Úvod do mobilní robotiky — NAIL028

Martin Dlouhý

md at robotika.cz

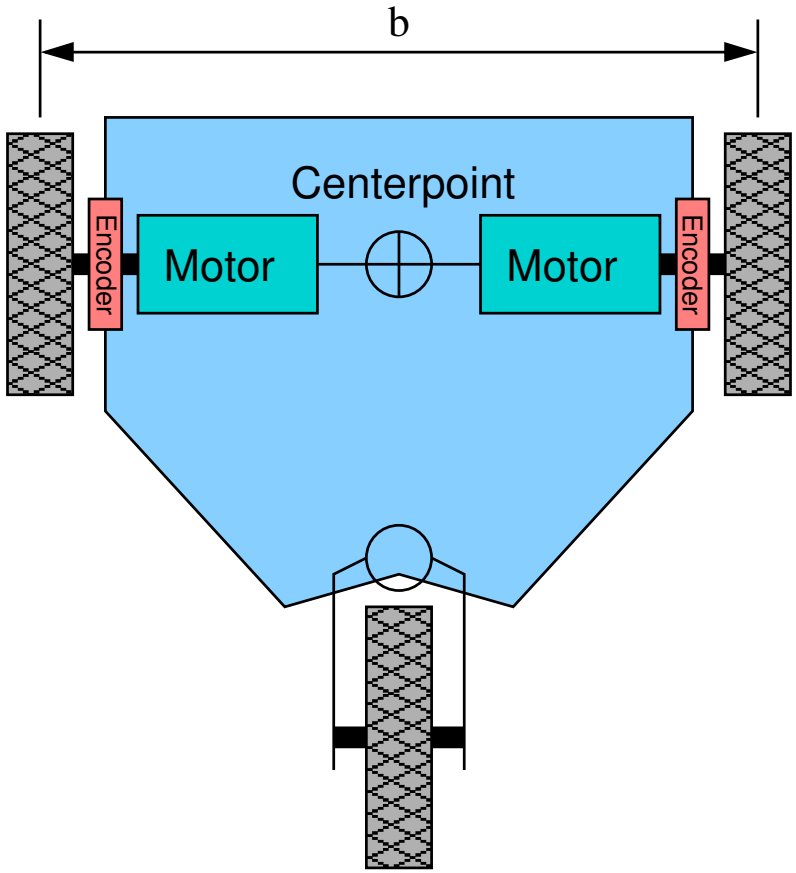
http://robotika.cz/guide/umor08/cs

11. listopadu 2008

1. Lokalizace
   - Modely vozítek
   - První pokus o lokalizaci
   - Monte Carlo Localisation (MCL)
2. Řízení
   - PID
   - Sledování cesty

Modely kolových vozidel (1/5)

Diferenční řízení – tank



Modely kolových vozidel (2/5)

Diferenční řízení – výpočet

- ujetá vzdálenost pravým/levým kolem za daný interval

$$\Delta U_R, \Delta U_L$$

- změna pozice robota v robocentrických souřadnicích

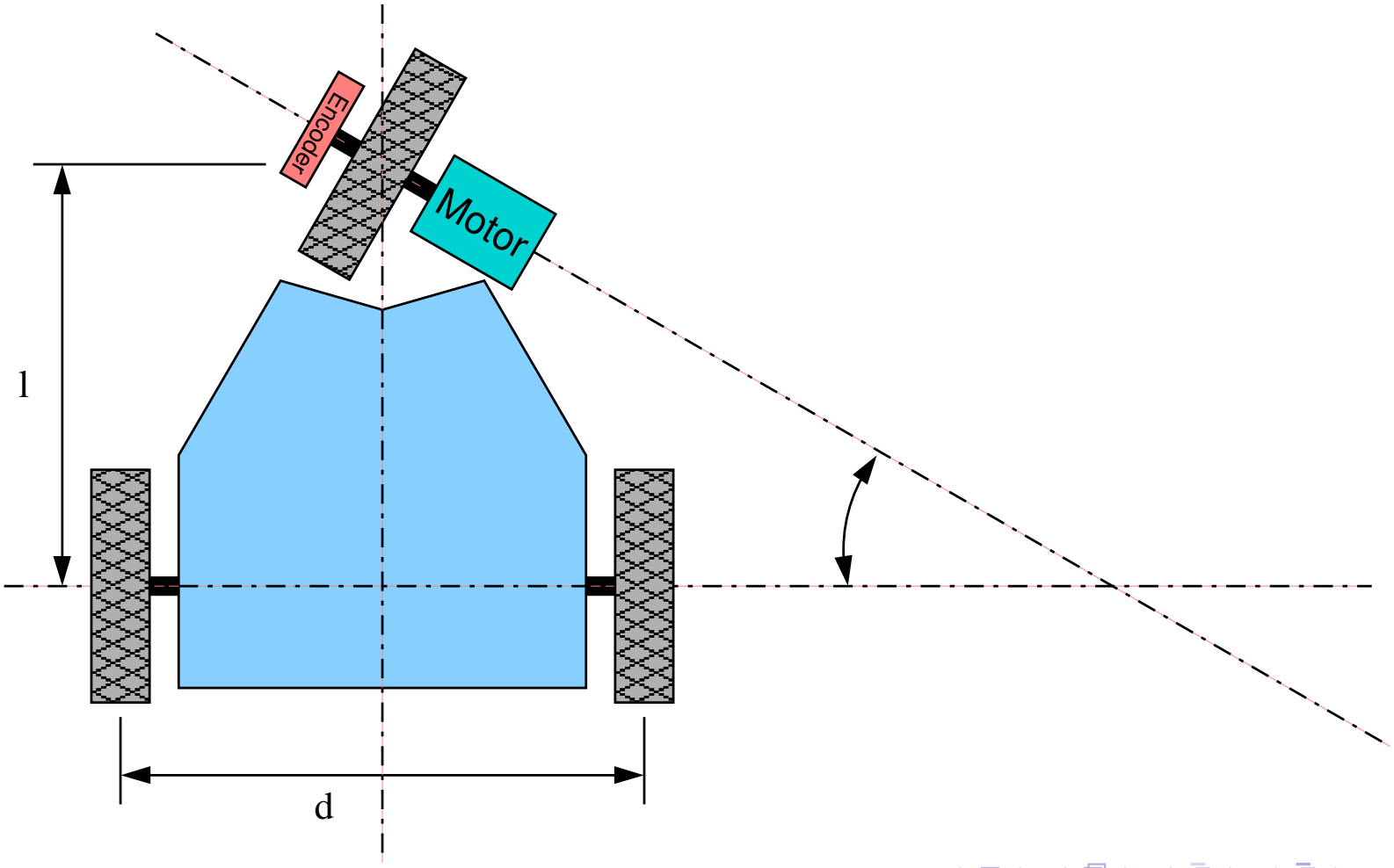
$$\begin{aligned} \Delta U_F &= (\Delta U_R + \Delta U_L)/2 \\ \Delta\theta &= (\Delta U_R - \Delta U_L)/S_{wheel\ \ base} \end{aligned}$$

- změna pozice robota v globálních souřadnicích

$$\begin{aligned} \theta_i &= \theta_{i-1} + \Delta\theta \\ x_i &= x_{i-1} + \Delta U_F \cos(\theta_i) \\ y_i &= y_{i-1} + \Delta U_F \sin(\theta_i) \end{aligned}$$

Modely kolových vozidel (3/5)

Tříkolka (zatáčení vpředu/vzadu)



Modely kolových vozidel (4/5)

Tříkolka — výpočet

- Pohyb po kružnici, střed definován
  - natočení řídícího kola α
  - vzdáleností náprav l
  - ujetá vzdálenost řídícího kola U
- změna pozice robota v globálních souřadnicích

$$\begin{aligned} \theta_i &= \theta_{i-1} + \Delta U \sin(\alpha)/l \\ x_i &= x_{i-1} + \Delta U \cos(\alpha)\cos(\theta_i) \\ y_i &= y_{i-1} + \Delta U \cos(\alpha)\sin(\theta_i) \end{aligned}$$

Modely kolových vozidel (5/5)

Akckermanovo řízení pro 4 kola

- splňuje $\cot\theta_i - \cot\theta_o = \frac{d}{l}$
  - θ_i – natočení vnitřního kola
  - θ_o – natočení vnějšího kola
  - l – vzdálenost náprav (délka)
  - d – vzdálenost kol (šířka)

První pokus o lokalizaci

- Enkodéry i akcelerometry jsou zdrojem pouze relativní informace.
- Co se stane, když budeme delší dobu uplatňovat následující rovnice?

$$\begin{aligned} \theta_i &= \theta_{i-1} + \Delta\theta \\ x_i &= x_{i-1} + \Delta U_F \cos(\theta_i) \\ y_i &= y_{i-1} + \Delta U_F \sin(\theta_i) \end{aligned}$$

- Divergence :-(.
- Proč?

Řešení problémů prvního pokusu

- Modelovat nepřesnosti používaných senzorů
  - Kolečka prokluzují
  - Akcelerometry šumí
- Používat i informace globálního charakteru
  - Měřit vzdálenosti ke známým předmětům (mapa, GPS)
  - Měřit orientace (kompas, majáky)
- Naštěstí nejsme sami, kdo má tento problém :-)

Připomenutí Kalmanova filtru (1/2)

Stavová rovnice

$$x_{k+1} = Fx_k + w_k$$

kde x_k je n dimenzionální stavový vektor, F je transformační matice a w_k šum/chyba stavu ($w_k \sim N(0, Q)$)

Rovnice měření

$$z_k = Hx_k + v_k$$

kde z_k je m dimenzionální vektor měření, H je $m \times n$ matice určující vztah mezi stavem a měřením a v_k je šum/chyba měření ($v_k \sim N(0, R)$)

Připomenutí Kalmanova filtru (2/2)

Predikce stavu a chyby — pomocí stavové rovnice

$$\begin{aligned} x_{k+1}^{-} &= Fx_k \\ P_{k+1}^{-} &= FP_kF^T + Q \end{aligned}$$

Korekce pomocí měření — pomocí rovnice měření

$$\begin{aligned} z_k &= Hx_k + v_k \\ K_k &= P_k^{-}H^T(HP_k^{-}H^T + R)^{-1} \\ x_k &= x_k^{-} + K_k(z_k + Hx_k^{-}) \\ P_k &= (I - K_kH)P_k^{-} \end{aligned}$$

Monte Carlo Lokalizace

- Pozice reprezentovaná množinou vážených vzorků
- 3 fáze
  - pohyb (predikce)
  - měření (korekce)
  - převzorkování
- Jednoduché ohodnocovací funkce:
  `double eval(Pozice p);`
- Zásadní výhoda - jednoduchá implementace

Vzorky

- Prvky konfiguračního prostoru
  - pro robota 2D s otáčením to je (x_i, y_i, θ_i)
- Každý má nějakou váhu důležitosti
- Množství vzorků závisí na složitosti prostředí a výpočetní síle
- Na startu rovnoměrné pokrytí prostoru (globální lokalizace) nebo v okolí známého startu (tracking)

Model pohybu/predikce

- Model popisuje pravděpodobnostní přechody mezi jednotlivými body konfiguracemi
- Každý vzorek se počítá nezávisle
- Triviální modely, kdy se pouze „zašumí“ vstupy (např. data z enkoderů)
- Pokud by na začátku byly všechny vzorky stejné, tak po predikci dostaneme pravděpodobnostní model

Měření/korekce

- Ohodnocuje pravděpodobnost pozice/vzorku pro dané měření
- Mění váhu vzorku (nějakým násobícím faktorem > 0)
- Každý vzorek je zpracováván nezávisle

Převzorkovaní

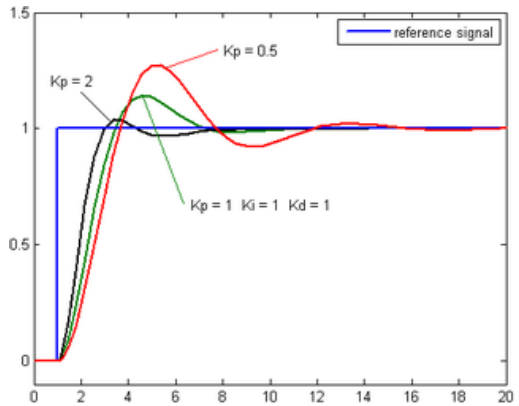
- Zajišťuje rovnoměrnost vah vzorků
- Na výstupu všechny váhy stejné, tj. některé vzorky zaniknou a jiné se znásobí
- Pravděpodobnost výskytu vzorku ve výsledné množině je přímo úměrná jeho váze
- $O(n)$ algoritmus — poskládá normalizované váhy za sebe, náhodně zvolí hodnotu z intervalu $< 0, 1)$ a přičítáním 1 dostane n nových vzorků

Slabiny MCL

- Není jedna pozice, tak podle čeho má robot jet?
- Možnost skokových změn
- Problém nastavení parametrů a ladění
- Manévry nezachycené v distribuční funkci (čelní náraz)

PID controller (1/2)

- Proportional–Integral–Derivative controller
- „Univerzální“ metoda řízení
- Cíl dosáhnout požadované hodnoty (setpoint) pomocí řízené hodnoty (process variable)



PID controller (2/2)

Proportional — rozdíl mezi aktuální a požadovanou hodnotou

$$K_P e(t)$$

Integral — součet proporcionálních rozdílů

$$K_I \int_0^t e(\tau) d\tau$$

Derivative — rozdíl chyby

$$K_D \frac{de}{dt}$$

PID příklad

- Řízení rychlosti motoru
  - Vstup aktuální rychlost z enkodérů
  - Ovládání pomocí PWM (Pulse Width Modulation)

Úloha sledování cesty

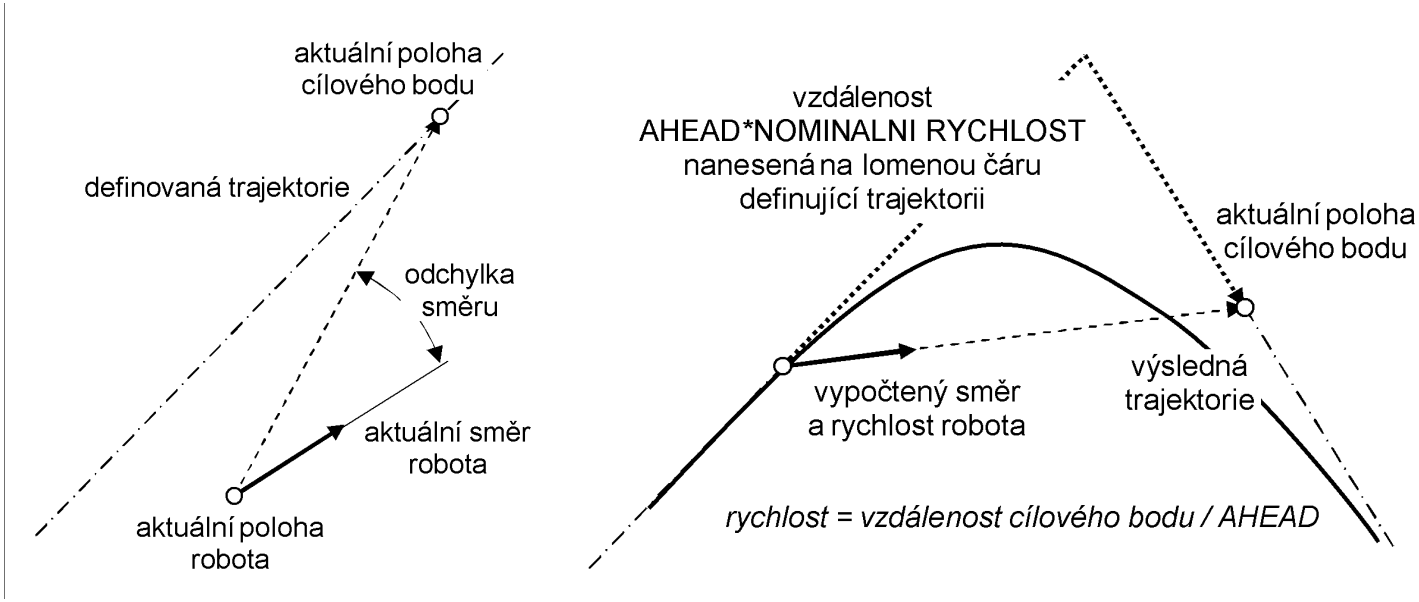
- Jak vhodně cestu popsat?
  - lomená čára
  - lomená čára s úseky kružnic
  - spline křivky

Sledování cesty ver0

- Dostatečná je lomená čára, případně se šířkou koridoru
- Verze 0 — jeď rovně do bodu zlomu, zastav, otoč se na místě o požadovaný úhel, opakuj
- Většinou stačí mírná korekce po malou chybu a v případě velké chyby zastavit a směrovat robota na cílový bod

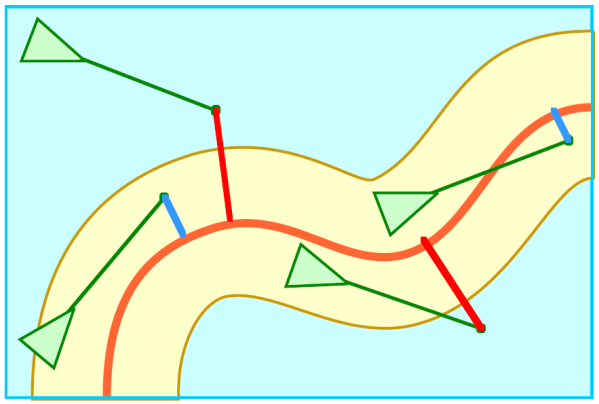
Sledování cesty pomocí PID (R-team)

- Cílová pozice v každém kroku předbíha robota o určitou vzdálenost
- Dva proporcionální regulátory: regulace směru a výpočet požadované rychlosti robota



Sledování cesty — Steering behaviour

- Craig W. Reynolds: Steering Behaviors For Autonomous Characters
- Pohyb v koridoru definovaném lomenou čarou a poloměrem kružnice r
- Predikce pozice s projekcí na nejbližší bod na křivce
- Je-li vzdálenost menší než r neprovádí se žádá korekce, v opačném případě se přepne do módu seek na projekci na cestě



Příště

- Global Position System (GPS)