EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Metody lokalizace v mobilní robotice

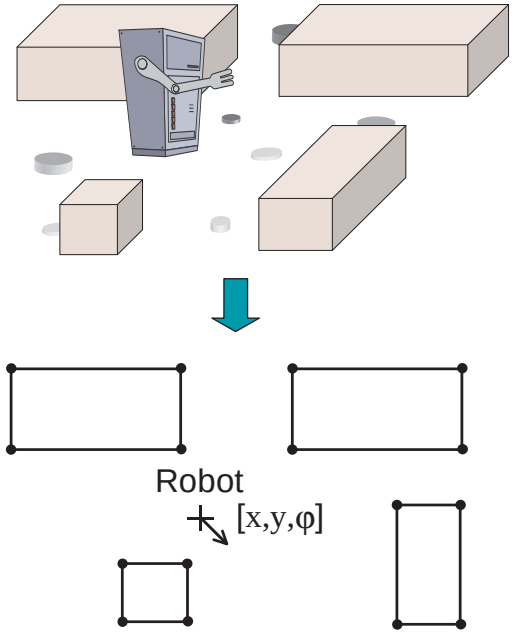
RNDr. Miroslav Kulich, Ph.D. (ČVUT v Praze)

4. března 2011

Tato prezentace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky.

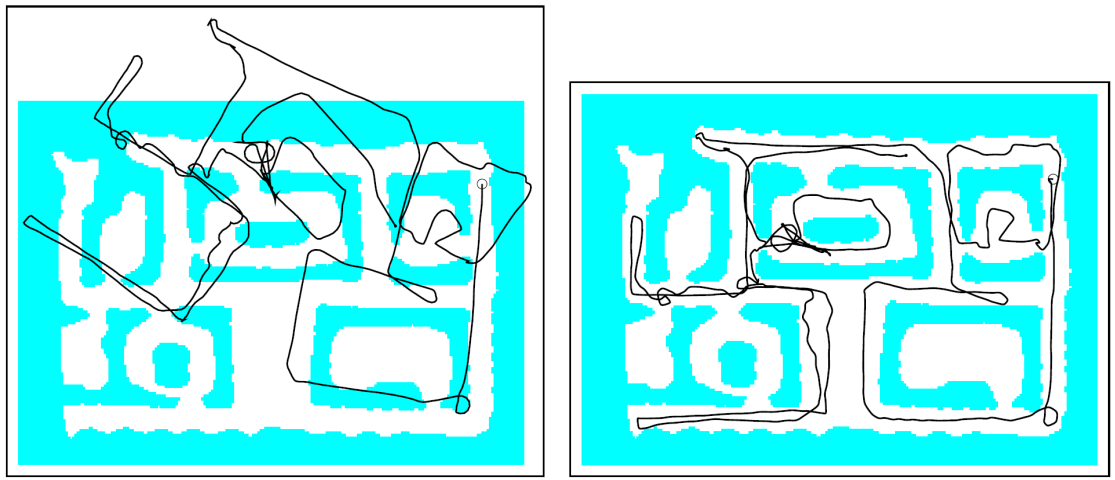
Lokalizace

- Lokalizace je problém určení pozice robotu
  - relativně vzhledem k počáteční pozici robotu
  - v daném systému souřadnic (ve známé mapě)
- Pozice je uvažována včetně natočení!
  tj. (x, y, ϕ), příp. $(x, y, z, roll, pitch, yaw)$



Proč je lokalizace těžká?

- Polohu nelze měřit žádným senzorem => pozice je vypočtena ze senzorických dat
- Ale senzory jsou nepřesné!
- Příkazy jsou prováděny nepřeně
- Jedno senzorické měření k určení polohy nestačí



Taxonomie

- Lokální (kontinuální, „position tracking“)
  - Počáteční pozice je známa
  - Korekce odometrických chyb
  - Omezená chyba
  - Unimodální distribuce (Gausian)
- Globální (absolutní, „lost robot problem“)
  - Počáteční pozice není známa
  - Neomezená chyba
  - Unimodální distribuce nestačí
- Unesený robot („kidnapped robot problem“)
  - Detekce a oprava chyby
  - Vhodné pro testování

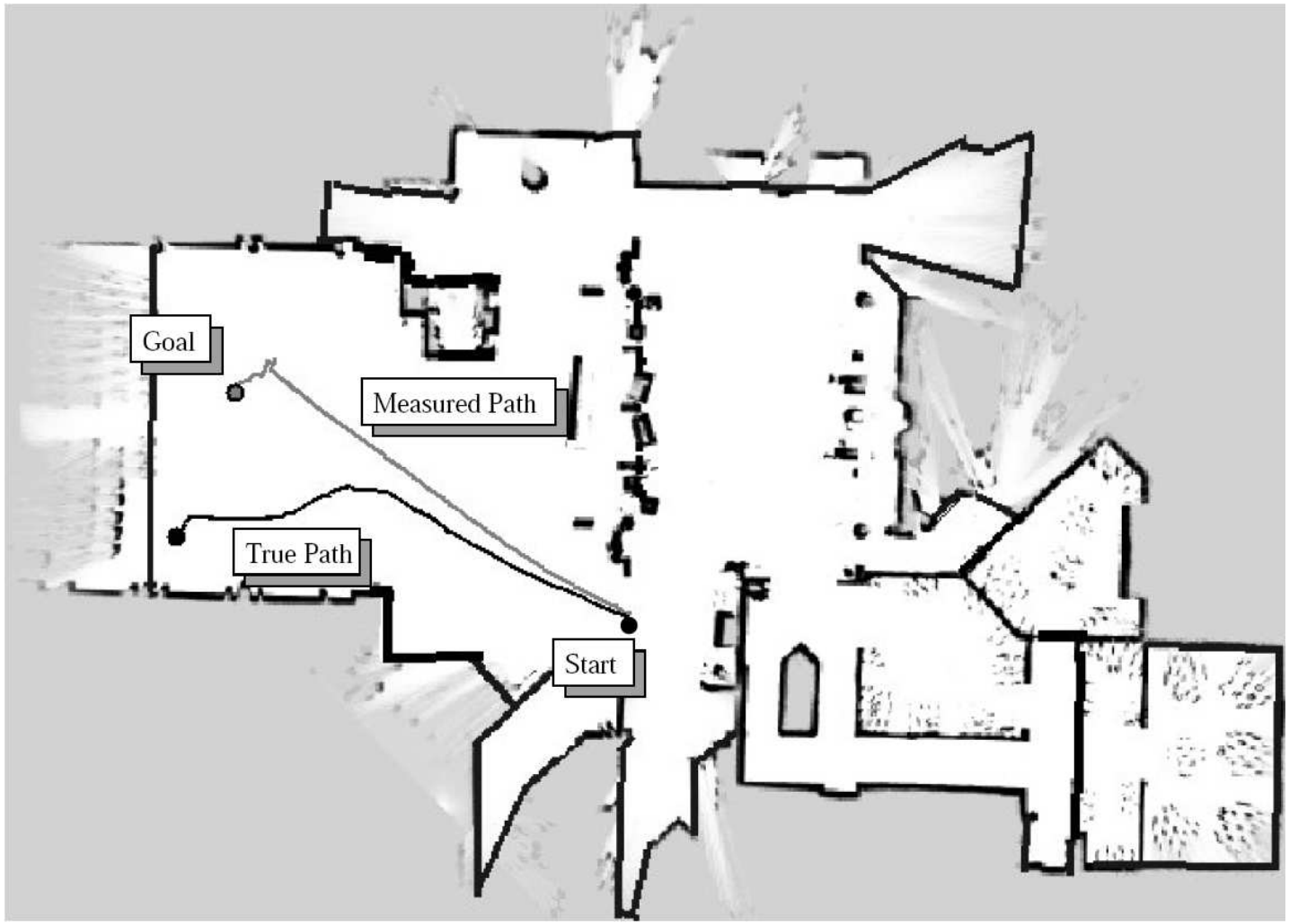
Taxonomie

- Statické prostředí
  - Robot je jediné, co se pohybuje
  - Pozive robotu je jediná proměnná „Hezké matematické“ vlastnosti
- Dynamické prostředí
  - Objekty a/nebo jiné roboty se pohybují
  - Objekty: lidé, světlo (pro kamery), dveře, ...
  - Dva přístup řešení
    - Pohyb objektů lze modelovat => tvoří část stavového popisu úlohy
    - Data o pohyblivých objektech jsou filtrována

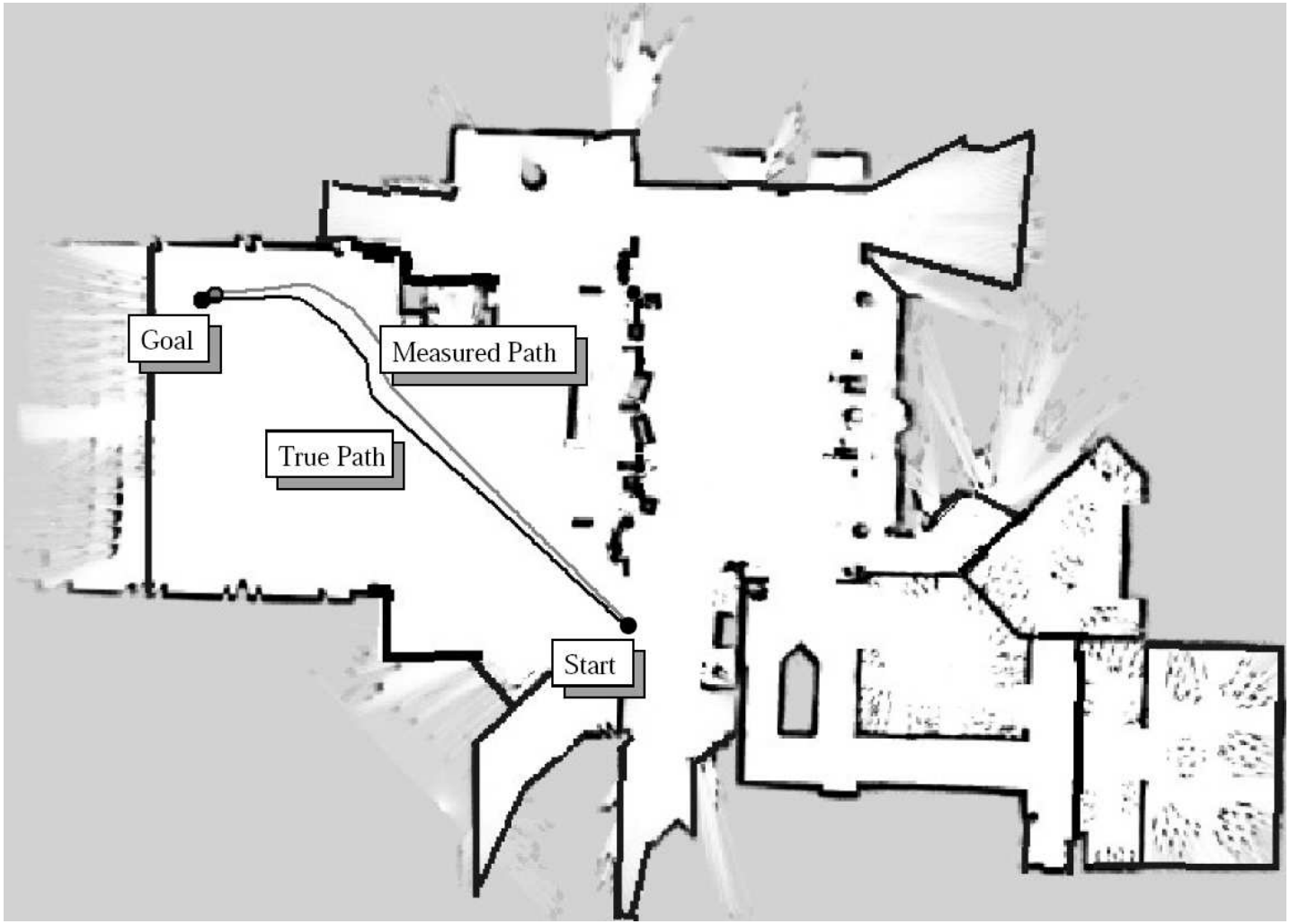
Taxonomie

- Pasivní lokalizace
  - Lokalizační modul pouze sleduje, co se děje - nemá vliv na řízení robotu
  - Lokalizace je vedlejší produkt jiné úlohy
- Aktivní lokalizace
  - Robot je řízen za účelem lepšího určení polohy
  - Dává lepší výsledky ...
  - ... ale ne vždy jde použít
  - Střídání cílů: chvíli se lokalizuje, chvíli se řeší primární úloha
  - Cíl se generuje jako kompromis

Pasivní lokalizace



Aktivní lokalizace

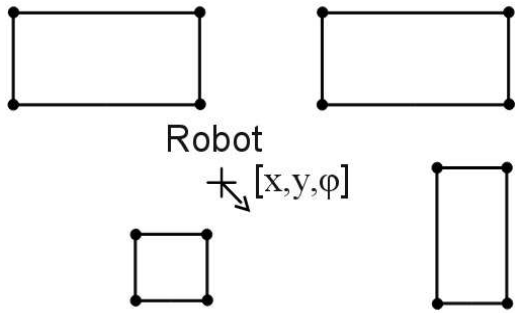


Taxonomie

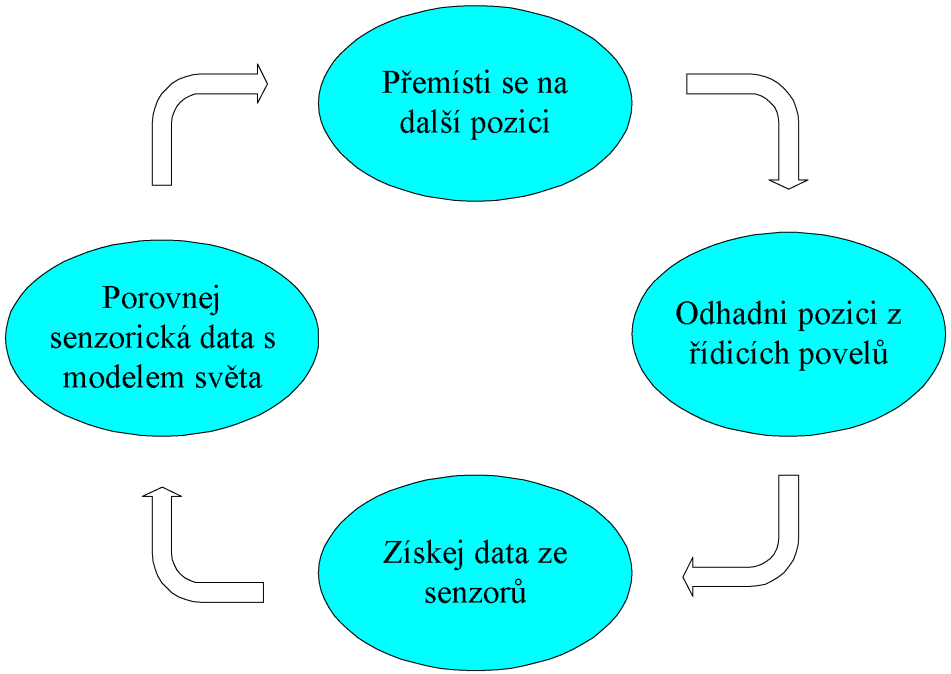
- Jeden robot
  - Klasický problém
  - Není nutná žádná komunikace, vše v jednom
- Více robotů
  - Každý robot se lokalizuje samostatně(tváří se, že ostatní neexistují)
  - Ale pokud jsou roboty schopny se navzájem detekovat, je to lepší
  - Lokalizace sebe sama v mapě druhého robotu
  - Znám svoji polohu, polohu druhého robotu a naši relativní pozici => optimalizace („gumičky“)

My se budeme zabývat

- Terestriální robot pohybující se na rovné horizontální ploše (2D)
- Snímána je jen jedna rovina prostředí
- Většina objektů je statická a detekovatelná senzory
- Používáme senzory měřící vzdálenost
- Pasivní lokalizace
- Jeden robot
- Lokální, globální i nakopnutý robot



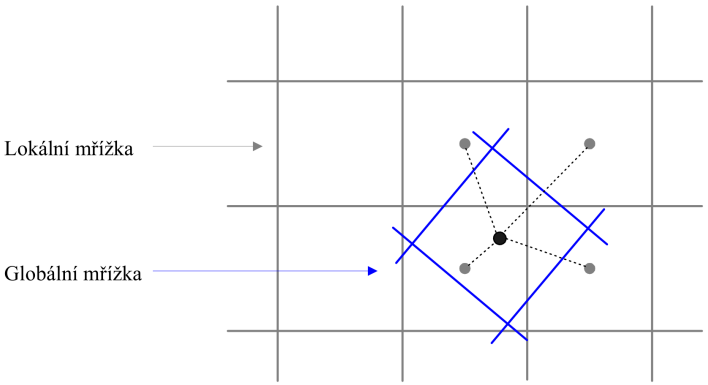
Kontinuální lokalizace

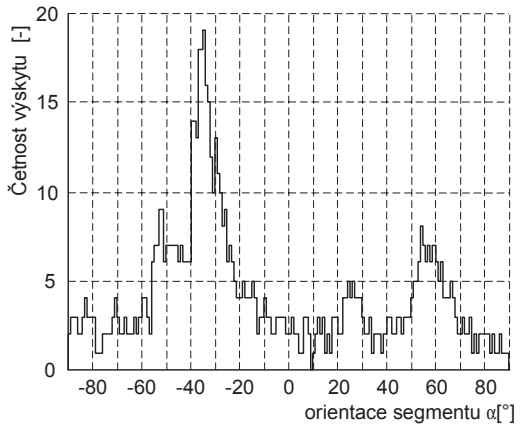


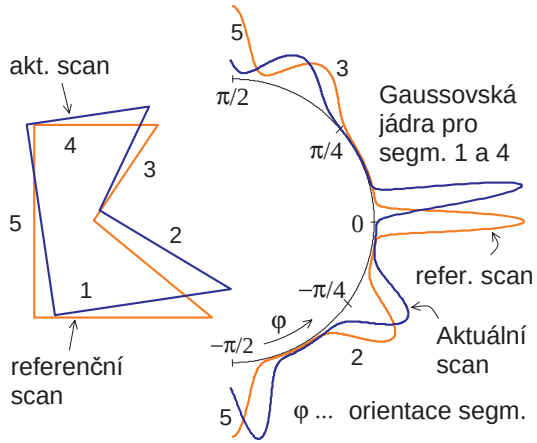
- Poloha se hledá pouze v nejbližším okolí odhadnuté pozice
- Vyžaduje se vyšší přesnost
- Vstup: lokální a globální mapa (scan)
- Výstup: transformace minimalizující míru nesouladu map

Kontinuální lokalizace

- Existuje spousta metod lišících se datovou reprezentací map (scanů)
- Vždy se však jedná o optimalizaci vzhledem k danému kritériu nesouladu map
- Metody:
  - Lokalizace na mřížkách
  - Histogramy
  - Lokalizace s geometrickými primitivy (point-to-line, line-to-line, Houghova transformace)



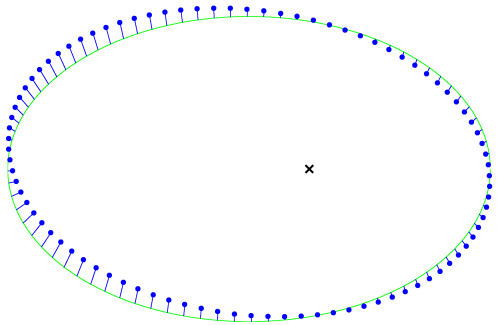




Iterative Closest Point (ICP)

- Iterativní
  - aktuální scan se proloží lomenou čarou
  - naleznou se korespondující body (nejmenší vzdálenost)
  - spočítá se transformace metodou nejmenších čtverců

$$E_{dist}(T_x, T_y, \omega) = \sum_i^n |R_\omega P + T - P'|$$



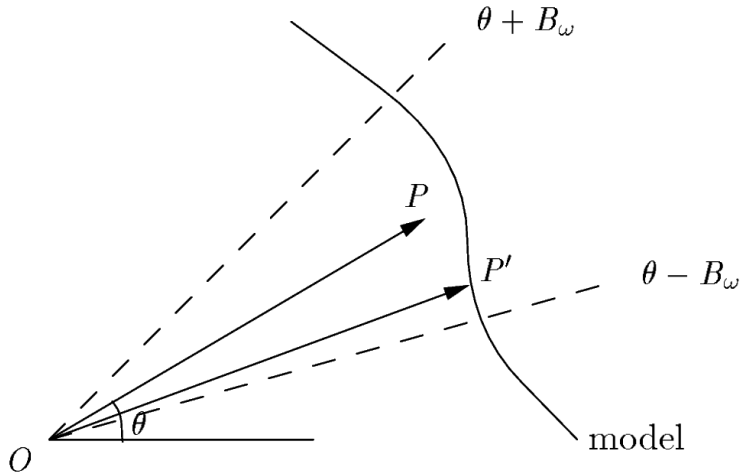
- Má analytické řešení

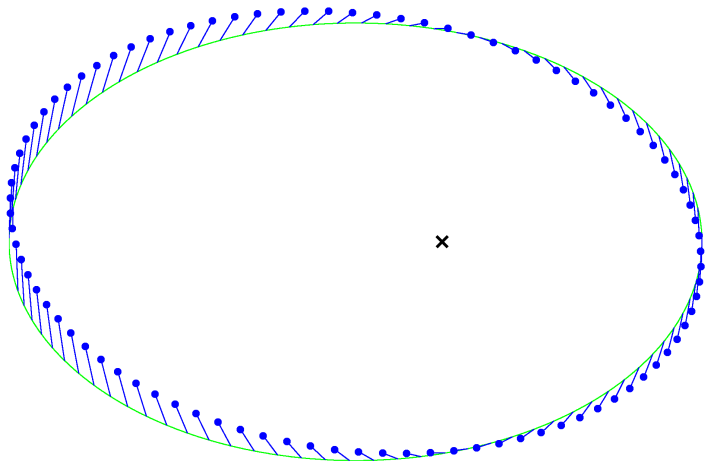
$$\begin{aligned} T_x &= \bar{x'} - (\bar{x}\cos(\bar{\omega} - \bar{y}\sin(\bar{\omega}) \\ T_y &= \bar{y'} - (\bar{x}\sin(\bar{\omega} + \bar{y}\cos(\bar{\omega}) \\ \bar{\omega} &= \arctan \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i' - \bar{y'}) - \sum_{i=1}^n (y_i - \bar{y})(x_i' - x')}{\sum_{i=1}^n (x_i - \bar{x})(y_i' - \bar{y'}) + \sum_{i=1}^n (y_i - \bar{y})(x_i' - \bar{x'})} \end{aligned}$$

- Konverguje pomalu, špatné v rotační složce

Iterative Matching Range Point (IMRP)

- Postup stejný jako u ICP, pouze jiné pravidlo korespondence
- Předpokládáme, že posunutí je nulové $=>$ $|P| \approx |P'|$ a $\tilde{\phi}\phi + \omega$
- Velikost okolí pro hledání korespondencí se postupně zmenšuje
- Zpočátku pomalejší konvergence než u ICP, ale pak rychlejší
- IMRP se snaží chybu posunutí řešit otočením $=>$ nestabilita

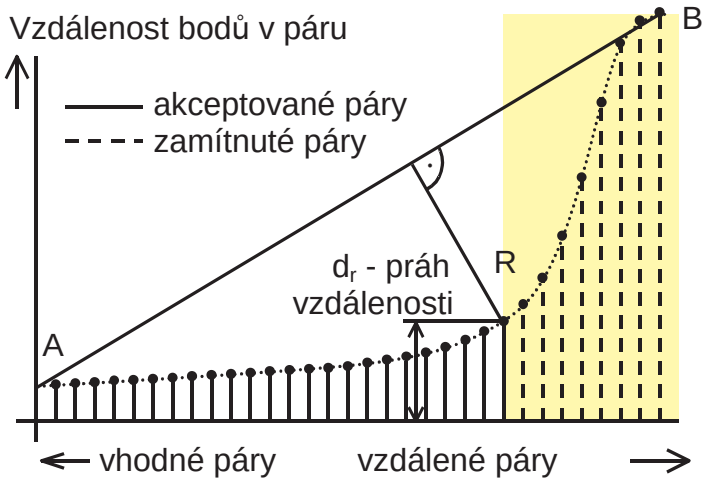




Iterative Dual Correspondence (IDC)

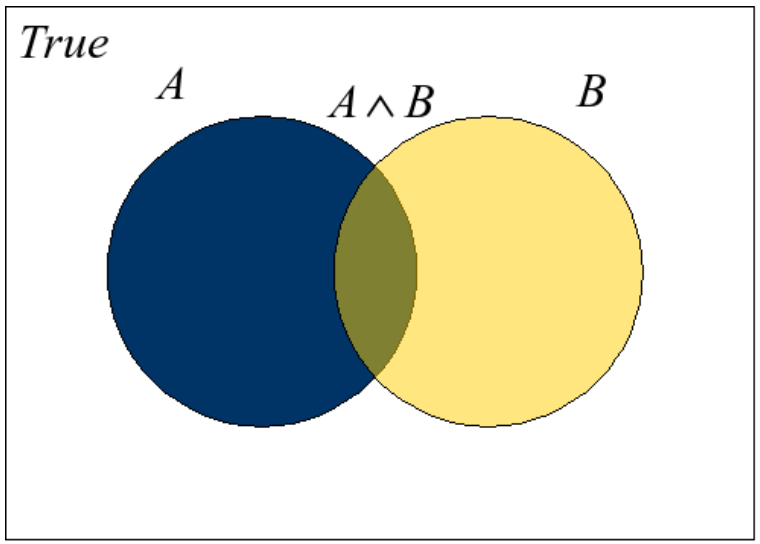
- Kombinace obou pravidel: posunutí z ICP, otočení z IMRP
- Algoritmus
  - Pro každý bod P_i z aktuálního scanu:
    - Použij pravidlo nejbližšího bodu pro nalezení kor. bodu P'
    - Použij pravidlo MRP pro nalezení korespondujícího bodu P''
    - Spočítej metodou nejmenších čtverců (ω_1, T_1) z kor. (P, P')
    - Spočítej metodou nejmenších čtverců (ω_2, T_2) z kor. (P, P'')
    - Hledaná korespondence je (ω_2, T_1)
  - Přechozí kroky opakuj, dokud postup nekonverguje

Vylepšení: Po nalezení párů se z následující optimalizace vyloučí odlehlé bodové páry



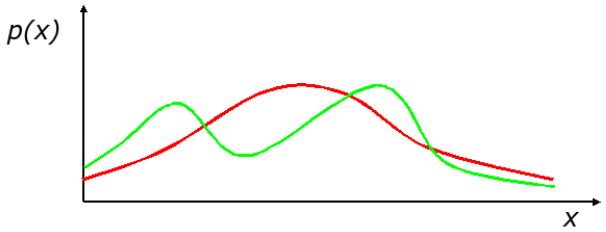
Lehký úvod do teorie pravděpodobnosti

- Idea: explicitní reprezentace nejistoty pomocí kalkulu teorie pravděpodobnosti
- p(X=x) pravděpodobnost, že náhodná proměnná X nabývá hodnoty x
- $0 \leq p(x) \leq 1$
- $p(true) = 1, p(false) = 0$
- $p(A \vee B) = p(A) + p(B) - p(A \wedge B)$

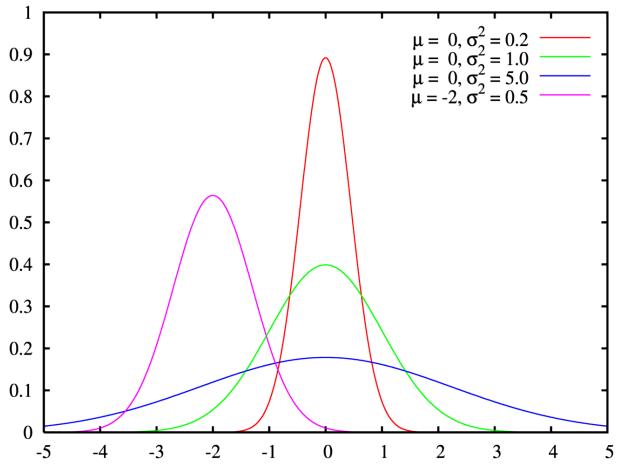


Diskrétní a spojitá náhodná veličina

- **Diskrétní**: X je spočetná, tj. $X = x_1, x_2, \ldots, x_n$
- **Spojitá**: X nabývá nespočetně mnoha hodnot (z intervalu)
- p se nazývá hustota pravděpodobnosti

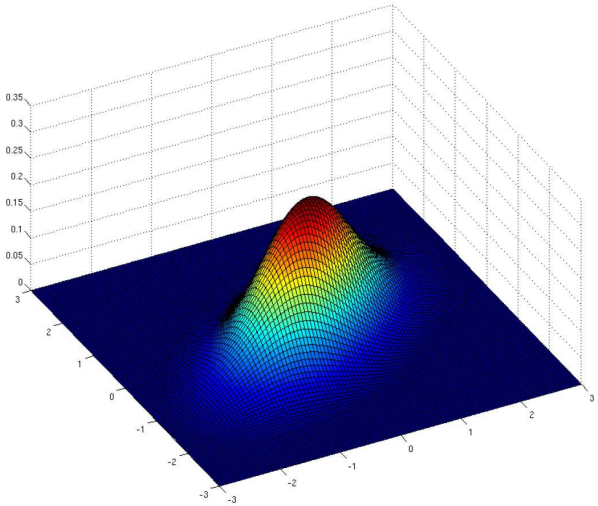


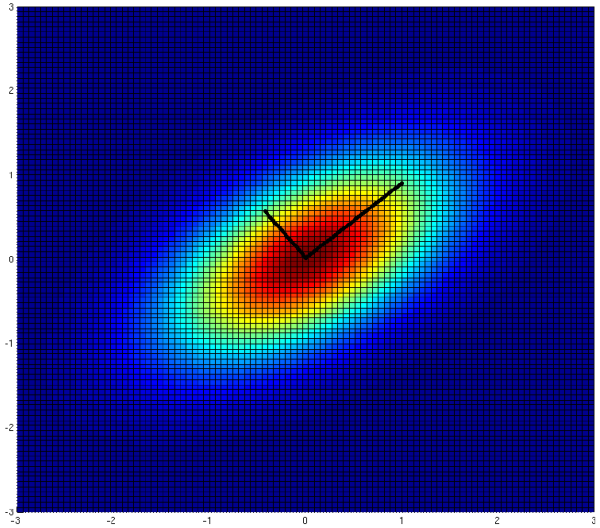
- Různá rozložení
- Nejznámější: Normální (Gausián)
- $p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$



Vícerozměrné normální rozložení

$$p(x) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$





- Vlastní čísla a vlastní vektory kovarianční matice definují elipsy.

Sdružená a podmíněná pravděpodobnost

- $p(X = x \text{ a } Y = y) = p(x, y)$
- Pokud X and Y jsou nezávislé, potom

$$p(x, y) = p(x)p(y)$$

- $p(x|y)$ je pravděpodobnost x za předpokladu y

$$p(x|y) = p(x, y)/p(y)$$

$$p(x, y) = p(x|y)/p(y)$$

- Pokud X a Y jsou nezávislé, potom

$$p(x|y) = p(x)$$

Teorém úplné pravděpodobnosti

Diskrétní případ

$$\sum_x p(x) = 1$$

$$p(x) = \sum_y p(x, y)$$

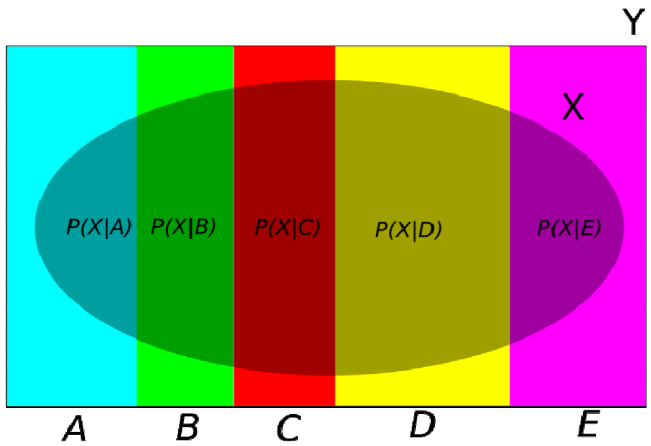
$$p(x) = \sum_y p(x|y)p(y)$$

Spojitý případ

$$\int_x p(x)dx = 1$$

$$p(x) = \int_y p(x, y)dy$$

$$p(x) = \int_y p(x|y)p(y)dy$$



Bayesův vzorec

$$p(x,y) = p(x|y)p(y) = p(y|x)p(x)$$

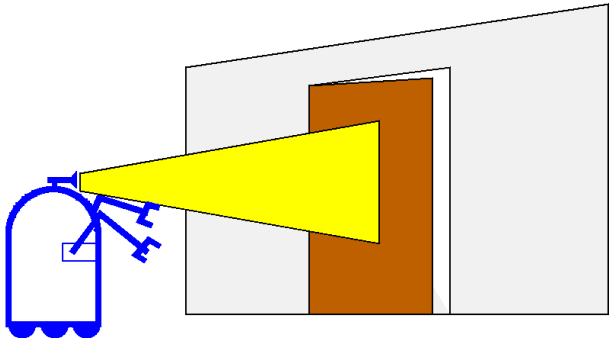
$$\Rightarrow$$

$$\boxed{p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{likelihood \cdot prior}{evidence}}$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \eta p(y|x)p(x)$$

$$\eta = p(y)^{-1} = \frac{1}{\sum_x p(y|x)p(x)}$$

Příklad - odhad stavu



- Robot naměří z
- Jaká je pravděpodobnost $p(open|z)$?
- $p(open|z)$ je diagnostika
- $p(z|open)$ je příčina
- Většinou je jednodušší získat příčinu (počítání frekvencí)
- Bayesův vzorec umožňuje z příčiny spočítat důsledek:

$$p(open|z) = \frac{p(z|open)p(open)}{p(z)}$$

Příklad - otevřené dveře

- $p(z|open) = 0.6\ p(z|\neg open) = 0.3$
- $p(open) = p(\neg) = 0.5$

$$p(open|z) = \frac{p(z|open)p(open)}{p(z|open)p(open) + p(z|\neg open)p(\neg open)}$$

$$p(open|z) = \frac{0.6 \cdot 0.5}{0.6 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{2}{3} = 0.67$$

- Po naměření z se jistota otevření dveří zvýší.

Příklad - druhé měření

- $p(z_2|open) = 0.5\ p(z_2|\neg open) = 0.6$
- $p(open|z_1) = \frac{2}{3}$

$$p(open|z_2 z_1) = \frac{p(z_2|open)p(open|z_1)}{p(z_2|open)p(open|z_1) + p(z_1|\neg open)p(\neg open|z_1)}$$

$$= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625$$

- z_2 snižuje pravděpodobnost, že jsou dveře otevřené.

Akce

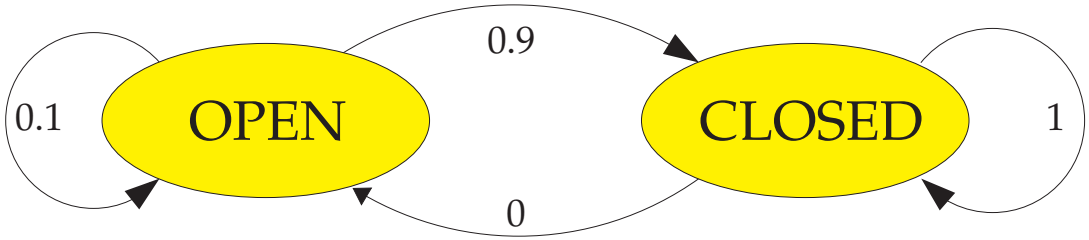
- Svět je dynamický
  - robot se hýbe,
  - jiné objekty v prostředí se hýbou
  - a čas prostě plyne (květiny rostou).
- Akce se nikdy nestane tak, jak byla naplánována.
- Na rozdíl od měření akce obvykle zvýší míru nejistoty.
- Pro integraci akce u do aktuální „víry“ použijeme podmíněnou pravděpodobnost

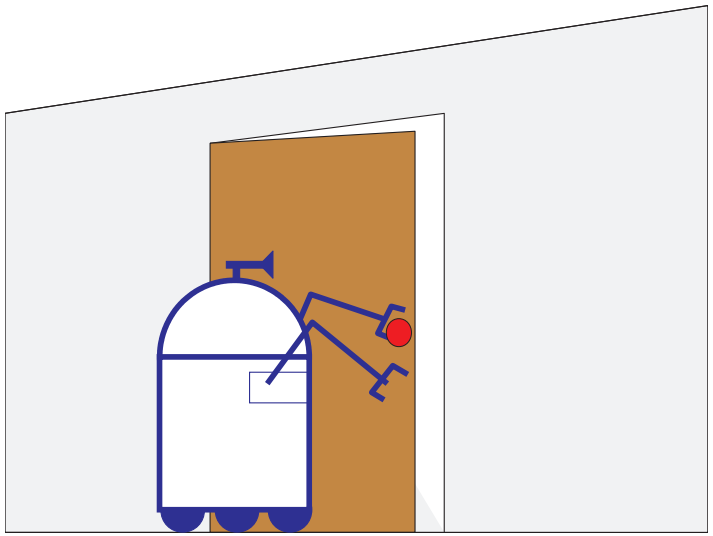
$$p(x|u, x')$$

- Tento výraz specifikuje pravděpodobnost(hustotu pravděpodobnosti) stavu x, pokud provedeme akci u ve stavu x'.

Pokračování příkladu - zavíráme dveře

$p(x|u, x')$ pro $u =$ „zavři dveře“





$$p(x, u) = \sum_{x'} p(x|u, x')p(x')$$

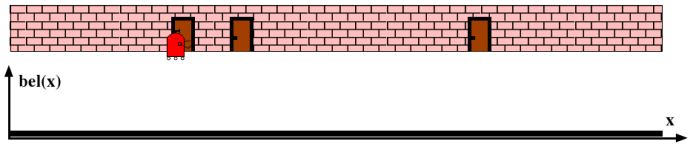
Pokud jsou dveře otevřené, akce „zavři dveře“ je úspěšná v 90% případech.

Pokračování příkladu - zavíráme dveře

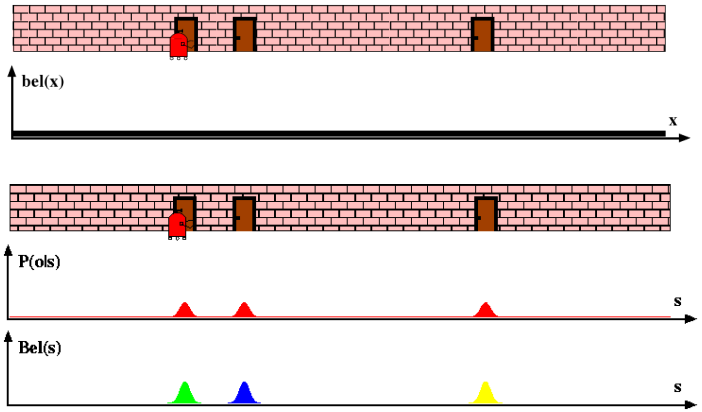
$$
\begin{aligned}
p(closed|u) &= \sum_{x'} p(closed|u,x')p(x') \\
&= p(closed|u,open)p(open) \\
&+ p(closed|u,closed)p(closed) \\
&= \frac{9}{10}\cdot\frac{5}{8} + \frac{1}{1}\cdot\frac{3}{8} = \frac{15}{16}
\end{aligned}
$$

$$
\begin{aligned}
p(open|u) &= \sum_{x'} p(open|u,x')p(x') \\
&= p(open|u,open)p(open) \\
&+ p(open|u,closed)p(closed) \\
&= \frac{1}{10}\cdot\frac{5}{8} + \frac{0}{1}\cdot\frac{3}{8} = \frac{1}{16} \\
&= 1 - p(closed|u)
\end{aligned}
$$

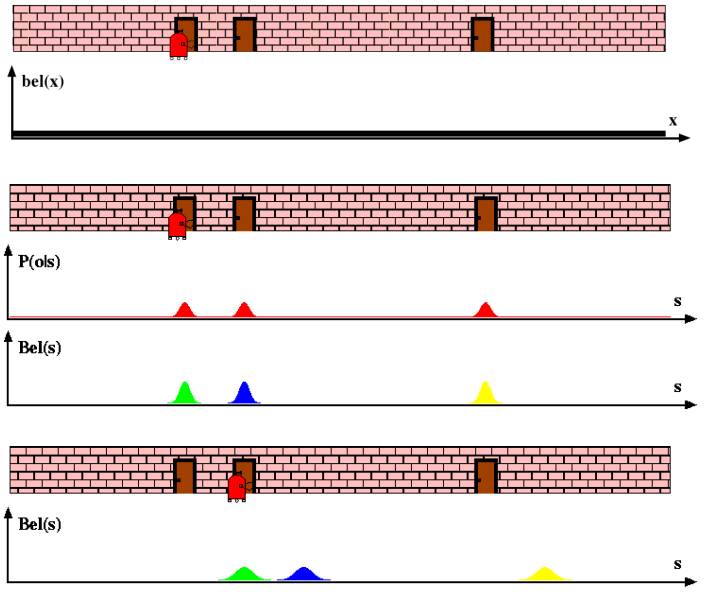
Motivace



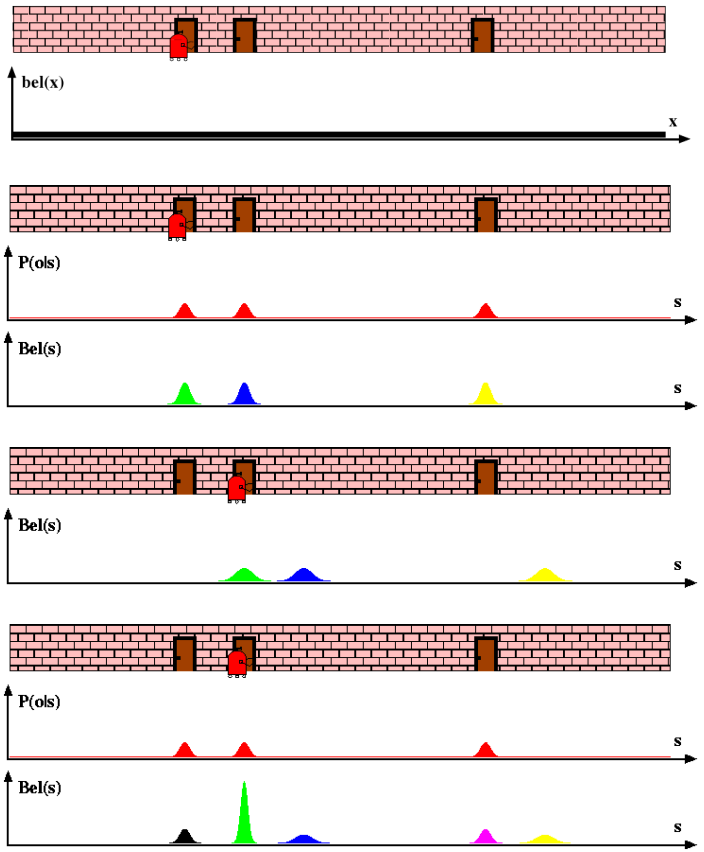
Motivace



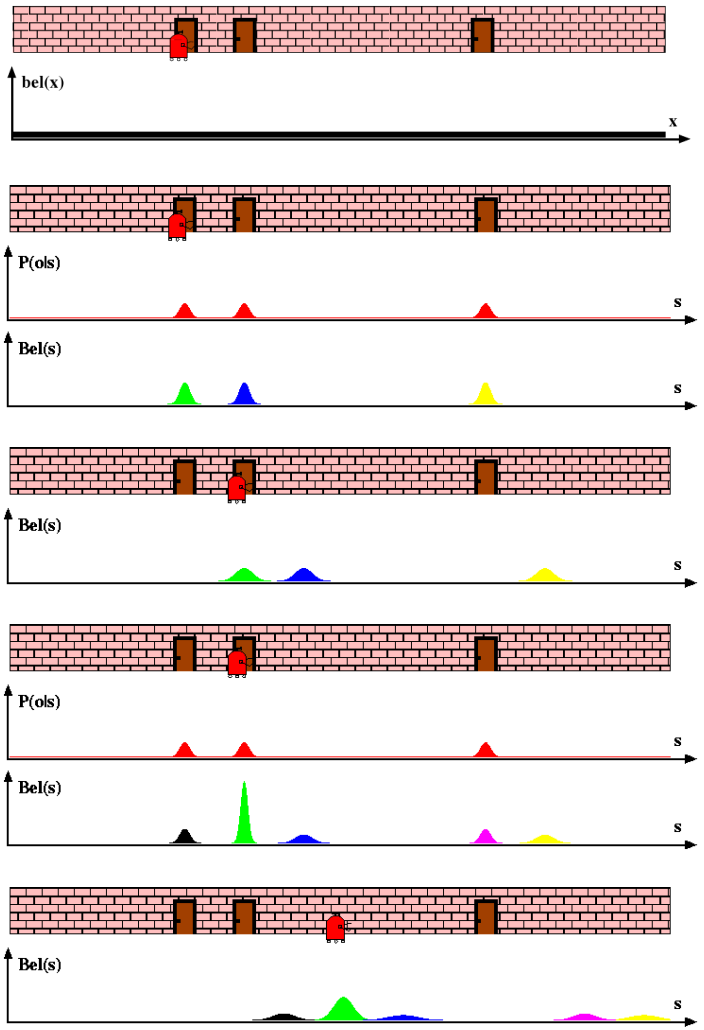
Motivace



Motivace



Motivace



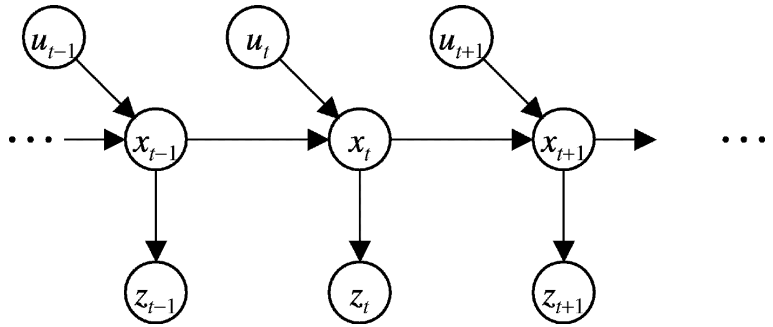
Bayesův filtr: framework

- Je dáno:
  - Posloupnost pozorování z a akcí u:

$$d_t = \{u_1, z_1, \ldots, u_t, z_t\}$$

  - Sensorický model $p(z|x)$
  - Model akcí (pohybu) $p(x|u, x')$
  - Počáteční (prior) pravděpodobnost stavu $p(x)$
- Chceme:
  - Odhad stavu (posterior)
  - Posteriorní pravděpodobnost označujeme jako belief (víra): $Bel(x_t) = p(x_t|u_1, z_1, \ldots, u_t, z_t)$

Markovův předpoklad



Následující stav závisí pouze na aktuálním stavu a nezávisí na stavech předchozích.

$$p(z_t|x_{0:t}, z_{1:t}, u_{1:t}) = p(z_t|x_t)$$
$$p(x_t|x_{1:t-1}, z_{1:t}, u_{1:t}) = p(x_t|x_{t-1}, u_t)$$

S tím souvisí:

- Statický svět (nebo dynamika modelovaná v x)
- Dokonalý model světa (model senzoru) a model pohybu
- Nezávislý šum
- Přesná reprezentace $p(x)$

Bayesův filtr - odvození

$$
\begin{aligned}
\boxed{Bel(x_t)} &= p(x_t|u_1, z_1, \ldots, u_t, z_t) \\
\text{Bayes} \quad &= \eta p(z_t|x_t, u_1, z_1, \ldots, u_t) p(x_t|u_1, z_1, \ldots, u_t) \\
\text{Markov} \quad &= \eta p(z_t|x_t) p(x_t|u_1, z_1, \ldots, u_t) \\
\text{Total prob.} \quad &= \eta p(z_t|x_t) \int p(x_t|u_1, z_1, \ldots, u_t, x_{t-1}) \\
&\quad p(x_{t-1}|u_1, z_1, \ldots, u_t) dx_{t-1} \\
\text{Markov} \quad &= \eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|u_1, z_1, \ldots, u_t) dx_{t-1} \\
\text{Markov} \quad &= \eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|u_1, z_1, \ldots, z_{t-1}) dx_{t-1} \\
&= \boxed{\eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}}
\end{aligned}
$$

Bayesův filtr

$$Bel(x_t) = \eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Algoritmus Bayes_filter($Bel(x), d$)

if d is a measurement z **then**
  $\eta = 0$
  for all x **do**
    $Bel'(x) = p(z|x)Bel(x)$
    $\eta = \eta + Bel'(x)$
  end for
  for all x **do**
    $Bel'(x) = \eta^{-1} Bel'(x)$
  end for
end if

if d is a action u **then**
  for all x **do**
    $Bel'(x) = \int p(x|u, x')Bel(x')dx'$
  end for
end if

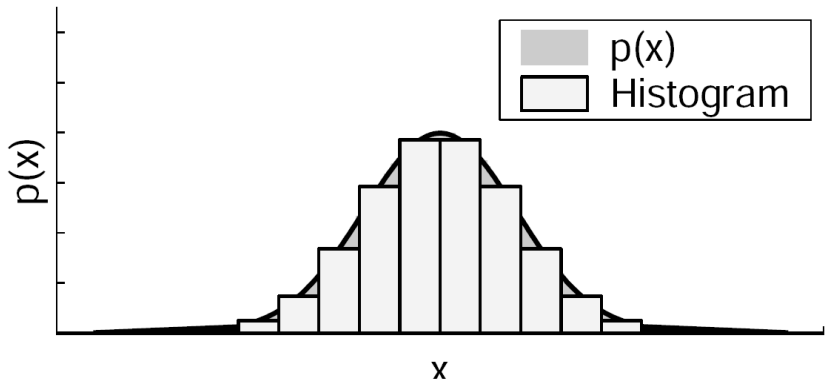
return $Bel'(x)$

Bayesovy filtry - celá rodina metod

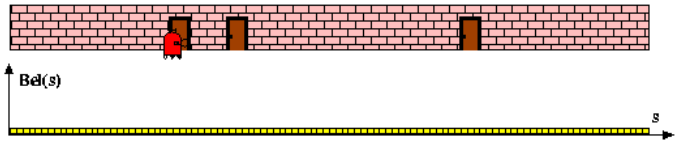
- Kalmanův filtr
- Histogramový filtr
- Particle filtr
- Skryté Markovovy modely
- Dynamické Bayesovy sítě
- Částečně pozorovatelné Markovovy rozhodovací procesy (POMDPs)

Neparametrické filtry

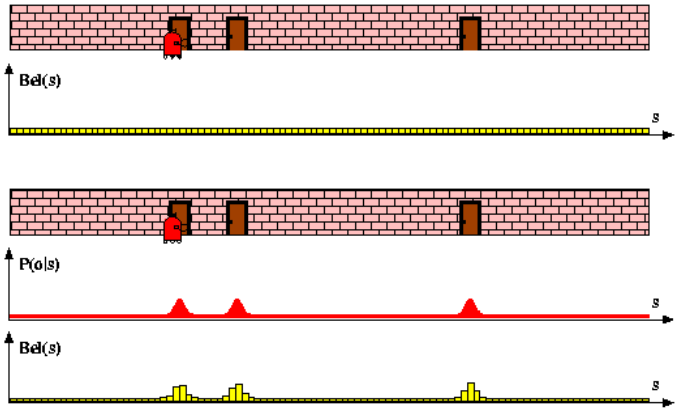
- Aproximace hustoty pravděpodobnosti konečnou množinou hodnot
- Umí reprezentovat libovolné rozložení
- Velikost množiny reprezentantů ovlivňuje rychlost výpočtu a kvalitu filtrace
- Histogramový x částicový filtr



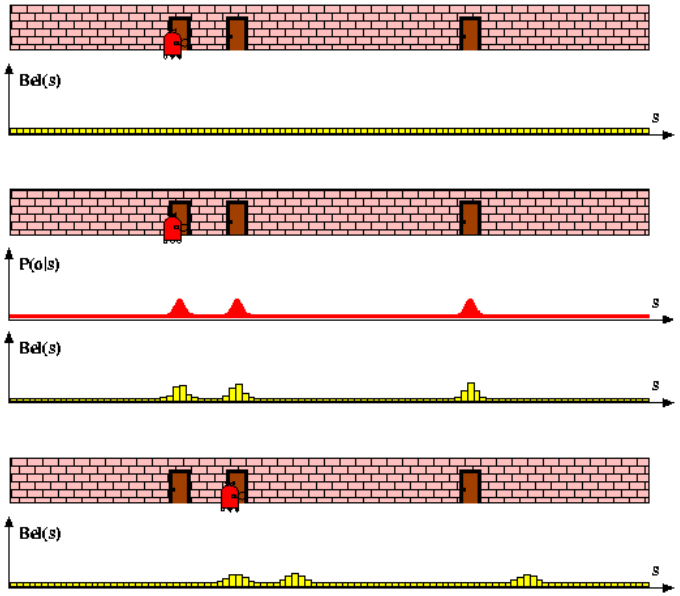
Histogramový filtr



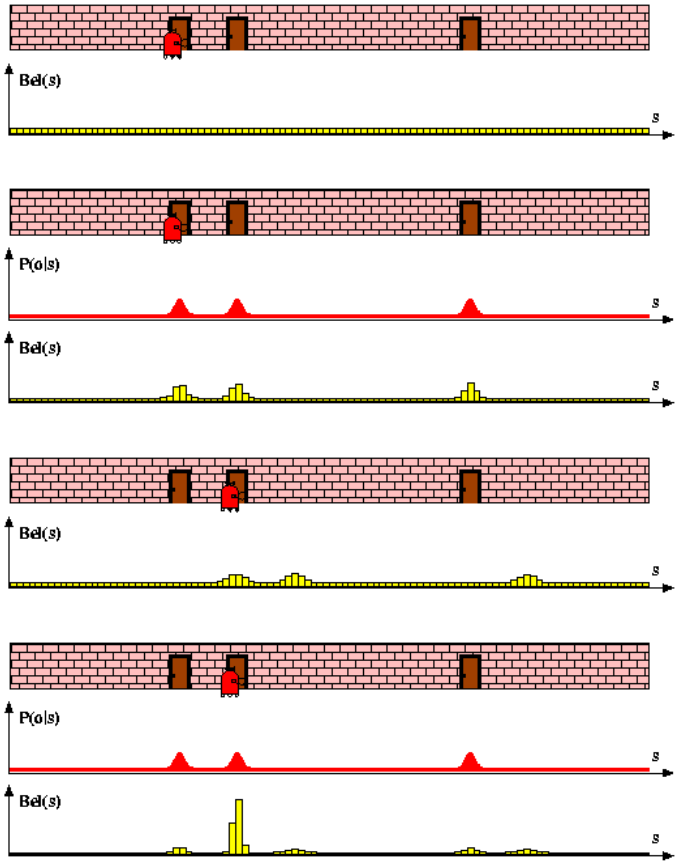
Histogramový filtr



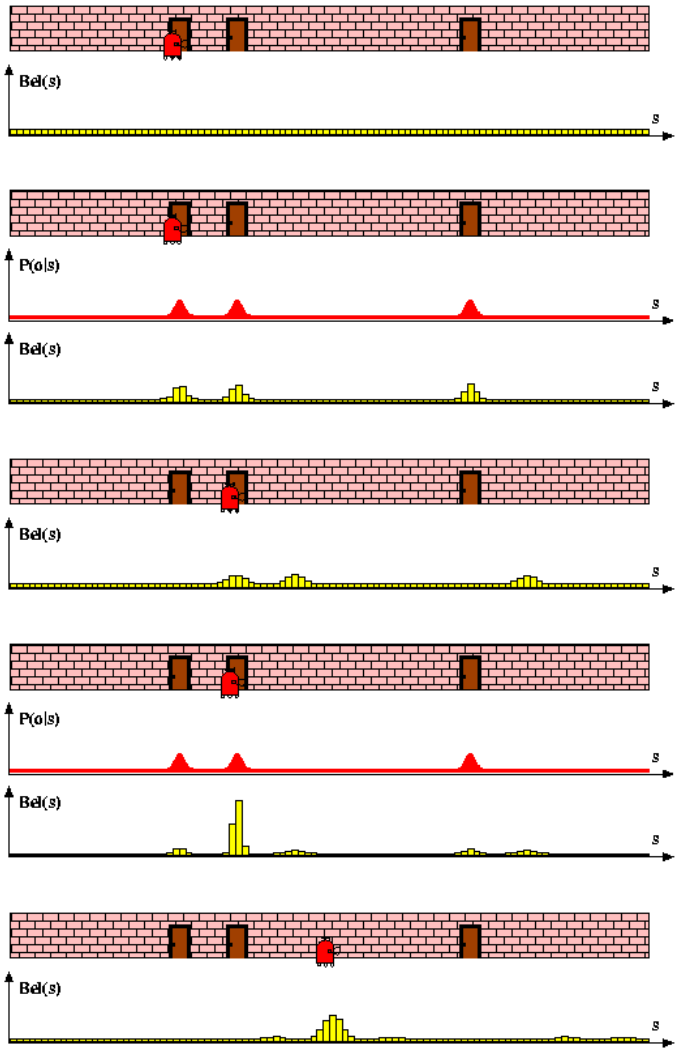
Histogramový filtr



Histogramový filtr

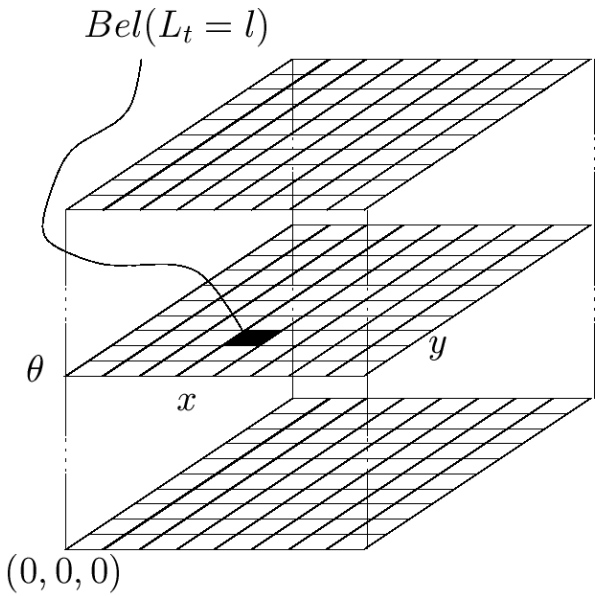


Histogramový filtr



Histogramový filtr

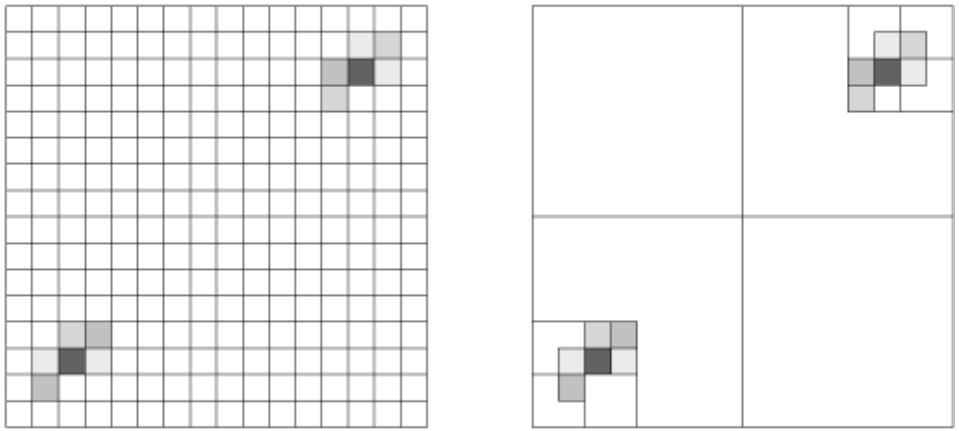
if d is a measurement z **then**
  $\eta = 0$
  for all x **do**
    $Bel'(x) = p(z|x)Bel(x)$
    $\eta = \eta + Bel'(x)$
  end for
  for all x **do**
    $Bel'(x) = \eta^{-1}Bel'(x)$
  end for
else if d is a action u **then**
  for all x **do**
    $Bel'(x) = \int p(x|u, x')Bel(x')dx'$
  end for
end if
return $Bel'(x)$



$$Bel(x_t = \langle x, y, \phi \rangle)$$

Histogramový filtr

- Složitost $O(n^2)$ – při aktualizaci z modelu pohybu se prochází každá buňka!
- Selektivní update
  - uvažuje se jen část stavového prostoru
  - ale pak je potřeba sledovat, jak úspěšně je robot lokalizován
- Dynamická dekompozice stavového prostoru – kd-stromy (density trees): „jemnost“ dělení závisí na hustotě pravděpodobnosti (vyšší hustota => jemnější dělení)

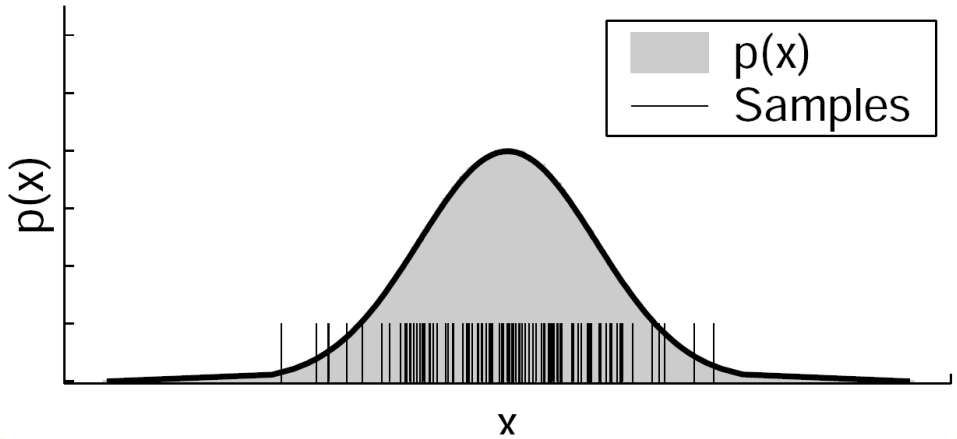


Částicový filtr (particle filter)

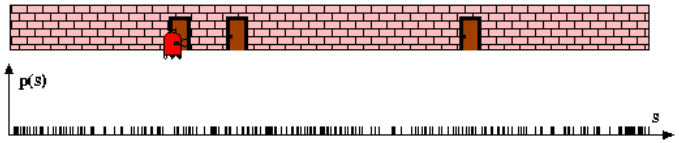
- Hustota pravděpodobnosti je reprezentována „vhodně“ (náhodně umístěnými vzorky):

$$Bel(x_t) \approx \{x_{(i)}, w_{(i)}\}_{i=1,\ldots,m}$$

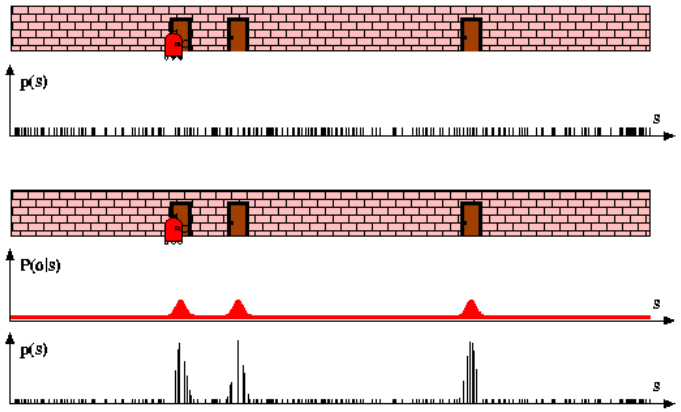
- Vzorky jsou vážené.
- Podle váhy v čase $t-1$ se vybírají vzorky pro čas t
- Velice jednoduchý na implementaci
- Nejuniverzálnější z BF: reprezentace ne Gausovské distribuce a nelineárních procesů



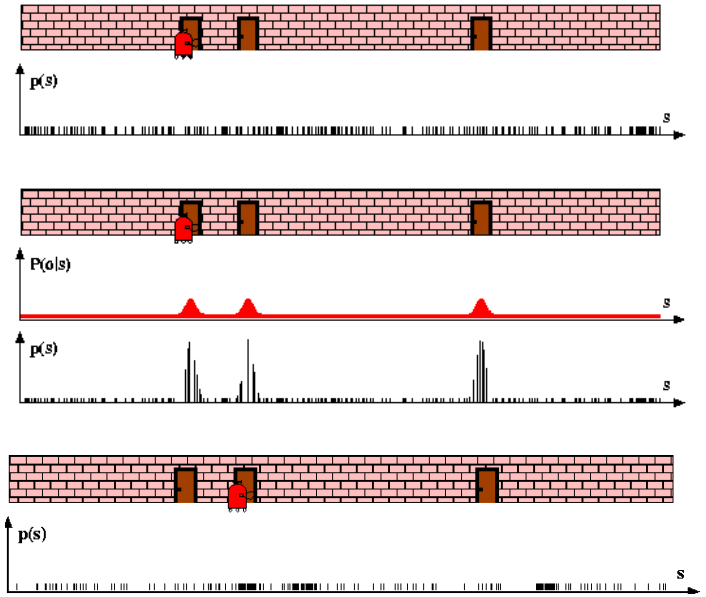
Částicový filtr



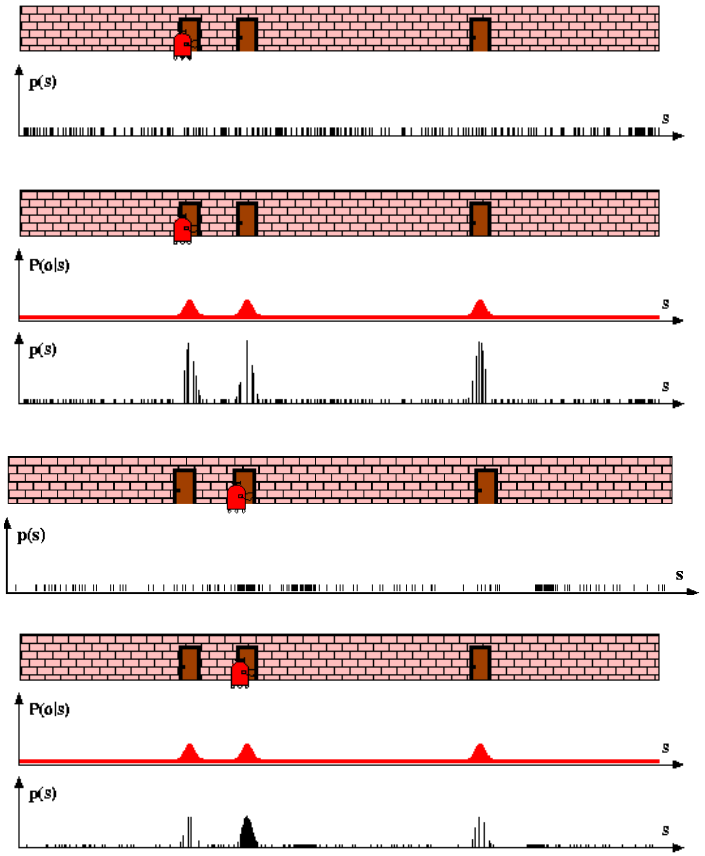
Částicový filtr



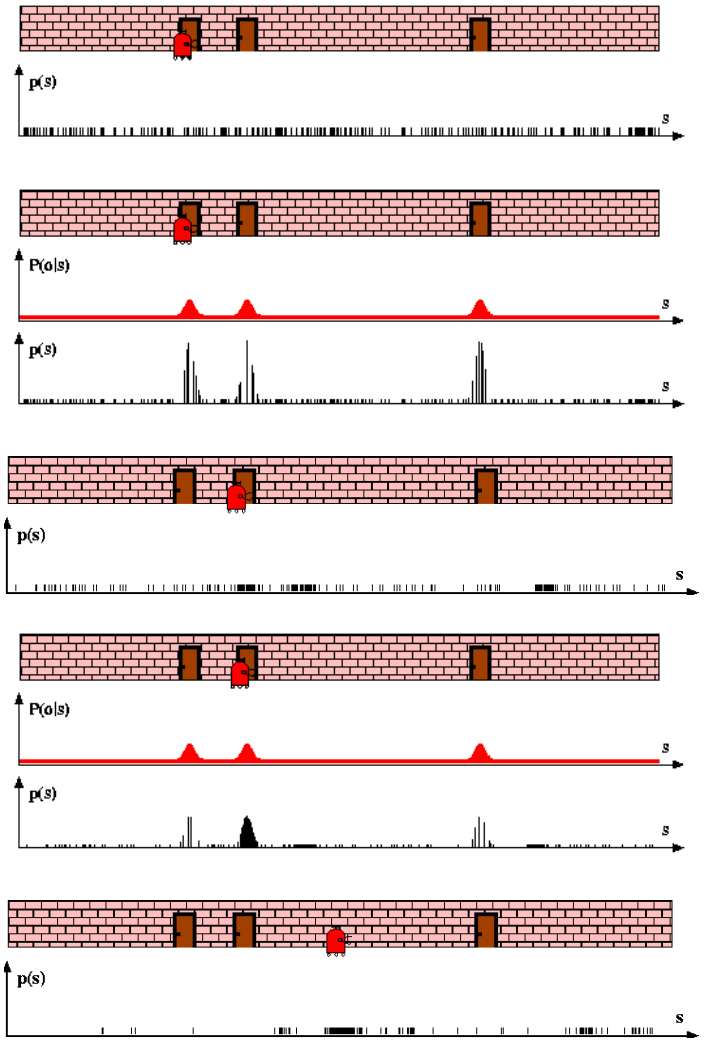
Částicový filtr



Částicový filtr



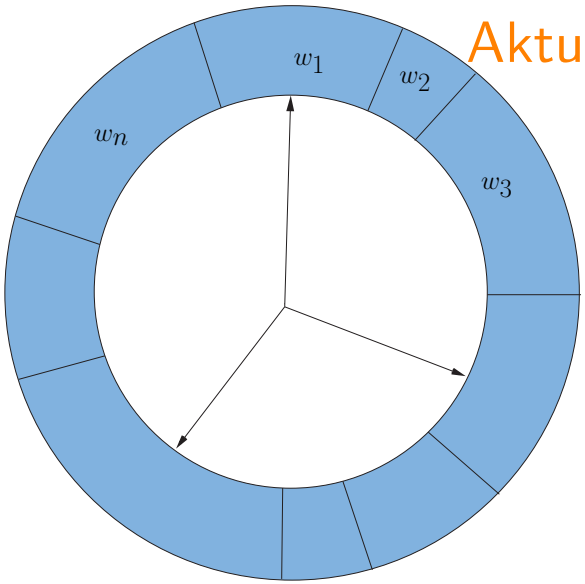
Částicový filtr



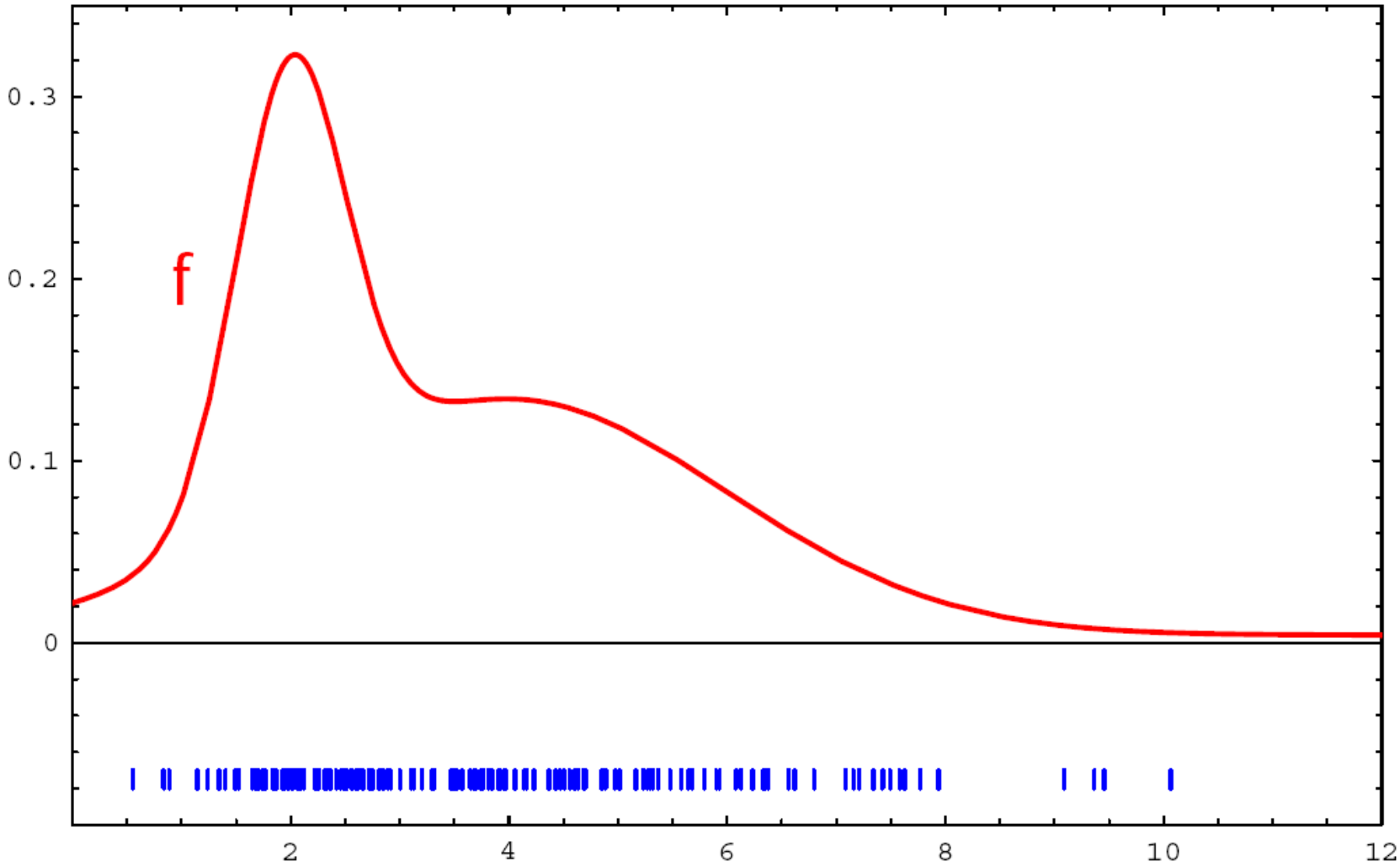
Algoritmus částicového filtru

Particle_filter(S_{t-1}, u_{t-1}, z_t)

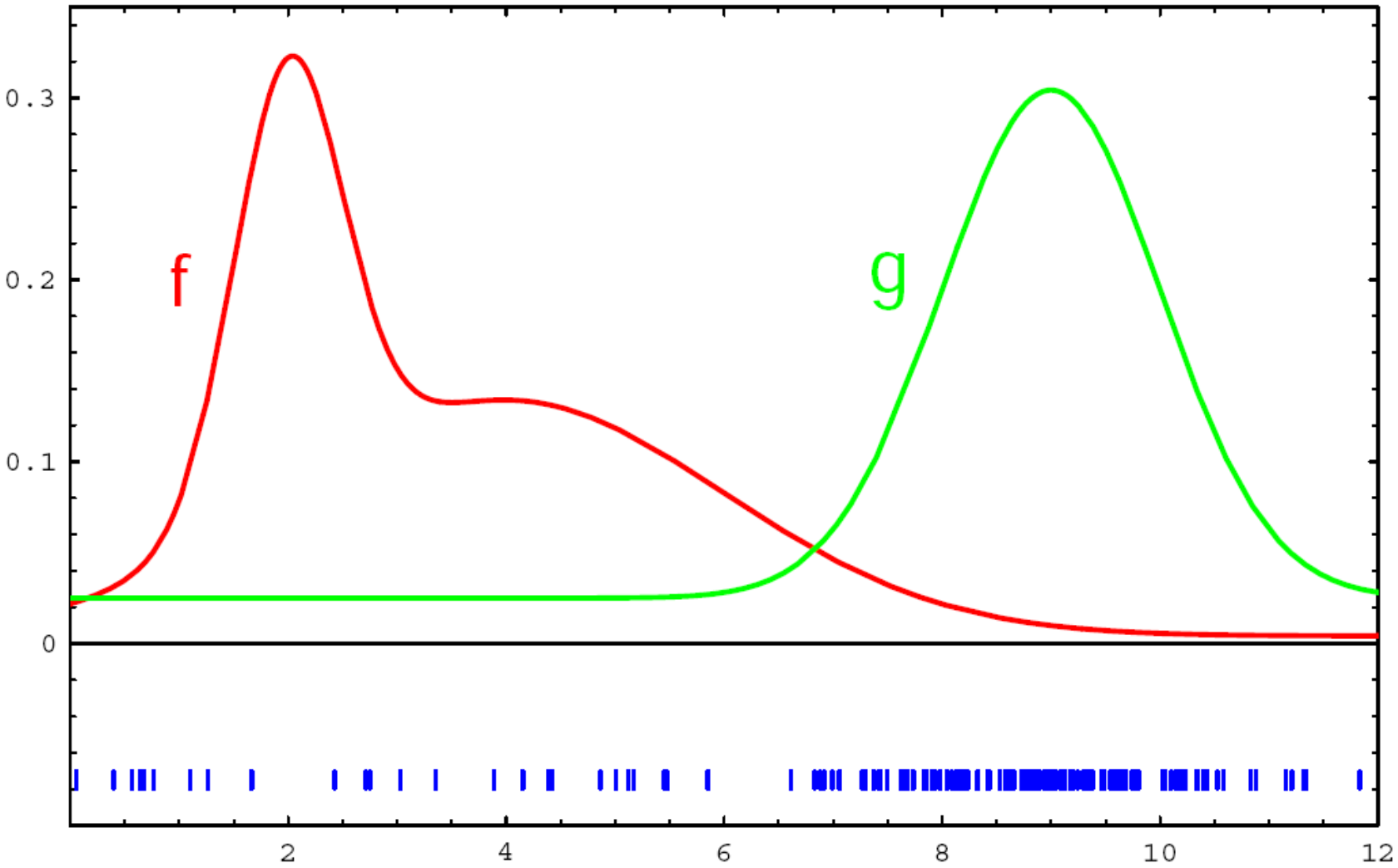
$S_t = \emptyset,\ \eta = 0$
for $i = 1, \ldots, n$ **do** — Generuj nové vzorky
  Vyber index j_i podle diskrétní distribuce dané w_{t-1}
  Vyber x_t^i z $p(x_t|x_{t-1}, u_{t-1})$ na základě $x_{t-1}^{j_i}$ a u_{t-1}
  $w_t^i = p(z_t|x_t^i)$ — Spočítej váhu
  $\eta = \eta + w_t^i$ — Aktualizuj normalizační faktor
  $S_t = S_t \cup \{\langle x_t^i, w_t^i \rangle\}$ — Přidej vzorek
end for
for $i = 1, \ldots, n$ **do** — Normalizuj váhy
  $w_t^i = \frac{w_t^i}{\eta}$
end for



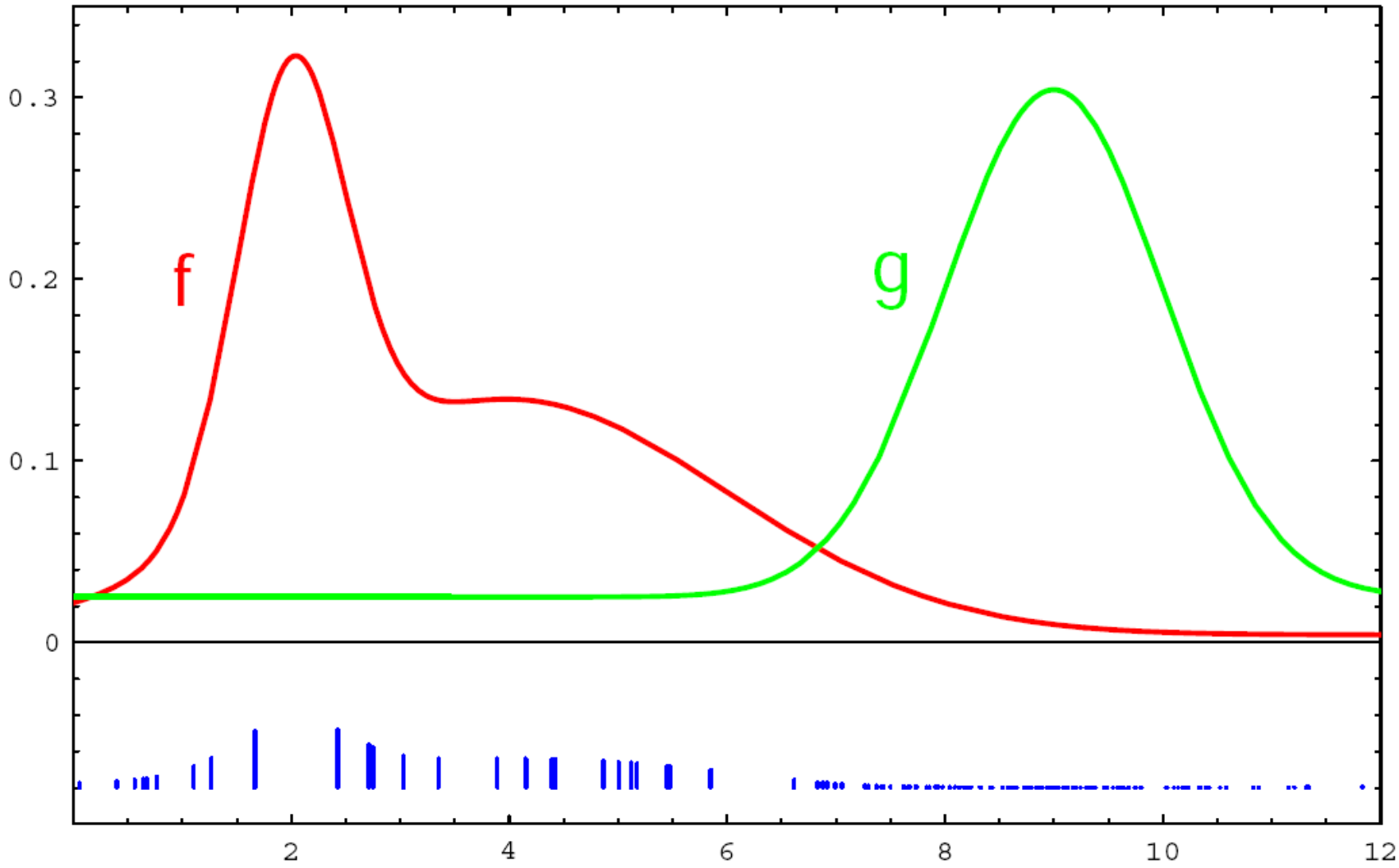
Proč se váží?



Proč se váží?



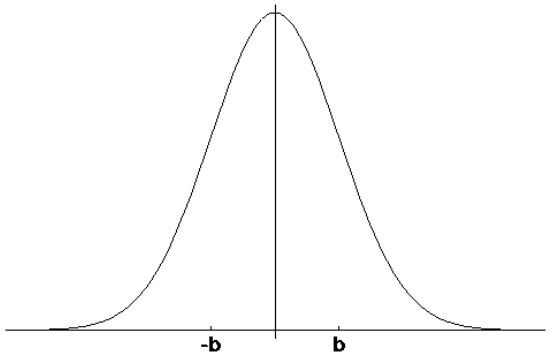
Proč se váží?

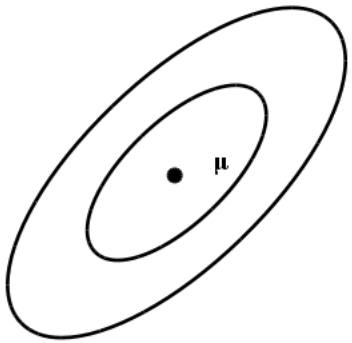


Kalmanův filtr

- Unimodální reprezentace hustoty pravděpodobnosti Gaussiánem

$$
\begin{aligned}
p(x) &\sim N(\mu, \sigma^2) \\
p(x) &= \frac{1}{\sqrt{2\pi\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}} \\
p(x) &\sim N(\mu, \mathbf{\Sigma}) \\
p(x) &= \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}
\end{aligned}
$$





Lineární transformace

- Lineární transformace zachovávají normální rozdělení

$$\left.\begin{array}{r} X \sim N(\mu, \sigma^2) \\ Y = aX + b \end{array}\right\} => Y \sim N(a\mu + b, a^2\sigma^2)$$

$$\left.\begin{array}{r} X_1 \sim N(\mu_1, \sigma_1^2) \\ X_2 \sim N(\mu_2, \sigma_2^2) \end{array}\right\} => p(X_1)p(X_2) \sim N\left(\frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\mu_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\mu_2, \frac{1}{\sigma_1^{-2} + \sigma_2^{-2}}\right)$$

$$\left.\begin{array}{r} X \sim N(\mu, \Sigma) \\ Y = AX + B \end{array}\right\} => Y \sim N(A\mu + B, A\Sigma A^T)$$

$$\left.\begin{array}{r} X_1 \sim N(\mu_1, \Sigma_1) \\ X_2 \sim N(\mu_2, \Sigma_2) \end{array}\right\} => p(X_1)p(X_2) \sim N\left(\frac{\Sigma_2}{\Sigma_1 + \Sigma_2}\mu_1 + \frac{\Sigma_1}{\Sigma_1 + \Sigma_2}\mu_2, \frac{1}{\Sigma^{-1} + \Sigma_2^{-1}}\right)$$

Základní předpoklady

- Musí platit Markovova podmínka + tři následující:
  - Pravděpodobnost přechodu stavu (model pohybu) $p(x|u, x')$ je lineární s přidaným Gausovým šumem:

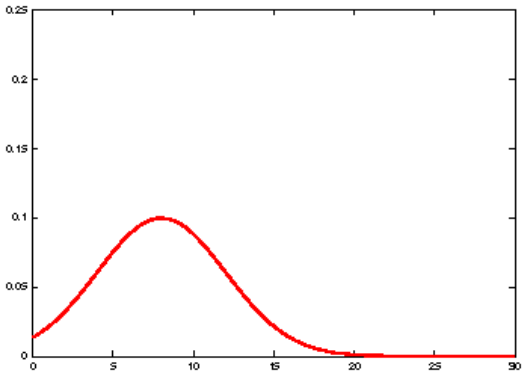
$$x_t = Ax_{t-1} + B_t u_t + \varepsilon_t$$

  - Model senzoru musí být lineární s Gausovým šumem

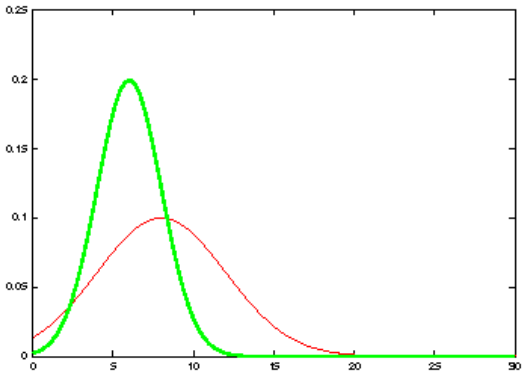
$$z_t = Cx_t + \delta_t$$

  - Počáteční informace (víra) musí mít normální rozložení

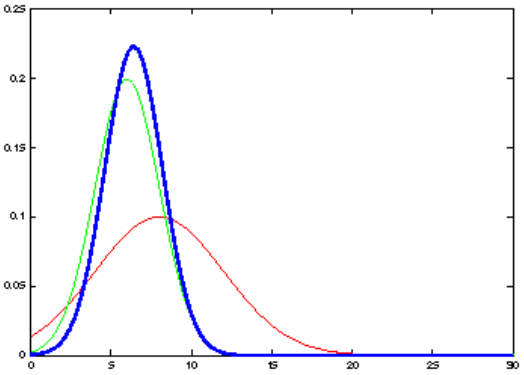
Kalmanův filtr



Apriorní znalost

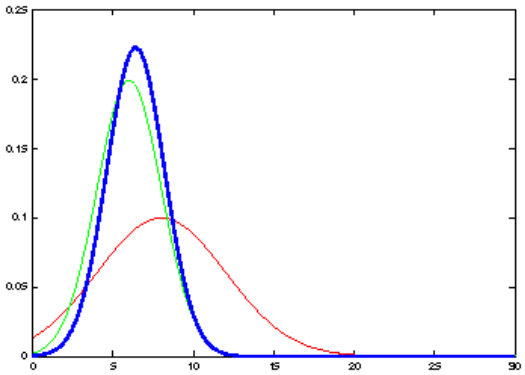


Nové měření

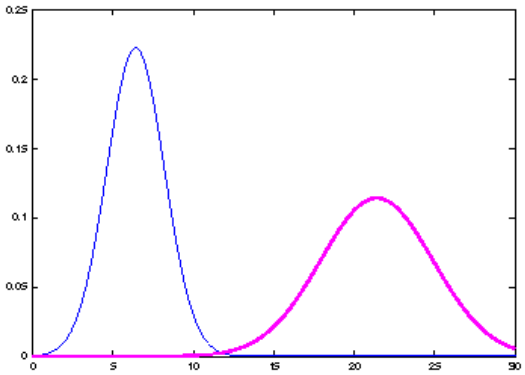


Integrace nového měření

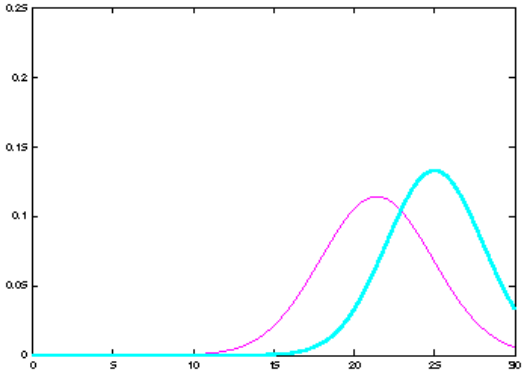
Kalmanův filtr



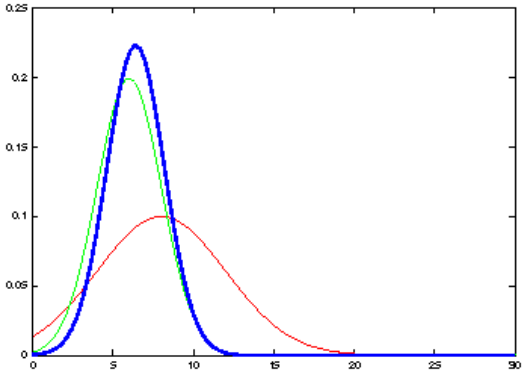
Aktuální stav



Provedení akce



Nové měření



Integrace nového měření

Algoritmus

Algoritmus Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)

Predikce (integrace akce)

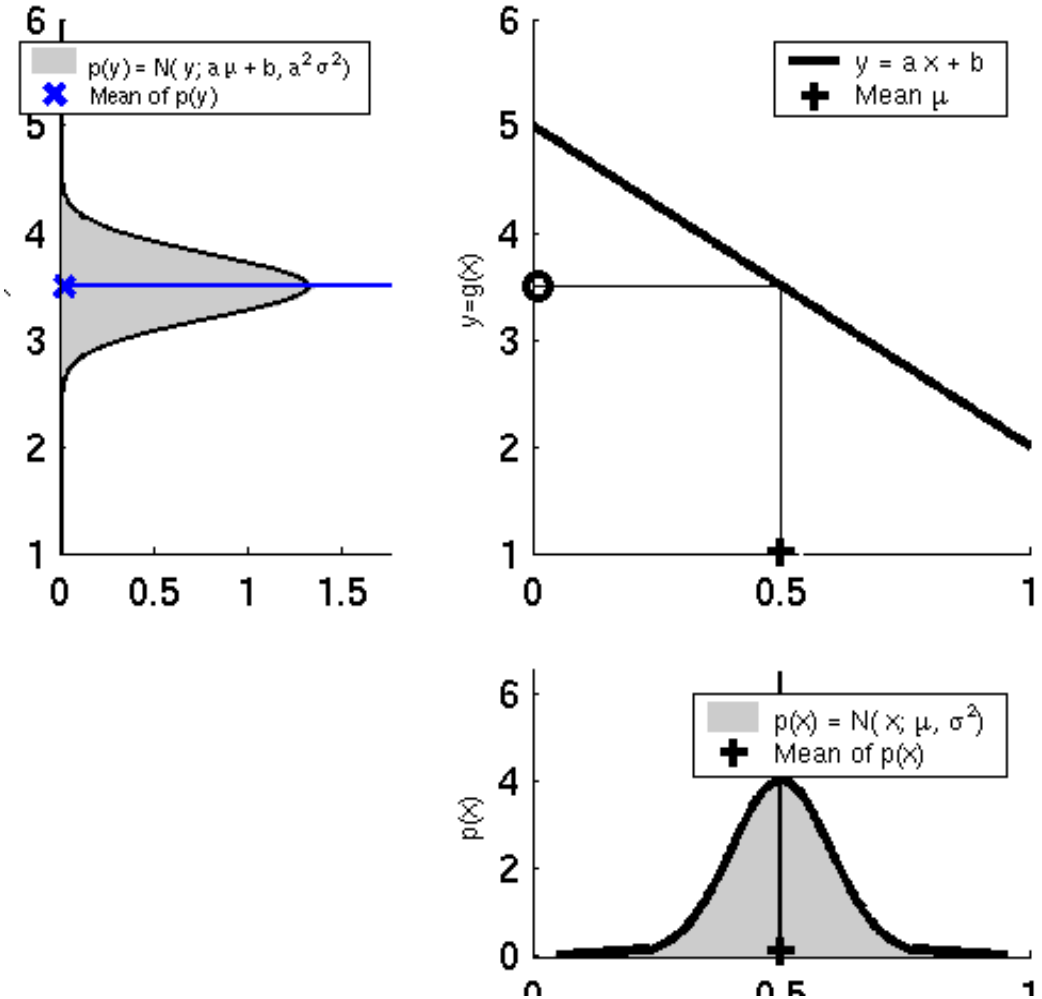
$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

Korekce (integrace měření)

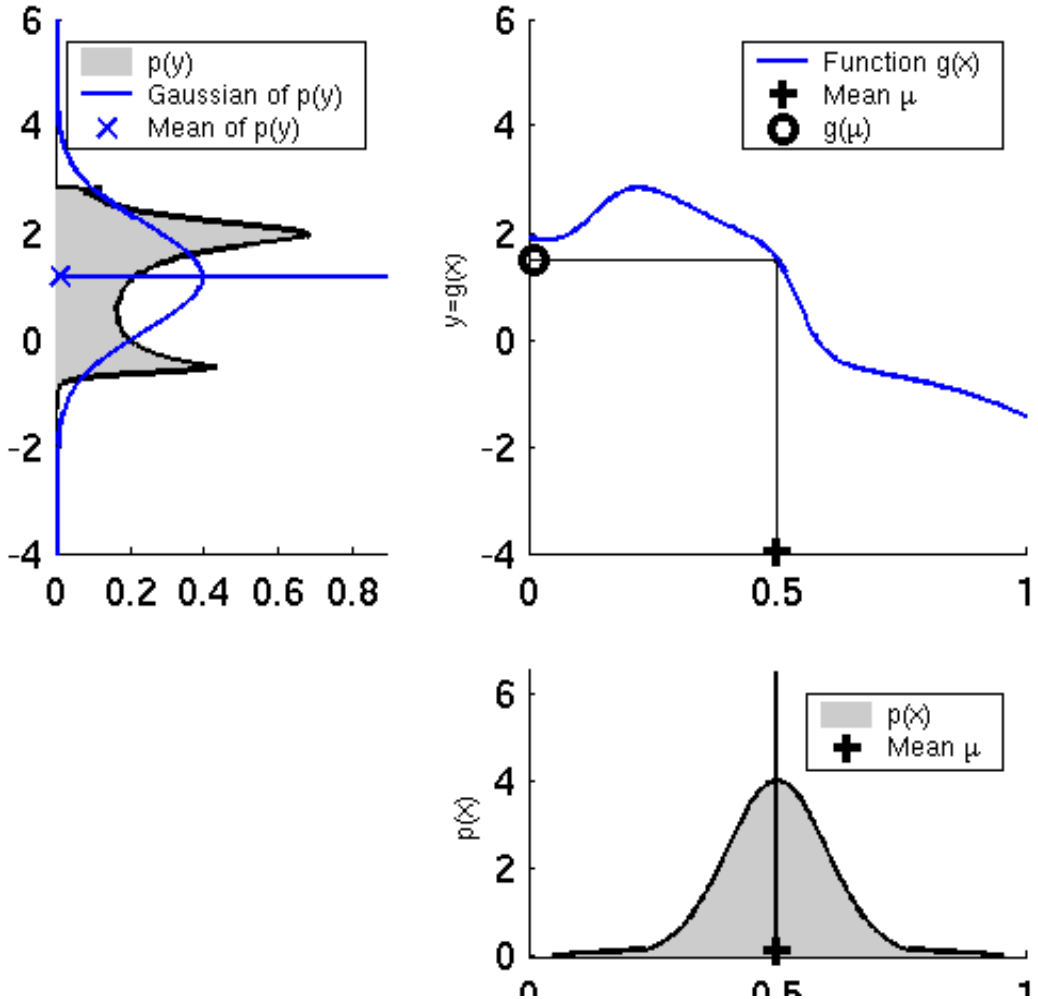
$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-}1$$
$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (E - K_t C_t) \bar{\Sigma}_t$$

return μ_{t,Σ_t}

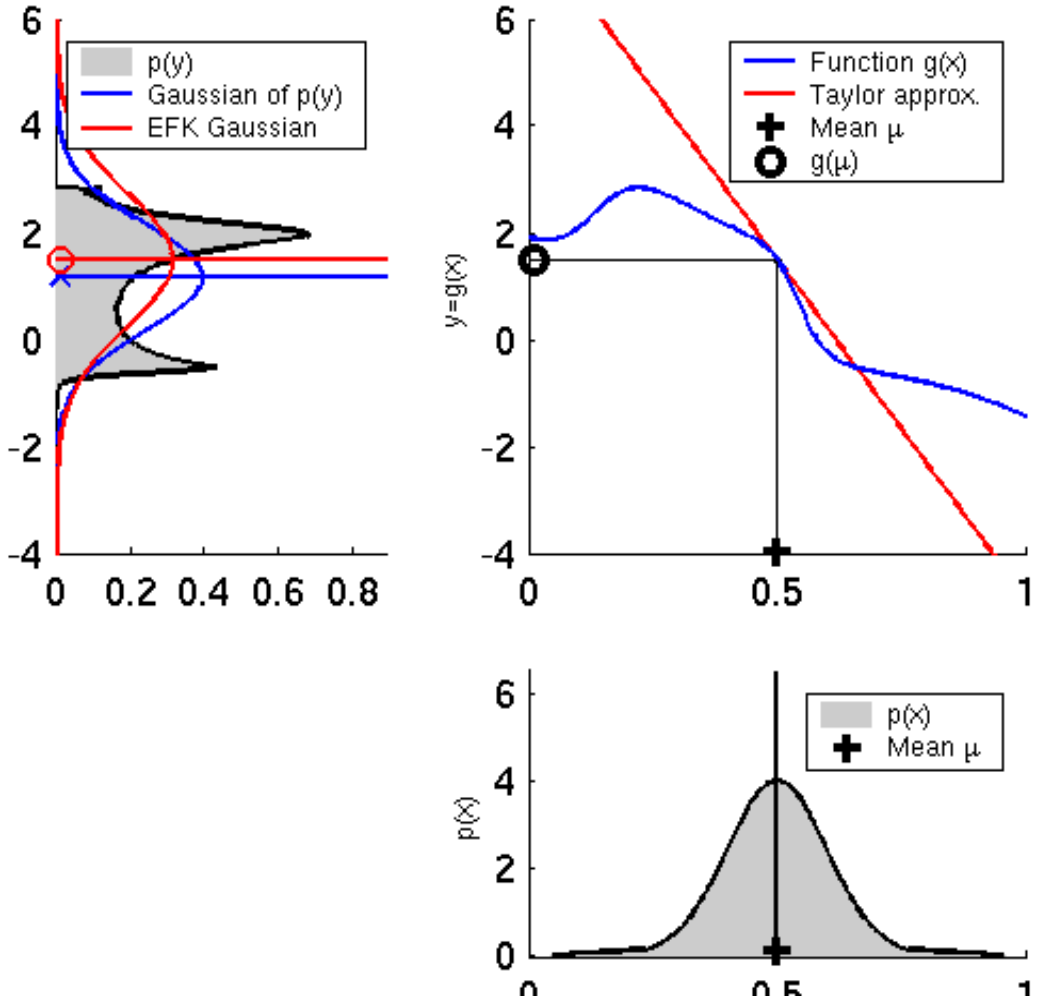
Lineární transformace Kalmanova filtru



Nelineární transformace



Rozšířený Kalmanův filtr



$$
\begin{aligned}
x_t &= g(u_t, x_{t-1}) \\
z_t &= h(x_t)
\end{aligned}
$$

Rozšířený Kalmanův filtr

- Pro linearizaci používáme Taylorův rozvoj v μ, tj. matice derivací funkcí – Jakobiány
- Predikce:

$$\begin{aligned} g(u_t, x_t - 1) &\approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1}) \\ g(u_t, x_t - 1) &\approx g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1}) \end{aligned}$$

- Korekce:

$$\begin{aligned} h(x_t) &\approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t}(x_t - \bar{\mu}_t) \\ h(x_t) &\approx h(\bar{\mu}_t) + H_t(x_t - \bar{\mu}_t) \end{aligned}$$

Algoritmus Extended_Kalman_filter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)

Predikce (integrace akce)

$$\bar{\mu}_t = g(u_t, \mu_{t-1})$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$$

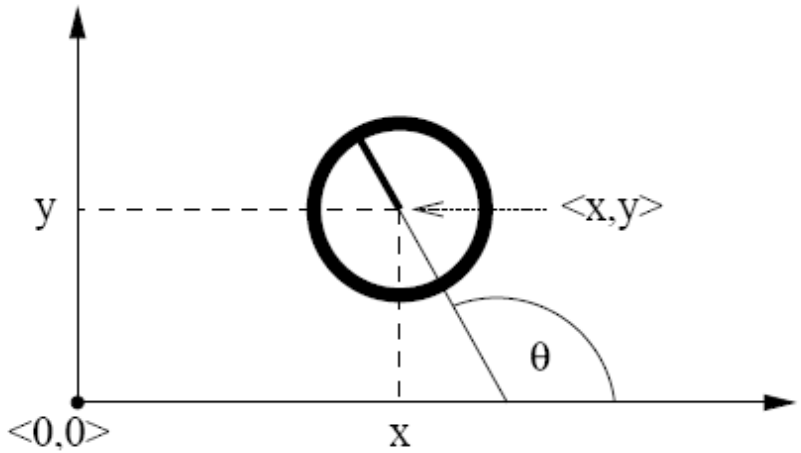
Korekce (integrace měření)

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-}1$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t)$$

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$$
$$G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

Model pohybu

- Pro implementaci Bayesova filtru potřebujeme definovat model pohybu $p(x|x', u)$.
- Model pohybu definuje pravděpodobnost, že po provedení akce u ve stavu x' bude robot ve stavu x.Předpokládáme robot pohybující se v rovině, tj $x = \langle x, y, \phi \rangle$
- Existují různé modely (v závislosti na typu řízení, zahrnutí kinematiky, atd.)

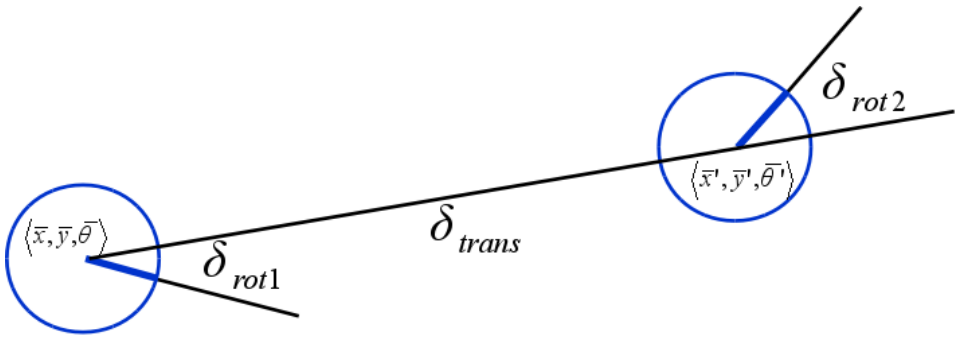


Odometrický model pohybu

Ideální případ

- Robot se pohybuje z $\langle \bar{x}, \bar{y}, \bar{\phi} \rangle$ do $\langle \bar{x}', \bar{y}', \bar{\phi}' \rangle$
- Z odometrie získáme $u = \langle \delta_{rot_1}, \delta_{rot_2}, \delta_{trans} \rangle$

$$
\begin{aligned}
\delta_{trans} &= \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2} \\
\delta_{rot_1} &= atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\phi} \\
\delta_{rot_2} &= \bar{\phi}' - \bar{\phi} - \delta_{rot_1}
\end{aligned}
$$



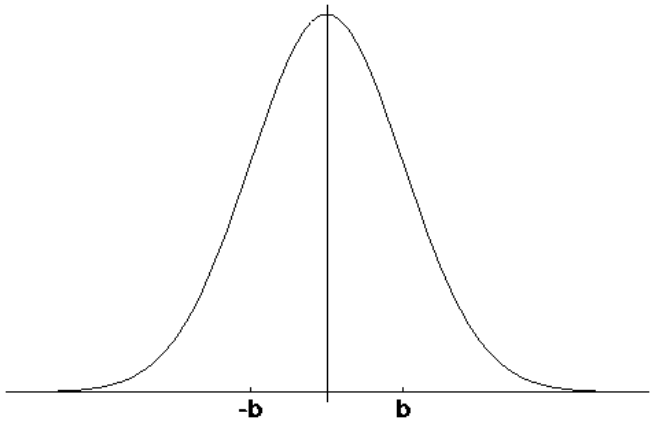
Přidáme šum

$$
\begin{aligned}
\hat{\delta}_{rot_1} &= \delta_{rot_1} + \varepsilon_{\alpha_1|\delta_{rot_1}|+\alpha_2|\delta_{trans}|} \\
\hat{\delta}_{trans} &= \delta_{trans} + \varepsilon_{\alpha_3|\delta_{trans}|+\alpha_4(|\delta_{rot_1}|+|\delta_{rot_2}|)} \\
\hat{\delta}_{rot_2} &= \delta_{rot_2} + \varepsilon_{\alpha_1|\delta_{rot_2}|+\alpha_2|\delta_{trans}|}
\end{aligned}
$$

- Šum je dán čtyřmi parametry
- Na celém modelu je nejtěžší získat právě parametry šumu – experimentálně

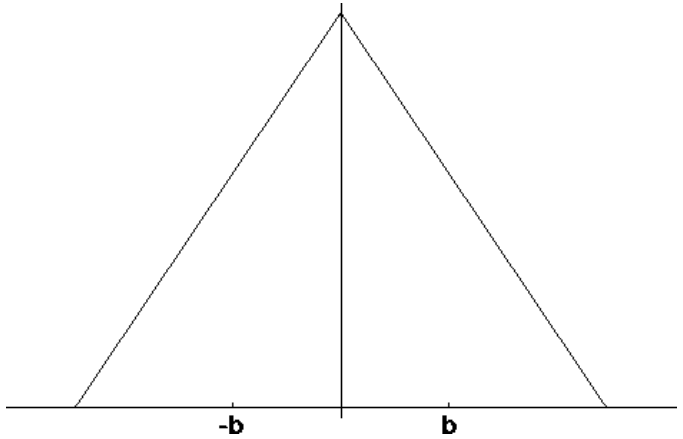
Typické distribuce šumu

Normální



$$\varepsilon_{\sigma 2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Trojúhelníková



$$\varepsilon_{\sigma 2}(x) = \begin{cases} 0 & \text{pokud } |x| > \sqrt{6\sigma^2} \\ \frac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} & \text{jinak} \end{cases}$$

Odometrický model pro vzorkování

$$u = \langle \delta_{rot_1}, \delta_{rot_2}, \delta_{trans} \rangle, x = \langle x, y, \phi \rangle => x' = \langle x', y', \phi' \rangle$$

Náhodné řízení

$$\begin{aligned}
\hat{\delta}_{rot_1} &= \delta_{rot_1} + sample(\alpha_1 |\delta_{rot_1}| + \alpha_2 |\delta_{trans}|) \\
\hat{\delta}_{trans} &= \delta_{trans} + sample(\alpha_3 |\delta_{trans}| + \alpha_4 (|\delta_{rot_1}| + |\delta_{rot_2}|)) \\
\hat{\delta}_{rot_2} &= \delta_{rot_2} + sample(\alpha_1 |\delta_{rot_2}| + \alpha_2 |\delta_{trans}|)
\end{aligned}$$

Výpočet pozice

$$\begin{aligned}
x' &= x + \hat{\delta}_{trans} + cos(\phi + \hat{\delta}_{rot_1}) \\
y' &= x + \hat{\delta}_{trans} + sin(\phi + \hat{\delta}_{rot_1}) \\
\phi' &= \phi + \hat{\delta}_{rot_1} + \hat{\delta}_{rot_1} \\
&\text{return } \langle x', y', \phi' \rangle
\end{aligned}$$

sample (normální rozložení): $\frac{1}{2} \sum_{i=1}^{12} rand(-b, b)$

Výpočet $p(x|u, x')$

Hodnoty z odometrie (u)

$$\begin{aligned} \delta_{trans} &= \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2} \\ \delta_{rot_1} &= atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\phi} \\ \delta_{rot_2} &= \bar{\phi}' - \bar{\phi} - \delta_{rot_1} \end{aligned}$$

Ideální případ

$$\begin{aligned} \hat{\delta}_{trans} &= \sqrt{(x' - x)^2 + (y' - y)^2} \\ \hat{\delta}_{rot_1} &= atan2(y' - y, x' - x) - \phi \\ \hat{\delta}_{rot_2} &= \phi' - \phi - \delta_{rot_1} \end{aligned}$$

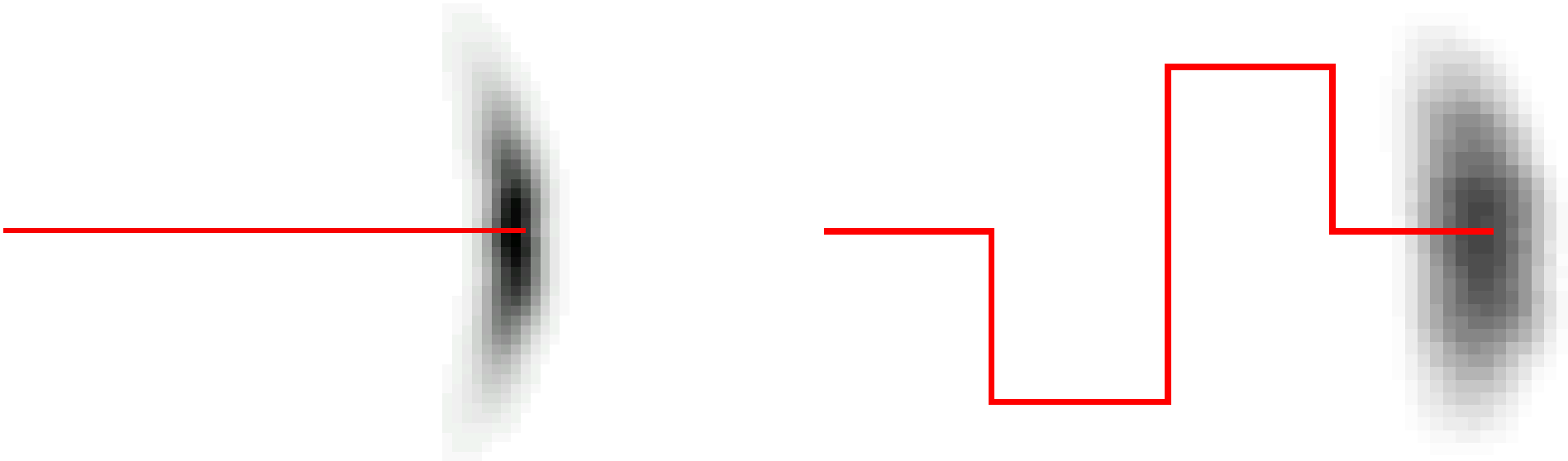
Výpočet pravděpodobnosti

$$\begin{aligned} p_1 &= prob(\delta_{rot_1} - \hat{\delta}_{rot_1}, \alpha_1|\delta_{rot_1}| + \alpha_2\delta_{trans}) \\ p_2 &= prob(\delta_{trans} - \hat{\delta}_{trans}, \alpha_3\delta_{trans} + \alpha_4(|\delta_{rot_1}| + |\delta_{rot_2}|) \\ p_3 &= prob(\delta_{rot_2} - \hat{\delta}_{rot_2}, \alpha_1|\delta_{rot_2}| + \alpha_2\delta_{trans}) \end{aligned}$$

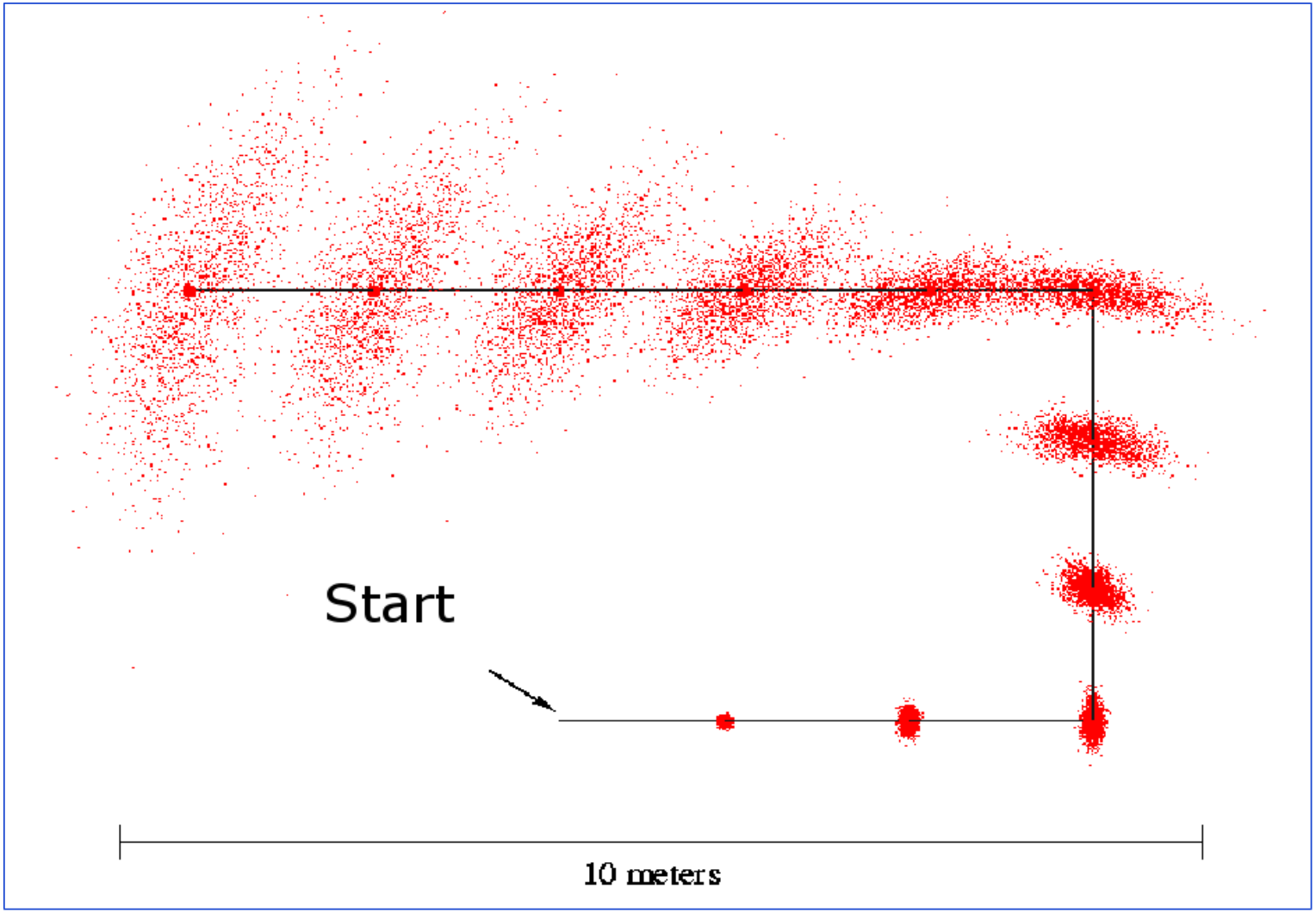
return $p_1 p_2 p_3$ $\longleftarrow$ předpokládáme nezávislost

Aplikace

- Výsledná hustota pravděpodobnosti závisí na ujeté trajektorii, ne pouze na výsledné pozici!
- Pro složitější pohyby se opakovaně provádí předchozí algoritmus.
- Typický případ, jak vypadá distribuce (2D projekce)

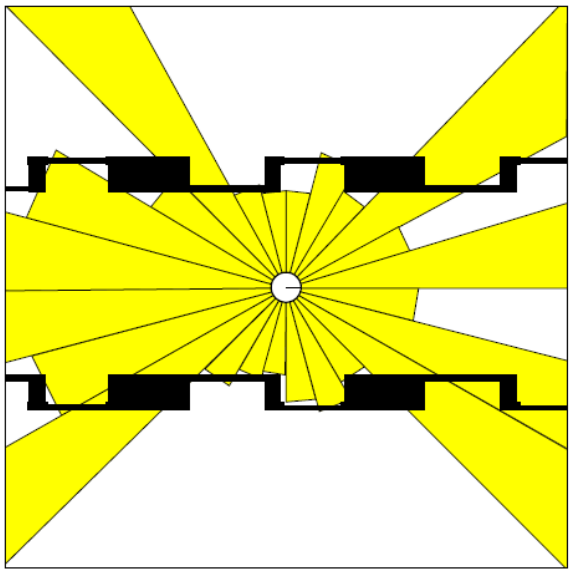


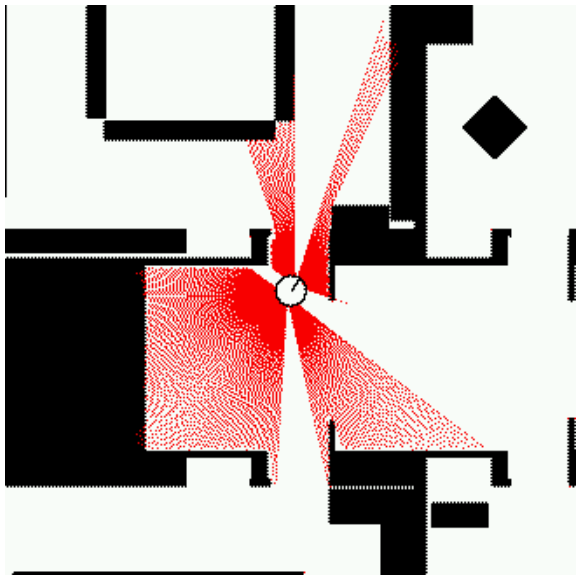
Skládání trajektroií



Model senzoru

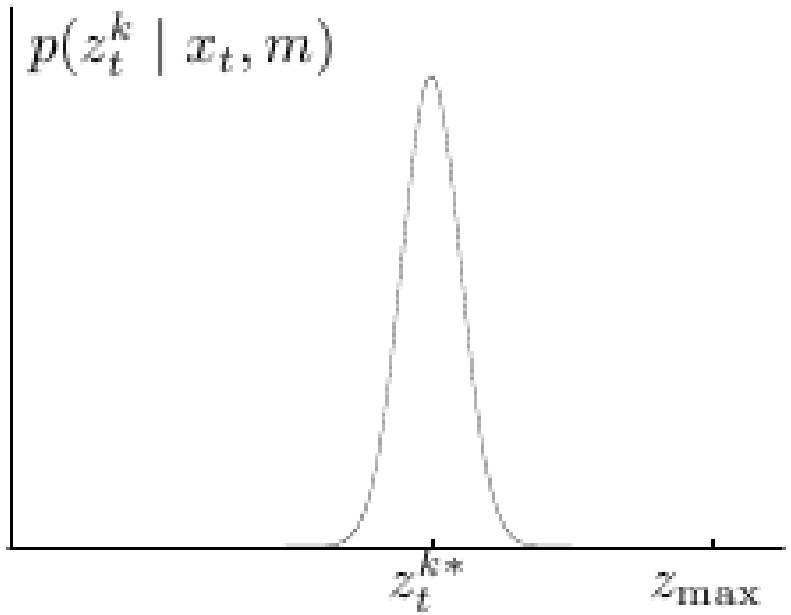
- Cílem je určit $p(z|m, x)$.
- Budeme používat proximitní senzory (laser).
- Scan se skládá z k měření (paprsků): $z = \{z_1, z_2, \ldots, z_k\}$
- Předpokládáme, že paprsky jsou nezávislé: $P(z|x, m) = \prod_{k=1} k|x, m$





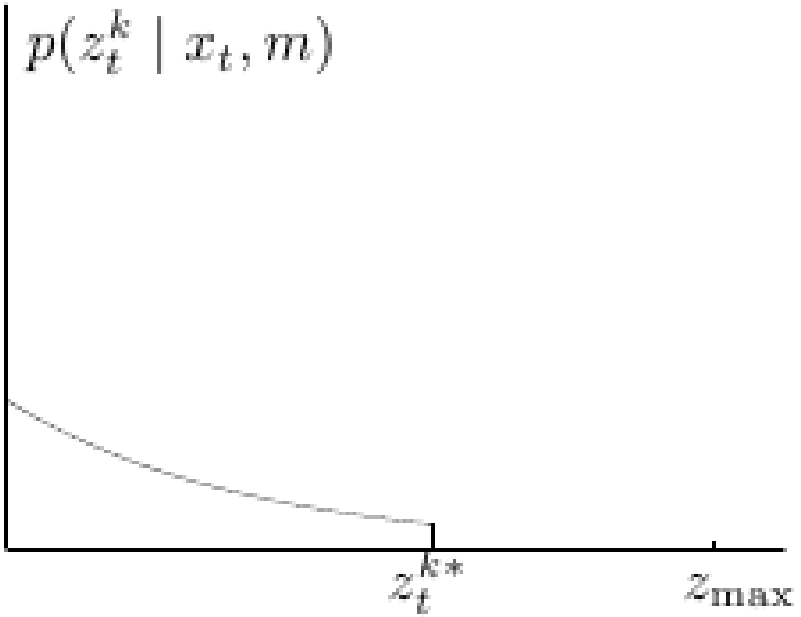
Paprskový model - komponenty

Normální rozložení



$$p_{hit}(z|x,m) = \begin{cases} \eta N(z, z^*, \sigma^2_{hit}) & \text{pokud } 0 \leq z \leq z_{max} \\ 0 & \text{jinak} \end{cases}$$

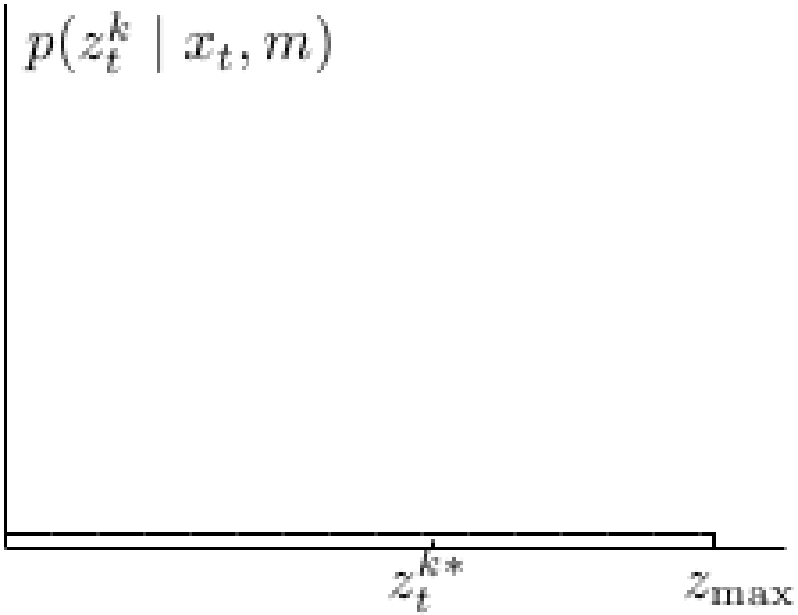
Poisonovo rozložení



$$p_{short}(z|x,m) = \begin{cases} \eta \lambda_{short} e^{-\lambda_{short} z} & \text{pokud } 0 \leq z \leq z^* \\ 0 & \text{jinak} \end{cases}$$

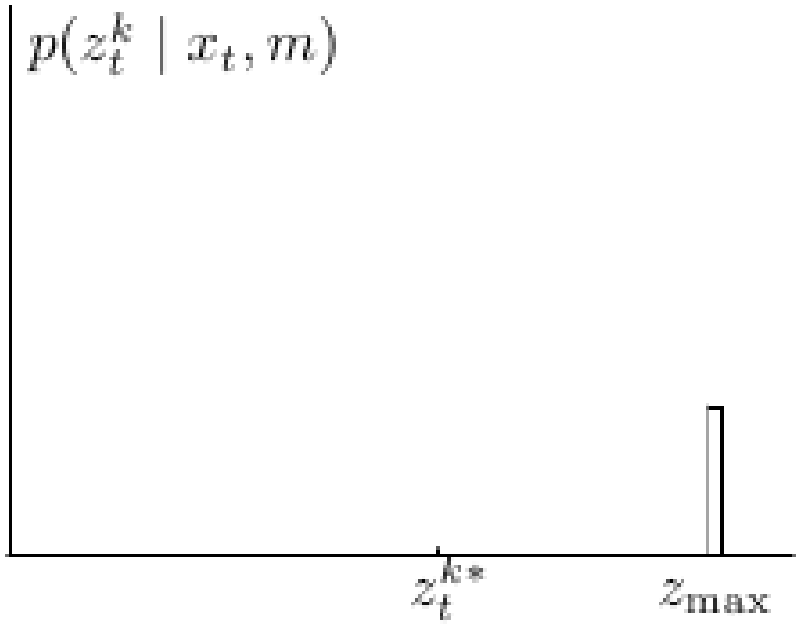
Paprskový model - komponenty

Rovnoměrné rozložení



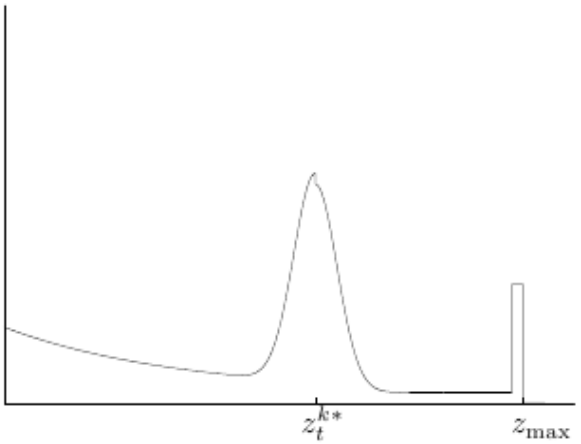
$$p_{rand}(z|x,m) = \begin{cases} \frac{1}{z_{max}} & \text{pokud } 0 \leq z \leq z_{max} \\ 0 & \text{jinak} \end{cases}$$

Diskrétní rozložení :-(



$$p_{max}(z|x,m) = \begin{cases} 1 & \text{pokud } z = z_{max} \\ 0 & \text{jinak} \end{cases}$$

Sloučení



$$p(z|x,m) = \begin{pmatrix} \alpha_{hit} \\ \alpha_{short} \\ \alpha_{rand} \\ \alpha_{max} \end{pmatrix}^T \begin{pmatrix} p_{hit}(z|x,m) \\ p_{short}(z|x,m) \\ p_{rand}(z|x,m) \\ p_{max}(z|x,m) \end{pmatrix}$$

- Parametry modelu se určí nějakým učícím algoritmem na základě naměřených dat(E-M, genetické algoritmy)
- z_{exp} se počítá raytracingem, což je časově náročné.
- Lze počítat jen některé paprsky (např. 8); navíc větší nezávislost.
- Lze použít předpočítané mapy (pro každé $\langle x, y, z\phi \rangle$)
- Vynásobením modelu senzoru $\lambda < 1$ omezíme vliv senzoru.
- Model je nespojitý $=>$ aproximační vyjádření hustoty pp. může minout daný stav.

Sloučení

V prezentaci byly použity obrázky z prezentací Sebastiana Thruna (http://probabilisticrobotics.org/).