

Robots

- manipulators
- mobile robots

základní pravidla:

- 1) Robot nesmí ublížit člověku nebo svou realitativní zprůčinnit zdanoten' újmou člověku.
- 2) Robot musí poslouchat lidské příkazy, mimo pravidlo 1.
- 3) Robot musí chránit svou existenci; dokud nepře v újímky 1 a 2.

Kinematika

- pohyb jednotlivých částí bez ohledu na hybridní síly
- Degree of Freedom (DOF)
 - 2D: 3 DOF (x, y, α)
 - 3D: 6 DOF $(x, y, z, \alpha, \beta, \gamma)$

→ pravidlo pravé ruky (palec je směr osy)
Tedy nejenom pozice, ale i orientace v lib. pozici
- Translations and rotations
- LCS, GCS — local/global coordinate system
- Wrench:
 - prizmatický, otáčivý, sférický, cylindrický, kulový

Rotace

Ve 2D: $\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \sim \begin{matrix} x' = x \cdot \cos \alpha - y \cdot \sin \alpha \\ y' = x \cdot \sin \alpha + y \cdot \cos \alpha \end{matrix}$

Ve 3D:

- podle osy x o α : $R_{x,\alpha} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix}$

$R_{y,\alpha} = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}$ *→ sinus inverted*

$R_{\alpha,\beta,\gamma} = R_{x,\alpha} \cdot R_{y,\beta} \cdot R_{z,\gamma}$

$R_{z,\alpha} = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Translate:

$$\begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translate + Rotate:

$$\begin{pmatrix} - & - & - & t_x \\ - & - & - & t_y \\ - & - & - & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Dennit-Hartenberg system

- řetězcí R, T, T, R pohyb Z může být dokončená spojitá

Vztah mezi LCS_{i-1} a LCS_i je transformace:

- 1) Nakloní osu x_{i-1} kolem osy z_{i-1} o úhel α_i .
- 2) Posune osu x_{i-1} ve směru osy z_{i-1} o vzdálenost d_i .
- 3) Posune osu z_{i-1} podél osy x_i o vzdálenost a_i .
- 4) Nakloní osu z_{i-1} kolem osy x_i o úhel β_i .

DH parametry: $(\alpha_i, d_i, a_i, \beta_i)$

Matrice:

$$1) \begin{pmatrix} \cos \alpha_i & -\sin \alpha_i & 0 & 0 \\ \sin \alpha_i & \cos \alpha_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad 2) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$3) \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad 4) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \beta_i & -\sin \beta_i & 0 \\ 0 & \sin \beta_i & \cos \beta_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

DH transformace $A_{i-1}^i = R_1 \cdot R_2 \cdot R_3 \cdot R_4$ s parametry $(\alpha_i, d_i, a_i, \beta_i)$

- vždy jen jeden parametr, ostatní konstanty (v jednotkových maticích)

DH konstanty:

- Osa z_{i-1} je osou kloubov i . $\rightarrow$ kloubový směr vede do kloubu konstanty zač! sys.
- Osa x_i je kolmá na z_{i-1}, z_i .
- Osa y_i doplňuje, aby šel o pravotočivý systém
- počátek LCS_i je umístěn do průsečíku z_{i-1}, z_i

Motion Model

Holonomie vs. Non-holonomie

$\hookrightarrow$ pokud $\#$ lokálních DOF = $\#$ globálních DOF

Ackermann steering $\rightarrow$ Non-holonomie

Změna orientace: $\Delta\theta = \frac{\Delta S}{r_s}$

Klancevrativnost závisí na rozvoru a maximálnímu zatáčení.

$r_s \rightarrow$ poloměr vodorovného kola, $\Delta S \rightarrow$ vzdálenost ved. kola

$\rightarrow$ ústřední křivka = $\theta \cdot r$, kde θ je v radiánech.

Differential steering $\rightarrow$ Non-holonomie

(aproximace) změna orientace: $\Delta\theta = \frac{S_n}{r_n}$

$$\frac{\frac{\Delta R + \Delta L}{2}}{\frac{b}{2} \cdot \frac{\Delta R + \Delta L}{\Delta R - \Delta L}}$$

$\rightarrow$ průměrná úhlová rychlost

$$\frac{\Delta R - \Delta L}{b}$$

$\rightarrow$ jedno kolo

levo šiklo

$$= \frac{\Delta R + \Delta L}{2} \cdot \frac{b}{\Delta R - \Delta L}$$

$$x_n = x_{n-1} + S_n \cdot [\cos(\Delta\theta + \theta_{n-1})]$$

$$y_n = y_{n-1} + S_n \cdot [\sin(\Delta\theta + \theta_{n-1})]$$

Omnidrive steering $\rightarrow$ Holonomie

$$\vec{v} = \vec{v}_t + \vec{\omega} \times \vec{r}$$

$\rightarrow$ vektorové posunutí k bodu

$$v_x = v_{tx} - \omega \cdot r_y$$

$$v_y = v_{ty} + \omega \cdot r_x$$

Motors:

DC:

- kartáčové, bezkartáčové

AC:

- synchronní, asynchronní

Piezo, ultrazvukový

Ampérovo pr. pravidlo:

- prst ve směru proudu

- palec ukazuje severní pól

Flamengovo pravidlo levé ruky

- ukazováček je mag. pole

- prostředníček je proud

- palec je směr pohybu

Regulace rychlosti:

- PWM $\rightarrow$ rychlé ON/OFF přepínání
- Frekvenční modulace u AC

Bezkontaktní:

- podle natočení rotorem se zapínají a vypínají jednotlivé diody uvnitř

Uhlíkový motor

- zubová konstrukce rotoru, stejné tak uvnitř statoru, dozdívá tak posunutí o jeden „zob“

Servo:

- nastavuje se přesná poloha natočení osy
- min. délka pulsu: 1ms
- max. délka pulsu: 2ms
- Stop pozice: 1,5ms

Senzory:

- aktivní vs. pasivní detekce
- detekce vs. měření
- aktivní vs. pasivní
- signal-reflection measurement
 - aktivní, musí se odrazit
- signal interruption measurement
 - klasický optický gate
- infrared
 - určitá vzdálenost
 - emitér, detektor

Odobratice

- BW terče $\rightarrow$ rozdělení na více steps pro určení směru / indikace směru na místě.
- Hallan sondu $\rightarrow$ detekce reaguje na mag. pole, přestupuje náhle na jeden stav
 - $\hookrightarrow$ používá se například v autě

PID:

proportional P-gain : $P \cdot e_n$

integral I - reset : $I \cdot \sum e_i$

derivative D - rate : $D \cdot \Delta e_n$

$$e = \text{req-speed} - \text{curr-speed}$$

$$\text{sum-e} += e$$

$$\text{control} = P \cdot e + I \cdot \text{sum-e} + D \cdot (\text{curr} - \text{last-speed})$$

Jak nastavit? P zvyšovat, dokud neoscillují. Pak snížit na polku
a přidat I, dokud se zvolená rychlost nestaví příliš pomalu.
D doinstalovat na vybranou odezvu

↳ Často se nepoužívá full PID, ale jenom jeho část: P, PD, ...

Control systems

- klasický automat $A = (Q, \Sigma, \delta, q_0, F)$

Petri net

- independent events sync
- waiting for release of other part

SPA

- sense, plan, act

Reactive

- just act

Hybrid control

- think and act

↳ 3 vrstvy: plánování, koordinace, execution

Behaviour control

- think the way you act

Environment repr.

- rozdělení podle typu
 - kdy byl vytvářen (online/offline)
 - manual/automatic
 - topografický / "surovinný"
- Robot často používá metrickou mapu + objektivitu mapy

Localizace:

- problém synchronizace LCS m GSC
- 1) Absolutní localizace bez zohlednění minulého stavu
- 1) Relativní localizace vzhledem k předchozímu stavu
- 2) Pasivní localizace (robot do toho nemůže vstupovat)
- 2) Aktivní localizace (aktivní může být aktivně)
- 3) Localizace ve statickém prostředí
- 3) Localizace v dynamickém prostředí

Řešení:

- co nej přesnější měření
- udělat z toho diskrétní prostor a diskrétní pohyb
- využít více informačních zdrojů.

Pravděpodobnostní pole:

$Bel(l)$ je šance, že jsem na pozici $l := B_{j,l}$

Topologický graf

- vrchol je pozice, hrana je možná cesta.
- $Bel(l)$ je šance, že jsem ve stavu l .

Monte Carlo Localization

- Prediction step
- Correction step
- Oversampling step

Plánování

Bug algo 1:

— = incomplete

- jde ve směru cíle
- přetáhne obcházet zlem, dokud nepřijde k cíli

Bug algo 2:

- jde ve směru cíle —> vždycky zhrnuje, bezpečný, koncový
- přetáhne jednou obíhají a pak už nepřijde k cíli

Bug algo 3:

- jde ve směru cíle —> koncový, lepší (většinou) jako Bug 2
- přetáhne znovu obcházet, jakmile narazí na start - goal čísl, jde po ní. —> zhrnuje a hrádou přetáhne potáhá jen $\frac{p}{2}$, kde p je počet minut čísl start - goal

Bug algo 4: (Tangent bug)

Dijkstra

- hlásí princip, všechny body mají default cost $+\infty$, postupně zhrnuje, jde po nejmenší.
- $O(V^2)$

Rapidly Exploring Random Trees

- počet iterací závisí na přesnosti, vždy se mění "větví" řešení.

Multirobot system

- globalizovaný / lokální / semi globalizovaný
- ...

GPS

- historie

- Transit
- Timation
- G2113
- Navstar GPS

GPS, Glonass, Galileo —> globální
Beidou, Doris