

Lokalizace v mobilní robotice

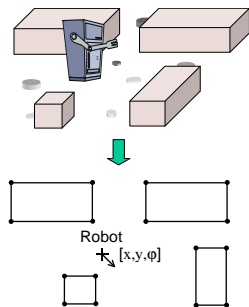
Miroslav Kulich

Intelligent and Mobile Robotics Group
Gerstner Laboratory for Intelligent Decision Making and Control
Czech Technical University in Prague

10/10/2012

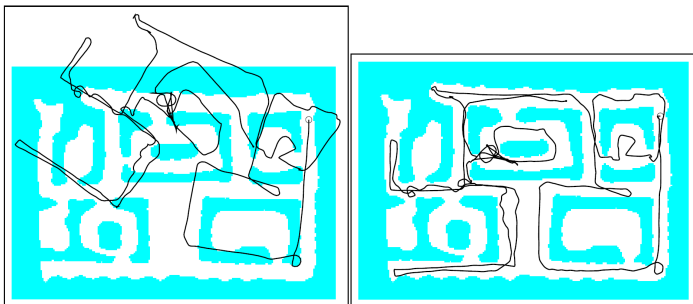
Lokalizace

- 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 \Rightarrow pozice je vypočtena ze sensorických dat
- Ale senzory jsou nepřesné!
- Příkazy jsou prováděny nepřesně
- Jedno sensorické měření k určení polohy nestačí



Taxonomie

- **Lokální** (kontinuální, „position tracking“)
 - Počáteční pozice je známa
 - Korekce odometrických chyb
 - Omezená chyba
 - Unimodální distribuce (Gaussian)
- **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

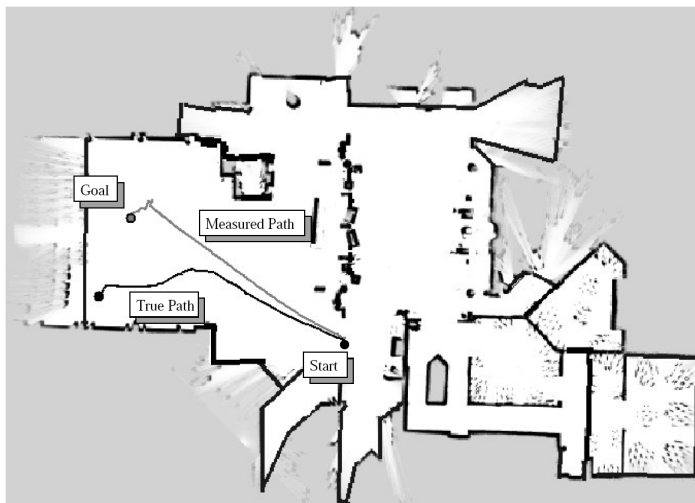
Taxonomie

- **Statické prostředí**
 - Robot je jediný, co se pohybuje
 - Pozice 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 \Rightarrow tvoří část stavového popisu úlohy
 - Data o pohyblivých objektech jsou filtrována

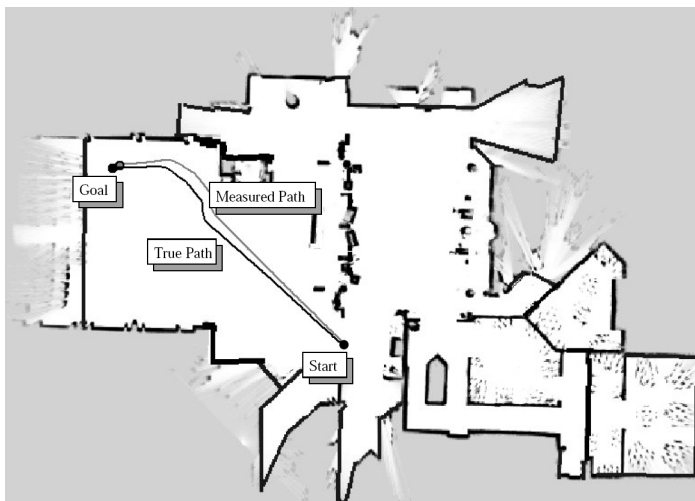
Taxonomie

- **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

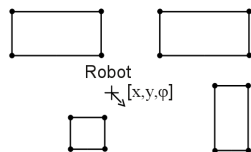


Taxonomie

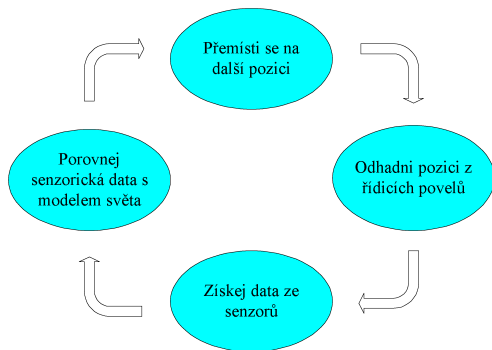
- **Jeden robot**
 - Klasický problém
 - Není nutná žádná komunikace, vše v jednom
- **Více robotů**
 - Každý robot se lokalizuje samostatně (tváří se, že ostatní neexistují)
 - Ale pokud jsou roboty schopny se navzájem detekovat, je to lepší
 - Lokalizace sebe sama v mapě druhého robotu Zním svoji polohu, 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

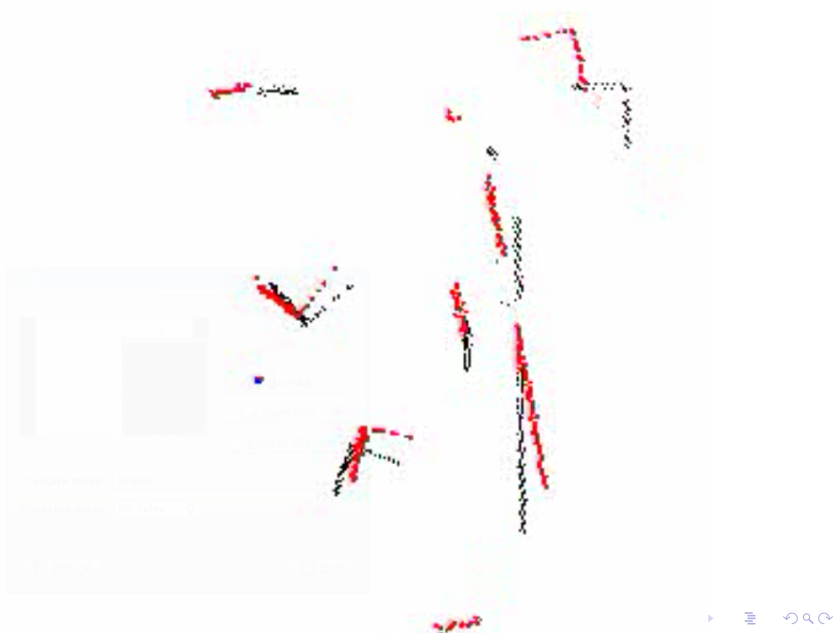


Kontinuální lokalizace



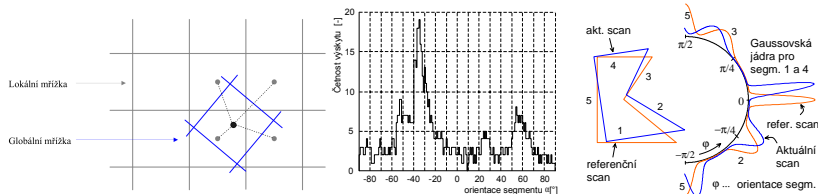
- Poloha se hledá pouze v nejbližším okolí odhadnuté pozice
- Vyžaduje se vyšší přesnost
- Vstup: lokální a globální mapa (scan)
- Výstup: transformace minimalizující míru nesouladu map

Ukázka



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)

(Lu, Milios)

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

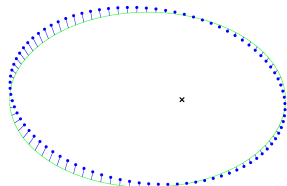
$$E_{dist}(T_x, T_y, \omega) = \sum_i^n |R_\omega P + T - P'|$$

- Má analytické řešení

$$T_x = \bar{x}' - (\bar{x} \cos(\bar{\omega}) - \bar{y} \sin(\bar{\omega}))$$

$$T_y = \bar{y}' - (\bar{x} \sin(\bar{\omega}) + \bar{y} \cos(\bar{\omega}))$$

$$\bar{\omega} = \arctan \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i' - \bar{y}') - \sum_{i=1}^n (y_i - \bar{y})(x_i' - \bar{x}')}{\sum_{i=1}^n (x_i - \bar{x})(y_i' - \bar{y}') + \sum_{i=1}^n (y_i - \bar{y})(x_i' - \bar{x}')}$$

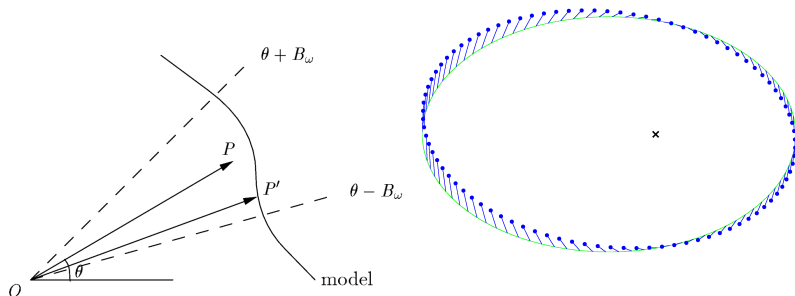


- Konverguje pomalu, špatné v rotační složce

Iterative Matching Range Point (IMRP)

(Lu, Milios)

- Postup stejný jako u ICP, pouze jiné pravidlo korespondence
- Předpokládáme, že posunutí je nulové $\Rightarrow |P| \approx |P'|$ 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 \Rightarrow nestabilita

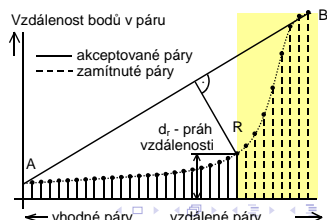


Iterative Dual Correspondence (IDC)

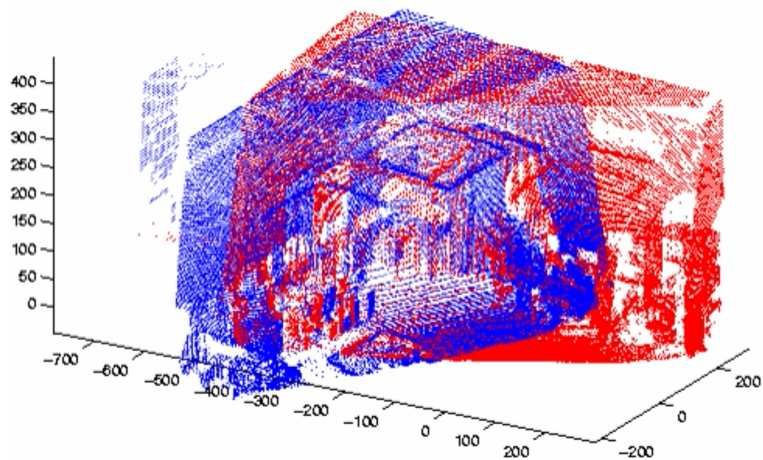
(Lu, Milios)

- Kombinace obou pravidel: posunutí z ICP, otočení z IMRP
- **Algoritmus**
 - Pro každý bod P_i z aktuálního scanu:
 - Použij pravidlo nejbližšího bodu pro nalezení korespondujícího bodu P'
 - Použij pravidlo MRP pro nalezení korespondujícího bodu P''
 - Spočítej metodou nejmenších čtverců (ω_1, T_1) z korespondencí (P, P')
 - Spočítej metodou nejmenších čtverců (ω_2, T_2) z korespondencí (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



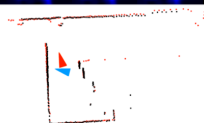
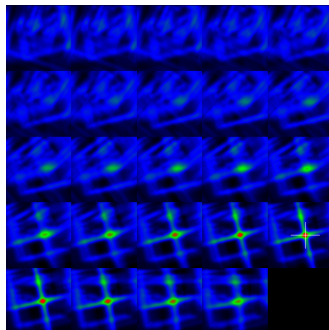
Lokalizace ve 3D



Correlative Scan Matching

(Edwin B. Olson)

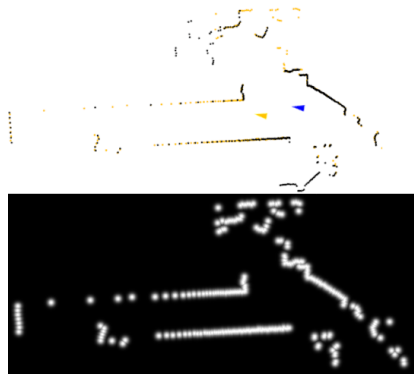
- Hodnotící funkce je typicky velmi komplikovaná s množstvím lokálních minim.
- Počáteční odhad pozice (odomoterie) může být velmi nepřesný.
- ICP nalézá pouze lokální optimum
- Vstup: dva scany
- Cíl: určit globální optimum



Correlative Scan Matching

Tvorba vyhledávací tabulky

- Pro každý bod v mapě se spočítá pravděpodobnost, že odpovídá měření referenčního scanu.



Correlative Scan Matching

Algoritmus

- Brutální síla
 - Pro každou pozici scanu (včetně orientace) se spočítá korelace.
 - Vybere se pozice s největší korelací .
- Výpočet 2D řezů
 - Spoustu času se tráví transformací scanu.
 - Idea: otočení se provede ve vnější smyčce, v ostatních se již jen posouvá.

Correlative Scan Matching

Algoritmus

- Více-úrovnové vyhledávání
 - Dvě vyhledávací tabulky: s nízkým (NVT) a vysokým (VVT) rozlišením.
 - Hodnota bunky v NVT je maximem odpovídajících buněk VVT.
 - To zajišťuje, že hodnota NVT je nejméně tak velká jako ve VVT.
 - Algoritmus
 - Spočítej korespondence pro NVT.
 - Najdi nejlepší bunku NVT_B v NVT, která dosud nebyla zpracována.
 - Pokud její hodnota je menší, než nejlepší dosud nalezená skonči.
 - Najdi bunku s nejlepší hodnotou v VVT, která odpovídá NVT_B
 - Pokud je nalezená hodnota lepší než aktuální maximum, nastav maximum na tuto hodnotu.

Correlative Scan Matching

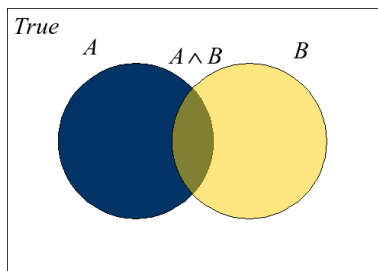
Výsledky

PC: Intel Core2 6600 @ 2.4 Ghz

| | ICP | ICL | Hill- Climb | Correlative (Naive) | Correlative (2D Slices) | Correlative (Multi-Res) | Correlative (GPU 7600GS) | Correlative (GPU GTX260) |
|---------------|--------|--------|----------------|------------------------|----------------------------|----------------------------|-----------------------------|-----------------------------|
| 0.5 m, 20 deg | 56 ms | 64 ms | 1.1 ms | 246 ms | 27 ms | 8.4 ms | 98 ms | 26 ms |
| 2.0 m, 40 deg | 99 ms | 105 ms | 1.3 ms | 7512 ms | 692 ms | 20.8 ms | 1563 ms | 289 ms |
| 4.0 m, 90 deg | 145 ms | 174 ms | 1.0 ms | 65282 ms | 5029 ms | 86.1 ms | 13166 ms | 2012 ms |

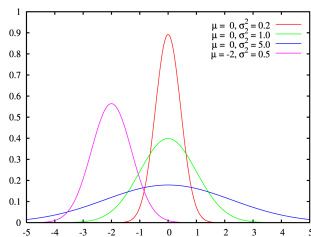
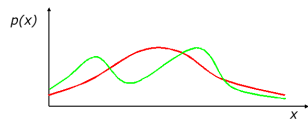
Lehký úvod do teorie pravděpodobnosti

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



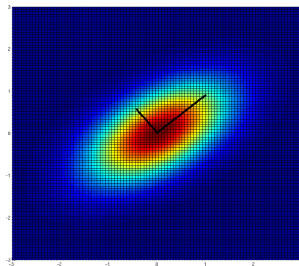
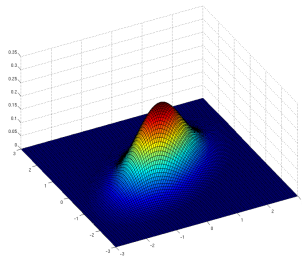
Diskrétní a spojitá náhodná veličina

- **Diskrétní:** X je spočetná, tj.
 $X = x_1, x_2, \dots, x_n$
- **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

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

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

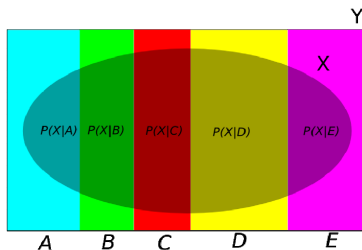
$$p(x) = \sum_y p(x|y)p(y)$$

Spojitéý 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

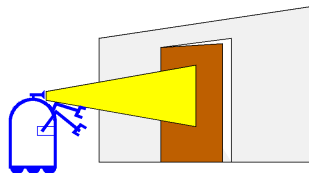
$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{\textit{likelihood} \cdot \textit{prior}}{\textit{evidence}}$$

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

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

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 - opetvřené 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

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

$$\begin{aligned} p(open|z_2z_1) &= \frac{p(z_2|open)p(open|z_1)}{p(z_2|open)p(open|z_1) + p(z_1|\neg open)p(\neg open|z_1)} \\ &= \frac{\frac{1}{2} \cdot \frac{2}{3}}{\frac{1}{2} \cdot \frac{2}{3} + \frac{3}{5} \cdot \frac{1}{3}} = \frac{5}{8} = 0.625 \end{aligned}$$

- z_2 snižuje pravděpodobnost, že jsou dveře otevřené.

Akce

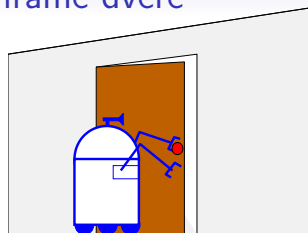
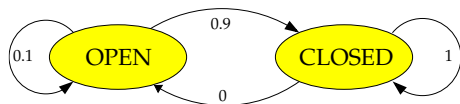
- Svět je dynamický
 - robot se hýbe,
 - jiné objekty v prostředí se hýbou
 - a čas prostě plyne (květiny rostou).
- Akce se nikdy nestane tak, jak byla naplánována.
- Na rozdíl od měření akce obvykle zvýší míru nejistoty.
- Pro integraci akce u do aktuální „víry“ použijeme podmíněnou pravděpodobnost

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

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

Pokračování příkladu - zavíráme dveře

$p(x|u, x')$ pro $u = \text{„zavři dveře“}$



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

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

Pokračování příkladu - zavíráme dveře

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

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