_M(\mathbf{x}_M), \\ \text{s.t.} \quad & \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \forall i \in \{0, \dots, M-1\}, \\ & \mathbf{x}_0 = \hat{\mathbf{x}}. \end{aligned} \tag{1}$$

Optimální řízení (MPC)

- Řešení optimálního řízení

$$\begin{aligned} \arg \min_{\substack{\mathbf{u}_0, \dots, \mathbf{u}_{M-1} \\ \mathbf{x}_1, \dots, \mathbf{x}_M}} \quad & \sum_{i=0}^{M-1} l_i(\mathbf{x}_i, \mathbf{u}_i) + l_M(\mathbf{x}_M), \\ \text{s.t.} \quad & \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \forall i \in \{0, \dots, M-1\}, \\ & \mathbf{x}_0 = \hat{\mathbf{x}}. \end{aligned} \tag{1}$$

- Ricatti linearizace

$$\boldsymbol{\tau}(\mathbf{x}) = \boldsymbol{\tau}_0 + \mathbf{K}_0(\mathbf{x} - \mathbf{x}_0) \tag{2}$$

Optimalizační funkce

- Sledování polohy/orientace

$$\left\| \log \left((T_{\text{BC}}(\mathbf{q}_k) T_k)^{-1} T_{\text{BC}}(\mathbf{q}) T_{\text{ref}} \right) \right\|^2 \quad (3)$$



Optimalizační funkce

- ▶ Sledování polohy/orientace

$$\left\| \log \left((T_{\text{BC}}(\mathbf{q}_k) T_k)^{-1} T_{\text{BC}}(\mathbf{q}) T_{\text{ref}} \right) \right\|^2 \quad (3)$$

- ▶ Je řešení jednoznačné?



Optimalizační funkce

- ▶ Sledování polohy/orientace

$$\left\| \log \left((T_{\text{BC}}(\mathbf{q}_k) T_k)^{-1} T_{\text{BC}}(\mathbf{q}) T_{\text{ref}} \right) \right\|^2 \quad (3)$$

- ▶ Je řešení jednoznačné?
- ▶ Regularizace:

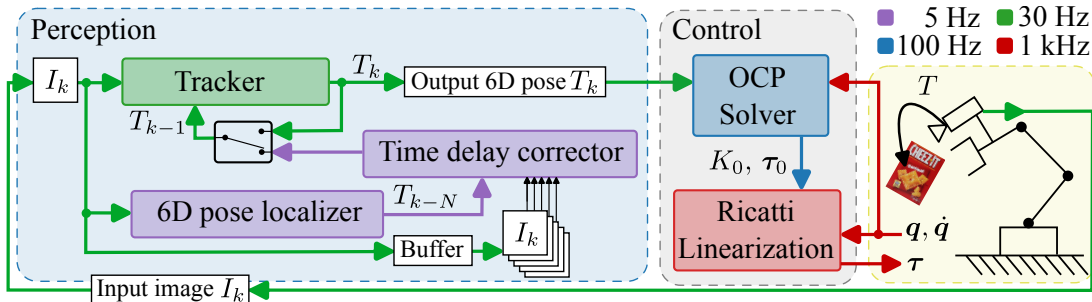
$$(\mathbf{x} - \mathbf{x}_{\text{rest}})^\top Q_x (\mathbf{x} - \mathbf{x}_{\text{rest}}) \quad (4)$$

$$(\mathbf{u} - \mathbf{u}_{\text{rest}}(\mathbf{x}))^\top Q_u (\mathbf{u} - \mathbf{u}_{\text{rest}}(\mathbf{x})) \quad (5)$$



Lokalizace a sledování objektu

- Kombinace pomalé lokalizace a rychlého sledovače
- Cíl: rychlá zpětná vazba pro řízení



Optimalizační řízení s řízením sledování

Static objects reaching

Scene cam:



Robot cam:



Run #1

Run #2

Run #3

Run #4

Závěr

- ▶ Existují různé cíle řízení
 - ▶ Řízení na polohu/trajektorii
 - ▶ Řízení síly
 - ▶ Hybridní řízení
 - ▶ Optimální řízení (MPC)
 - ▶ Neprobrané impedanční/admitanční řízení
- ▶ Pro každý stupeň volnosti můžeme ovládat sílu nebo pohyb
- ▶ Bez zpětné vazby bychom potřebovali dokonalý model
- ▶ Dopředný (dynamický) model zjednoduší řízení a zlepší kvalitu
 - ▶ Přímá dynamická úloha (spočti zrychlení)
 - ▶ Inverzní dynamická úloha (spočti momenty)
- ▶ Získat dynamické parametry není snadné
 - ▶ kompenzujeme alespoň gravitaci
 - ▶ pokud se nepohybujeme rychle, tak je kompenzace gravitace dostatečná

