Bayesovské programovanie robotov

BAKALÁRSKA PRÁCA

Jakub Kondela

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ INFORMATIKY

Študijný odbor: 9.2.9 Aplikovaná Informatika

Vedúci:Mgr. Pavel Petrovič, PhD.

BRATISLAVA 2009

Čestné prehlásenie

Prehlasujem, že som túto prácu napísal sám, s použitím uvedenej literatúry, a s využitím teoretických poznatkov získaných počas môjho štúdia, a potvrdzujem to svojím podpisom.

Bratislava, 2009

…..........................

Jakub Kondela

Poďakovanie

Ďakujem hlavne svojím rodičom, ktorí mi umožnili štúdium na vysokej škole, ďakujem aj za neustálu morálnu podporu počas štúdia od všetkých členov rodiny. Ďakujem Mgr. Pavlovi Petrovičovi, Phd za cenné rady a pripomienky, ktoré mi pomohli pri písaní tejto práce.

ABSTRAKT

Kondela, Jakub : Bayesovské programovanie robotov. [bakalárska práca]. Univerzita Komenského v Bratislave, Fakulta Matematiky,Fyziky a Informatiky,
Katedra slovenských dejín. Školiteľ : Mgr. Pavel Petrovič, PhD. Komisia pre obhajoby : .... Predseda...
Stupeň odbornej kvalifikácie : bakalár (Bc.), Bratislava: FMFI UK 2008, 51 s.

Obsah : Obsahom mojej práce bolo naštudovať si základné princípy Bayesovského programovania robotov. Oboznámiť sa s problémom aplikovania robotov do reálnom prostredia. Ako aj vyhotoviť jednoduchý experiment na reálnom robotovi.

Kľúčové slová : Bayesovské programovanie robotov,podmienená pravdepodobnosť,robotika.

Predhovor

Bayesovské programovanie robotov je metóda na riešenie nepresnosti a nejasnosti bázy znalosti robota v reálnom dynamickom prostredí. Používa základné princípy pravdepodobnosti a logiky na reprezentáciu bázy znalosti robota. Táto metóda bola publikovaná v článku *Bayesian Robot Programming* [1], kde boli popísané základne princípy tohto prístupu ako aj samostatný framework. V práci som sa zameral na podrobný popis jednotlivých časti tvorby programu. Ďalšou úlohou v mojej práci bolo realizovať jednoduchý experiment na reálnom robotovi.

OBSAH

1.Úvod

Robotika v súčasnosti prináša stále nové postupy na riešenie problémov v ich aplikácií na reálne prostredie. Predpokladáme, že každý model vytvorený v takomto prostredí je nekompletný. Vždy sa vyskytujú nejaké udalosti, ktoré tento model nepozná. Preto je potrebné navrhnúť riešenie tohto problému. Veľmi účinným prístupom k tejto problematike je obaliť celú bázu znalosti robota pravdepodobnosťou. Ako jedným z pravdepodobnostným prístupov k riešeniu takéhoto problému je práve Bayesovské programovanie robotov. Hlavnou zložkou Bayesovského programovanie je program, ktorý používa stále ten istý framework.

2. Základné pojmy

Na úvod sa treba oboznámiť so základnými pojmami, pravidlami, definíciami pravdepodobnosti a logiky . Neskôr tieto poznatky môžeme použiť pri konkrétnych postupoch tvorby Bayesovského programu.

2.1. Logika

V tejto sekcii sú definované základné pojmy z logiky dôležité pre Bayesovský program. Logika nielen v robotike je dôležitá pre jej jednoduchú a presnú matematickú formuláciu a symbolické pomenovanie daných javov v prostredí.

2.1.1. Výrok

Definícia: Výrokom nazývame každú oznamovaciu vetu, o ktorej má zmysel hovoriť, či je pravdivá alebo nepravdivá. Každý výrok má len jednu pravdivostnú hodnotu.

Označenie malými písmenami.(a,b,c,...)

$$a \in \{0,1\}$$

Logický výrok v našom prípade bude väčšinou reprezentovaný danou hodnotou senzora
(senzor *nameral hodnotu 30cm, robot je vzdialený od predmetu 10 jednotiek*)

Logické výroky môžu byť ďalej vytvorené pomocou logických operátorov :

- konjunkcia $a \wedge b$
- disjunkcia $a \vee b$
- negácia $\neg a$

Použitím týchto operátorov na logické výroky sa vytvárajú nové log. výroky.

2.2 Pravdepodobnosť

Logika sa ukázala ako veľmi dobre riešenie pri formulácií javov v prostredí. Problémom teraz je naše prostredie, ktoré je skoro vo väčšine prípadov v robotike dynamické. Čo ešte nieje veľký problém pre logiku. Prvým problémom sú samostatné logické výroky v robotike, ktoré sú definované nejakými hodnotami senzora. A tieto hodnoty môžu byť nepresné. Ďalším problémom je samostatná báza znalosti robota, ktorá nemusí obsahovať všetky možnosti situácií z definovaných premenných pre dané prostredie. Ako riešenie týchto dvoch problémov je práve pravdepodobnosť. Nato aby sme mohli riešiť problém nejasnosti, nepresnosti okolitého sveta musíme aplikovať na logické výroky pravdepodobnosť. Najskôr zadefinujeme základné pojmy. Pravdepodobnosť je funkcia, ktorá počíta pomer výskytu danej udalosti a všetkých udalosti.

2.2.1 Náhodná premenná

<u>Definícia:</u> Označenie resp. značka, ktorá reprezentuje obor určitých hodnôt.
Označenie veľkými začiatočnými písmenami.(Smer, Vzdialenosť, X,Y,Z,...)

V závislosti od oboru hodnôt náhodnej premennej (ďalej len premenná) rozdeľujeme ju na *diskrétnu* alebo *spojitú premennú*.
Diskrétna nadobúda spočítateľné množstvo hodnôt. Nadobúda izolované, väčšinou celo číselné hodnoty.(Počet hodnôt senzora z uzavretej množiny.)
Spojitá môže nadobúdať ľubovoľné hodnoty z ohraničeného alebo neohraničeného intervalu. (Počet chýb merania vzdialenosti v mm)

Hodnoty, ktoré nadobúda daná premenná, budú charakterizované logickými výrokmi. Preto v ďalšom texte hodnota premennej je ekvivalentná s pojmom logický výrok premennej. Teda ako príklad môžem uviesť premennú Vzdialenosť, ktorá bude nadobúdať hodnoty 0 až 10

Diskrétna premenná *X* je množina logických výrokov x_i a pre tieto výroky platí:

- navzájom sa vylučujú (pre všetky *i* a *j* platí , $x_i \neq x_j : (x_i \wedge x_j) = 1$)
- sú úplné (aspoň jeden výrok je pravdivý)

X_i znamená, že premenná X nadobúda í-tu hodnotu.

$[\![X]\!]$ znamená počet logických výrokov v X

<u>Použitie logických operátorov:</u>

Konjunkcia dvoch premenných: $X \wedge Y$

Je definovaná ako množina $[\![X]\!] * [\![Y]\!]$ výrokov $x_i \wedge y_j$ a výroky sa navzájom vylučujú a sú úplné. Použitím vzniká nová premenná.

Disjunkcia dvoch premenných:

Pre log. výroky neplatí podmienka vzájomného vylúčenia.

Dôkaz:

$$X \vee Y = \{x \vee y, x \in X \wedge y \in Y\} \, Ked' \, (x_a = 1 \vee y_b = 0) = 1 \wedge (x_c = 0 \vee y_d = 1) = 1$$
$$a \; preto \, ((x_a \vee y_b) \wedge (x_c \vee y_d)) = 0$$

2.2.2 Pravdepodobnosť logických výrokov

Aby sme mohli logický výrok „obaliť“ pravdepodobnosťou musíme najskôr logický výrok experimentálne merať niekoľko krát. Tieto informácie nazveme „predbežná znalosť“(π). Predbežná znalosť v bayesovskom programe bude neskôr rozšírená. Na základe tejto predbežnej znalosti dokážeme logický výrok definovať s určitou pravdepodobnosťou P. Niekedy predbežná znalosť o danom logickom výroku nevychádza z meranie, ale napr. z analytického poznania situácie alebo problému.

Pravdepodobnosť logických výrokov preto vždy bude podmienená predbežnou znalosťou a bude jej priraďovaná hodnota reálneho čísla v rozpätí 0 až 1.

Označenie:

$$P(a|\pi)\in\langle 0,1\rangle$$

Príklad:

Majme logický výrok a = "Predmet je vzdialený 0,5m". Úlohou je zistiť na koľko percent je tento výrok pravdivý pre konkrétne meranie. Predmet si položíme do vzdialenosti 0,5 metra a experimentálne meriame túto vzdialenosť prístrojom Ak chceme vedieť s akou pravdepodobnosťou je tento výrok pravdivý musíme experimentálne odmerať n-krát danú vzdialenosť meracím prístrojom . Hodnoty meracieho prístroja budú predbežnou znalosťou pre výrok „a" a na základe π *vypočítame pravdepodobnosť* $P(a|\pi)$ *= počet správnych meraní / počet všetkých meraní. Pričom chyba merania môže nastať zlým meracím prístrojom.*

Keďže budeme pracovať s viacerými log. výrokmi budeme potrebovať aj tieto pravdepodobnosti:

$$P(a\vee b|\pi), P(a\wedge b|\pi), P(\neg a|\pi) \quad .$$

2.2.3 Podmienenosť logických výrokov

Ako už bolo napísane pravdepodobnosť platnosti logických výrokov závisí od predbežnej znalosti. Teda pravdepodobnosť logického výroku je podmienená predbežnou znalosťou. Táto podmienenosť sa vyskytuje aj medzi logickými výrokmi a preto keď pravdivosť log. výroku a závisí od pravdivosti log. výroku b budeme túto pravdepodobnosť označovať $P(a|b\wedge\pi)$ a keď $P(b|\pi)>0$ tak platí :

$$P(a|b\wedge\pi)=P(a\wedge b|\pi)/P(b|\pi) \qquad [P.1]$$

Na pravdepodobnostné vyvodzovanie budú postačujúce len tieto dva pravidlá:

<u>1.Pravidlo konjunkcie pre výroky:</u>

Na základe [P.1] môžme vyvodiť :

$$\begin{aligned}P(a\wedge b|\pi)&=P(a|\pi)*P(b|a\wedge\pi)\\&=P(b|\pi)*P(a|b\wedge\pi)\end{aligned} \qquad [P.2]$$

<u>2.Pravidlo normalizácie pre výroky:</u>

Pre každý výrok platí:

$$P(a|\pi)+P(\neg a|\pi)=1 \qquad [P.3]$$

Príklad: Ak výrok „a“ má $P(a|\pi)=0{,}6$ *tak z toho vyplýva že* $P(\neg a|\pi)=0{,}4$.

2.2.4 Pravdepodobnosť premenných

Premenná je zložená z konečného počtu log. výrokov(hodnôt) a preto jej pravdepodobnosť budeme označovať:

$$P(X|\pi)=\{\textit{všetky } x_i \in X : P(X=x_i|\pi)=P(x_i|\pi)\} \quad [P.4]$$

Čo znamená, že pravdepodobnosť premennej je zložená z konečného počtu pravdepodobnosti. Pre premenné platia tie isté pravidla ako pre výroky, keďže sú z nich zložene. Tieto pravidla sú vo všeobecnom tvare aby, sme nemuseli každú premennú úplné rozpisovať.

1.Pravidlo konjunkcie pre náhodné premenné

z pravidla [P.2] :

$$\begin{aligned} P(X \wedge Y|\pi) &= P(X|\pi)\times P(Y|X\wedge\pi) \\ &= P(Y|\pi)\times P(X|Y\wedge\pi) \end{aligned} \quad [P.5]$$

2.Pravidlo normalizácie pre náhodné premenné

z pravidla [P.3] :

$$\sum_x P(X|\pi)=1 \quad [P.6]$$

3.Pravidlo marginalizácie pre náhodné premenné

$$\sum_x P(X\wedge Y|\pi) =_{[P.6]} P(1\wedge Y|\pi)=P(Y|\pi) \quad [P.7]$$

Tieto pravidlá sú dostatočné na hocijakú inferenciu v bayesovskom programe. Práve problém nejasnosti okolitého sveta a nepresnosti senzorov v robotike je riešený inferenčnou metódou, ktorá používa **bayesovské pravidlo.** Definícia tohto pravidla je v časti podmienenosť náhodných premenných

Každá náhodná premenná je definovaná rozdelením pravdepodobnosti (distribúcia pravdepodobnosti).

Distribúcia pravdepodobnosti

Rozdelenie pravdepodobnosti môže byť dané :

- tabuľkou pre diskrétne premenné
- grafom
- funkciou hustoty pravdepodobnosti pre spojité premenné

2.2.5 Podmienenosť náhodných premenných

Keď hovoríme o podmienenosti náhodnej premennej A od náhodnej premennej B musíme brať do úvahy či existuje závislosť medzi premennou A od premennej B

<u>Definícia</u>: Náhodná premenná A a náhodná premenná B sú nezávisle práve vtedy, keď platí

$$(P(A|B\wedge\pi)=P(A|\pi))\vee(P(B|A\wedge\pi)=P(B))\vee(P(A\wedge B\wedge\pi)=P(A|\pi)\times P(B|\pi))$$

Pre podmienené pravdepodobnosti platí:

$$P(X^i|Y^1\wedge\pi)\wedge P(X^i|Y^2\wedge\pi)\wedge\ldots\wedge P(X^i|Y^n\wedge\pi)=P(X^i|Y^1\wedge Y^2\wedge\ldots\wedge Y^n\wedge\pi)\ .$$

Ak A nieje závisle od B tak platí $P(A|B\wedge\pi)=P(A\wedge\pi)$ potom túto podmienenosť nazveme *nezávislá podmienenosť*. V opačnom prípade použitím pravidla [P.4] a [P.1] dostaneme

$$P(A|B\wedge\pi)=P(A\wedge B|\pi)/P(B\wedge\pi)\quad [P.8]$$

Takúto podmienenosť nazveme *závislá podmienenosť*.

<u>Bayesovské pravidlo</u> :

z pravidla [P.8] a [P.5] :

$$P(A|B\wedge\pi)=P(A|\pi)\times P(B|A\wedge\pi)/P(B|\pi)$$

Dôležitým aspektom bayesovského pravidla je, že závislosť, ktorú je ťažko vypočítať môžeme týmto pravidlom otočiť a dostaneme sa do obrátenej podmienky, ktorá je už ľahšie vyčísliteľná.

3. Bayesovské programovanie robotov

Na základe hore uvedených definícií a pravidiel je možné vnemy a akcie robota popísať množinou premenných a potom z takejto formulácie je možné počítať pravdepodobnostnú distribúciu na podmnožine premenných zo všetkých dostupných premenných za nejakých predpokladov.

Je ťažké organizovať takto dostupné informácie a popritom sa držať matematickej formulácie. A tu sa dostáva do popredia Bayesovský framework, ktorého cieľom je dodržiavať formalizmus a jednotvárnosť postupu riešenie úlohy. Bayesovské programovanie robotov bolo definované v článku *Bayesian Robot Programming* [1]. Hlavnou častou je Bayesovský program. Tento program je ďalej rozdelený na jednotlivé časti, ktoré vytvárajú jednoznačnú štruktúru Zložky Bayesovského programu:

1. <u>popis bayesovského programu</u>
    1. špecifikácia
    2. identifikácia
2. <u>otázka bayesovského programu</u>
    1. fáza výberu

Popisom bayesovského programu sa rozumie pravdepodobnostný model nášho problému. Je to súhrn informácií, ktoré sú nám známe a definujú daný problém. Teda je to vzájomná distribúcia pravdepodobnosti daných premenných. Pozostáva z dvoch častí : špecifikácia a identifikácia. Špecifikáciou sa vytvára *predbežná znalosť* (π). A identifikáciou sa vytvárajú *experimentálne dáta* (Δ). Výsledný program je potom charakterizovaný dvojicou $<\pi, \Delta>$.

Otázka bayesovského programu je vyvodená z cieľa bayesovského programu a je spojená so vzájomnou distribúciou procesom *inferencia*. Na zodpovedanie otázky slúži fáza výberu.

Predbežná znalosť

Pod pojmom predbežná znalosť v bayesovskom programe sa rozumie súhrn informácií, ktoré definujú správanie robota. Je to jeho báza znalosti. Obsahuje množinu vhodných premenných, dekompozíciu vzájomnej distribúcie a formy, ktoré vzniknú z dekompozície. Keďže v niektorých prípadoch sa bude predbežná znalosť líšiť od nejakého správania budeme musieť π rozdeliť na jednotlivé celky, ktoré budú charakterizovať dané správanie. $\pi=(\pi_{spravanie.1}\wedge\pi_{spravanie.2}\wedge\ldots\wedge\pi_{spravanie.n})$.

Experimentálne dáta

Tieto dáta sú získavané počas identifikačnej fázy. Sú to n-tice nameraných hodnôt z premenných, ktoré ovplyvňujú danú premennú. A z týchto dát potom môžeme definovať distribúcie. Tak ako predbežná znalosť aj experimentálne dáta sa rozlišujú od druhu správania a preto si ich rozdelíme na menšie celky $\delta=(\delta_{spravanie1}\wedge\delta_{spravanie1}\wedge\ldots\wedge\delta_{spravanien})$.

V nasledujúcich statiach popíšem jednotlivé zložky bayesovského programu .

4. Špecifikácia

Cieľom špecifikácie je zadefinovať všetky známe poznatky daného problému v pravdepodobnostných termínoch. Táto časť je definovaná programátorom. Špecifikácia je zložená z troch častí definovanie vhodných premenných, dekompozícia a definovanie parametrických foriem. V tejto časti som aj opísal bayesovskú inferenciua jej inferenčné mechanizmy.

4.1. Vhodné premenné

Na to aby sme zadefinovali čo najvhodnejšie premenné pre daný cieľ platia tieto nasledovné pravidla:

1. Vhodná premenná je taká premenná, ktorá vyplýva z cieľu, je odvodená z dostupných informácií o cieli, závisí od reakcií subjektu,...
2. Zadefinovať také premenné ktoré priamo alebo nepriamo nezávisia od ostatných.
3. Postupne dodávať premenné, ktoré priamo závisia od cieľu.c

Definovať vhodné premenné je veľmi dôležitá činnosť, pretože od toho závisí úspešnosť experimentu.

Pre nášho robota premenné rozdeľujeme na *motorické* a *senzorické. Motorické premenné* budú definované pohybom robota a *senzorické* premenné budú definované hodnotami senzorov.

4.2 Dekompozícia vzájomnej distribúcie

Vzájomná distribúcia sa dá rekurzívnym použitím pravidla konjukcie rozdeliť na :

$$
\begin{aligned}
&P(X^1 \wedge X^2 \wedge \ldots \wedge X^n | \Delta \wedge \pi) = \\
&=_{[P1.4]} P(X^1 | \Delta \wedge \pi) \times P(X^2 \wedge X^3 \wedge \ldots X^n | X^1 \wedge \Delta \wedge \pi) = \\
&=_{[P1.4]} P(X^1 | \Delta \wedge \pi) \times P(X^2 | X^1 \wedge \Delta \wedge \pi) \times P(X^3 \wedge X^4 \wedge \ldots \wedge X^n | X^2 \wedge X^1 \wedge \Delta \wedge \pi) = \\
&=_{[P1.4]} P(X^1 | \Delta \wedge \pi) \times P(X^2 | X^1 \wedge \Delta \wedge \pi) \times P(X^3 | X^2 \wedge X^1 \wedge \Delta \wedge \pi) \times .. \\
&.\times P(X^n | X^{(n-1)} \wedge X^{(n-2)} \wedge \ldots \wedge X^1 \wedge \Delta \wedge \pi)
\end{aligned}
$$

Túto dekompozíciu môžeme nazvať *úplná dekompozícia*. Naším cieľom je aproximovať danú dekompozíciu a na to využijeme závislé podmienenosti premenných. $P(X^i | X^{(i-1)} \wedge \ldots \wedge X^1 \wedge \Delta \wedge \pi)$. Nahradíme $P(X^i | L^i \wedge \Delta \wedge \pi)$. Kde premenná L^i je množina tých premenných od ktorých je premenná X^i podmienene závislá. Teda odoberieme podmienenosti ktoré sú nezávislé. Na to aby sme mohli toto uskutočniť použijeme pravidlo nezávislej podmienenosti. $P(A | B \wedge \Delta \wedge \pi) = P(A | \Delta \wedge \pi)$.

$$P(X^1 \wedge X^2 \wedge \ldots \wedge X^n | \Delta \wedge \pi) =_{dek} P(X^1 | \Delta \wedge \pi) \times P(X^2 | L^2 \wedge \Delta \wedge \pi) \times \ldots \times P(X^n | L^n \wedge \Delta \wedge \pi)$$

Použitím reťazovým pravidlom môžeme prepísať na:

$$P(X^1 \wedge X^2 \wedge \ldots X^n | \Delta \wedge \pi) =_{dek} P(X^1 | \Delta \wedge \pi) \times \Pi_{(i=2)} P(X^i | L^i \wedge \Delta \wedge \pi)$$

4.3. Definovanie foriem

Naším cieľom je teraz každú z distribúcií po dekompozícií správne definovať nejakou formou, ktorá môže byť definovaná funkciou, tabuľkou, histogramom alebo ďalším bayesovským programom. Definovaním funkciou distribúcie pravdepodobnosti sa vo veľkej miere zmenší priestor hľadania. Pretože takáto funkcia je definovaná výrazne menším počtom hodnôt = parametrov.

Funkcia f je definovaná parametrami u, ktoré vyplývajú práve zo závislosti L^i alebo aj Δ :

$$P(X^i | L^i \wedge \Delta \wedge \pi) = f_u(P(X^i | L^i \wedge \Delta \wedge \pi))$$

Takúto formu nazveme *parametrická forma*. Samozrejme samostatná forma môže byť aj bayesovský program, keďže sme stále v rovine premenných a pravdepodobnosti.

4.4. Definovanie otázky

Keďže teraz už na jednej strane vieme vzájomnú distribúciu dekomponovať. Dokážeme na druhej zo vzájomnej distribúcie vytvoriť hocijakú distribúciu. Vhodné premenné $\{X^{1,} ..., X^n\}$ si rozdelíme do troch podmnožín :

- *H – hľadané* : premenné, ktorej hodnoty chceme zistiť
- *S - skryté* : premenné ktorých hodnoty nevieme priamo zistiť.
- *Z - známe* : premenné, ktorých hodnoty poznáme

Otázka je potom v tvare $P(hľadané \wedge skryté | známe \wedge \Delta \wedge \pi)$. Dôvodom prečo sa premenné skryté objavujú na ľavej strane je, pretože vždy predpokladáme, že prostredie obsahuje nejaké skryté premenné, ktorých hodnoty nepoznáme. A preto si zadefinujeme intuitívnu verziu tejto otázky, ktorej definícia je nasledovná :

$$P(hľadané | známe \wedge \Delta \wedge \pi)$$

Premenné skryté sa nám vrátia pri inferencií. Na vypočítanie vzťahu tejto *otázky* a *dekompozície vzájomnej distribúcie* slúži **bayesovská inferencia.**

4.5. Bayesovská inferencia

Bayesovská inferencia sa drží jednotného postupu. Máme definovanú otázku v tvare pravdepodobnostnej distribúcie $P(hľadané \mid známe \wedge \Delta \wedge \pi)$. Naším cieľom je teraz sa dostať k vzájomnej distribúcií všetkých premenných v pravdepodobnostnom modeli, pretože procesom dekompozície sme si vzájomnú distribúciu aproximovali. Na to aby sme to dosiahli využijeme dva pravidlá. Distribúcia $P(hľadané \mid známe \wedge \Delta \wedge \pi)$ ignoruje výskyt premenných *skrytých,* ktoré taktiež ovplyvňujú výsledné pravdepodobnostné rozdelenie distribúcie, pretože meranie nám poskytlo len *známe* premenné z určitými hodnotami. Preto na túto distribúciu použijeme pravidlo marginálizácie [P.7] a dostaneme $\sum_{skryte} P(hľadané \wedge skryté \mid známe \wedge \Delta \wedge \pi)$ .Takto sme v našej distribúcií získali všetky premenné definujúc pravdepodobnostný priestor nášho modelu. Ale ešte sa musíme zbaviť podmienenosti. Na to použijeme pravidlo konjukcie premenných [P.6] a dostaneme:

$$\frac{\sum_{skryte} P(hľadané \wedge skryté \wedge známe \mid \Delta \wedge \pi)}{P(známe \mid \Delta \wedge \pi)}$$

Kde distribúcia $P(známe \mid \Delta \wedge \pi)$ vystupuje ako normalizačná hodnota pre našu vzájomnú distribúciu, čo v konečnom dôsledku nebude pre nás dôležité. Preto je možné označiť ju ako konštanta α a tak dostaneme výslednú distribúciu z otázky bayesovského programu procesom inferencie:

$$P(hľadané \mid známe \wedge \Delta \wedge \pi) =_{inf} \alpha * \sum_{skryte} P(hľadané \wedge skryté \wedge známe \wedge \Delta \wedge \pi)$$

Takáto výsledná distribúcia sa dá prepísať do parametrických foriem, ktoré vznikli dekompozíciou:

$$P(hľadané \mid známe) = \alpha * \sum_{skryte} (f(X^1) \times f(X^2 \mid L^2) \times \ldots \times f(X^n \mid L^n))$$

Pri inferencií je teraz dôležité túto formuláciu otázky čo najviac zjednodušiť. Problémom sú naše premenné skryté. Pretože na nájdenie odpovede musíme prehľadávať všetkými hodnotami premennej skryté všetky parametrické formy. Čo je pri realizácií, keď máme veľký počet skrytých premenných, veľmi pamäťovo a časovo náročné a preto v reálnom dynamickom prostredí nerealizovateľné. Preto si musíme pomôcť rôznymi inferenčnými metódami.

4.6. Inferenčné metódy v bayesovskom programe

- počas dekompozície
  - nezávislá podmienenosť ma za následok zredukovanie veľkosti distribúcií, ktoré vznikli dekompozíciou
- samostatne definovanie distribúcie formami nám zredukuje prehľadávací priestor
- počas inferencie:
  - definovanie uniformných distribúcií pri definovaní foriem ma za následok ,že jej hodnota je konštantná
  - distribúcie, ktoré obsahujú len premenné *známe* sú v tomto prípade konštantné
  - distribúcie, ktoré obsahujú len premenné *známe* alebo premenné *hľadané* tak tieto distribúcie môžeme vybrať pred sumu premenných *skrytých,* pretože premenné *skryté* žiadnym spôsobom neovplyvnia túto distribúciu
- definovaním správnej otázky pre daný problém nás môže viesť k rýchlemu výsledku

5. Identifikácia

Je to proces učenia sa nejakého správanie pomocou učiteľa. Učiteľom je osoba, ktorá riadi robota(joystickom, klávesnicou). Cieľom identifikácie je zaplniť všetky voľné parametre ktoré vznikli pri definovaní parametrických foriem. Počas identifikácie sa zbierajú n-tice dát všetkých premenných *známe* z otázky. Tieto n-tice sa ukladajú do premennej experimentálne dáta Δ .
Z ktorých sa následne vypočítavajú voľné parametre. Keďže počet všetkých kombinácií hodnôt známych premenných môže byť veľmi veľký, čo ma za následok, že niektoré hodnoty voľných parametrov nebudú vôbec definované. Preto je dôležité použiť vhodný učiaci algoritmus. Aby sme tak eliminovali nejasnosť našej bázy vedomosti.

Pri menšom počte premenných známe a pri vhodnom experimente je možné tieto udalosti zadefinovať parametrickými formami vopred určenými hodnotami.

Samozrejme ďalším dôležitým aspektom pri identifikácií je samostatné riadenie robota učiteľom. Pri zlom riadení dostaneme zlé výsledky.

6. Fáza výberu

Fáza identifikácie skončila s konkrétnymi distribúciami pre všetky hodnoty premenných *hľadané* a *skryté,* ktoré sú definované svojimi parametrami. V tejto fáze sa má robot autonómne rozhodovať na základe kladenej otázky. Otázka je v tvare $P(hľadané \mid známe \wedge \delta \wedge \pi)$. δ sú všetky namerané hodnoty z Δ . Teda robot z nameraných hodnôt premennej *známe* má definovanú distribúciu premennej *hľadané.* A z tejto distribúcie vyberieme hodnotu, ktorú potom pošleme robotovi a ten ju vykoná.

Výber konkrétnej hodnoty z distribúcie je možné vykonať rôznymi spôsobmi:

1. výber najlepšej hodnoty z distribúcie
2. stochastický výber vzhľadom na rozdelenie distribúcie
3. utility funkciou

V našom prípade budeme používať hlavne stochastický výber, pretože takto najlepšie zoberieme do úvahy nepresnosť vonkajšieho sveta a zároveň predídeme zablokovaniu robota v singulárnych bodoch, ktoré by mohli nastať pri deterministickom výbere. Aby sme mohli nášho robota použiť v bayesovskom programe, výber je vykonávaný každú desatinu sekundy. Typickým príkladom otázky je výber hodnôt motorických premenných (*hľadané)* z nameraných hodnôt senzorických premenných (*známe)*. Následne sa vykonáva cyklus pre autonómne riadenie robota, ktorý sa drží tohto postupu:

1. Získame hodnoty senzorových premenných
2. Stochastický vyberieme hodnotu z distribúcie motorickej premennej.
3. Pošleme vybranú hodnotu robotu a ten ju vykoná.

7. Rôzne správania

Niekoľko rôznych správaní je možne získať zmenou nejakých komponentov v bayesovskom programe:

- je možné pozmeniť otázku, pričom sa *popis bayesovského programu* nezmení. To znamená, že výsledná otázka bude len ochudobnená o niekoľko premenných, ktoré z nejakého dôvodu(výpadok senzora) nie je možne merať.

- je možne zmeniť iba *experimentálne dáta*. Týmto sa dokáže robot naučiť nové správania bez zásahu do programu. Čo znamená, že zmeníme spôsob získavania dát počas identifikačnej fázy. Ako príklad môžeme použiť bayesovský program
  so správaním tlačenie objektu. Zmenou experimentálnych dát jednoducho naučíme robota objekt obchádzať.

- zmena *popisu bayesovského modelu*. Týmto získame nový model s novou predbežnou znalosťou ako aj s novými experimentálnymi dátami, teda aj nové správanie.
- kombináciou niekoľkých správaní vznikne nové správanie .

8. Kombinácia dvoch správaní v jednom modeli

Ako som už spomínal bayesovský model môže v sebe obsahovať aj viacero správaní . Kombináciou dvoch správaní (spravanie1,spravanie2) vznikne nové správanie(správanie3). Naša *predbežná znalosť* a *experimentálne dáta* budú obsahovať po dvoch zložkách z každého správania :

$$\pi_{spravanie3} = \{\pi_{spravanie1}, \pi_{spravanie2}\}$$

$$\Delta_{spravanie3} = \{\Delta_{spravanie1}, \Delta_{spravanie2}\}$$

Na rozlíšenie týchto dvoch správaní bude slúžiť nová *skrytá* premenná. Táto premenná tiež závisí od nejakých ďalších premenných(väčšinou motorických) a to znamená, že ich distribúciu po dekompozícií vhodne zadefinujeme parametrickou formou. Tá bude signalizovať s akou pravdepodobnosťou sa má vykonať dané správanie. Teraz si rozoberiem kombináciu dvoch správaní.

Máme premenné $X^{1,} X^{2,} .., X^n$,ktoré vznikli konjunkciou vhodných premenných dvoch správaní. Ďalej sa tu vyskytuje nová premenná S .Zavedieme si ešte dve premenné U_1 , ktorá obsahuje konjunkciu všetkých premenných 1. správania a premenná U_2 , ktorá obsahuje konjunkciu všetkých premenných 2. správania. Premenná S nadobúda dve hodnoty $S = \{správanie_{1,} správanie_2\}$. Táto premenná rozhoduje, ktoré správanie sa bude vykonávať. Dekompozícia je potom nasledovná:

$$P(X^1 \wedge X^2 \wedge ... \wedge X^n \wedge S | \pi_{spravanie3} \wedge \Delta) =$$

$$P(X^1 | L(X^1) \wedge \Delta \wedge \pi) \times ... \times P(X^n | L(X^n) \wedge \Delta \wedge \pi) \times P(S | L(S) \wedge \Delta \wedge \pi)$$

Od premennej *S* závisí či sa robot bude riadiť prvým alebo druhým správaním.

Keď S = $správanie_1$ potom $P(U_1 | \Delta_{spravanie1} \wedge \pi_{spravanie1})$

S = $správanie_2$ potom $P(U_2 | \Delta_{spravanie2} \wedge \pi_{spravanie2})$

Pri definovaní parametrických foriem, ktoré sú konfliktné v tom zmysle že ich hľadané premenné sú totožné pre obidve správanie rozlišujeme ich práve na základe premennej S:

$P(X^i | L(X^i) \wedge \pi_{spravanie3} \wedge \Delta_{spravanie3})$ =

pre správanie1: $P(X^i | L(X^i) \wedge \pi_{spravanie1} \wedge \Delta_{spravanie1})$

pre správanie2: $P(X^i | L(X^i) \wedge \pi_{spravanie2} \wedge \Delta_{spravanie2})$

Kde množina $L(X^i)$ pre správanie1 alebo správanie2 použitím pravidla pre nezávislé podmienenosti neobsahuje premenné z pôvodnej množiny $L(X^i)$ pre správanie3.

Parametrická forma $P(S | L(S))$ bude rozdielna pre obidve hodnoty tejto premennej. Keďže máme len dve hodnoty premennej jedna bude komplementom k druhej na základe pravidla P[1.6]:

$$P(S = spravanie1 | L(S = spravanie_1)) = f(L(S = spravanie_1))$$

$$P(S = spravanie_2 | L(S = spravanie_2)) = 1 - P(S = spravanie_1 | L(S = spravanie_1))$$

Otázka bude definovaná nejakými motorickými premennými a premennou skrytou a tie budú závisieť od senzorických premenných

$$P(hľadané_3 \wedge S | známe_3 \wedge \Delta_{spravanie3} \wedge \pi_{spravanie3}) =$$

$$\Sigma_S P(hľadane_3 \wedge S | známe_3 \wedge \Delta_{spravanie3} \wedge \pi_{spravanie3}) =$$

$$\begin{aligned}&\alpha * [P(S = spravanie_1 | L(S = spravanie1)) \times P(X^1 | L(X^1) \wedge \Delta_{spravanie1} \wedge \pi_{spravanie1}) \times \ldots \\ &\times P(X^n | L(X^n) \wedge \Delta_{spravanie1} \wedge \pi_{spravanie1}) + \\ &P(S = spravanie_2 | L(S = spravanie2)) \times P(X^1 | L(X^1) \wedge \Delta_{spravanie2} \wedge \pi_{spravanie2}) \times \ldots \\ &\times P(X^n | L(X^n) \wedge \Delta_{spravanie2} \wedge \pi_{spravanie2})]\end{aligned}$$

Výsledok po inferencií bude obsahovať práve dekompozíciu s parametrickými formami pre jedno alebo druhe správanie. Výsledná distribúcia pre otázku bude preto zosumovaná týmito dvomi dekompozíciami. A bude závisieť od parametrických foriem s premennou S, ktoré budú do výsledku zasahovať ako „váha“. Teda keď $P(S = spravnie_1 | Rodič(S = spravanie_1)) = 0$ tak výsledná distribúcia bude obsahovať len $správanie_2$ a naopak. Keď ani jedno zo správaní nebude nulové výsledná dekompozícia bude spočítaním týchto dvoch distribúcií s určitou váhou. Takéto kombinovanie správanie môžeme nazvať ako „pravdepodobnostné „if then else“.

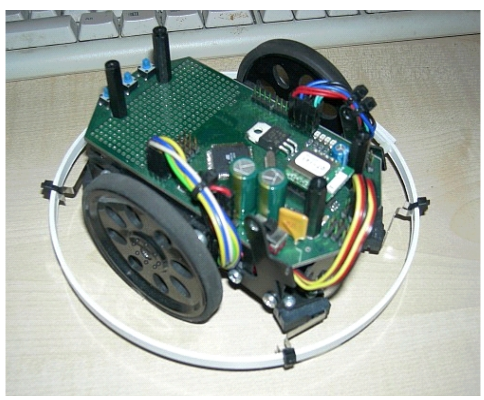
9. Experiment

Cieľom experimentu bolo naučiť robota tlačiť nejaký predmet. Predmet nemohol byť pripevnený k ploche a musel mať primeranú váhu na to aby ho robot mohol tlačiť. Tento experiment bol vykonaný aj v práci [1] na robotovi Khepera.

Robot Khepera bol schopný naučiť sa dané správanie do 20-30sekúnd. Mal osadených 6 senzorov vpredu, ktorými zisťoval aktuálnu polohu k najbližšiemu predmetu. Ďalším experimentom Khepery bolo naučiť robota obchádzať predmet. Postačilo len zmeniť identifikačné dáta a robot bol schopný prekážky obchádzať. Keďže ide o dva odlišné roboty porovnanie je na mieste. Experimenty som vykonával v Spoločnom robotickom laboratóriu FMFI UK a FEI STU v Bratislave. Robot, ktorý mi bol poskytnutý na vykonanie experimentov bol Sbot v 2.0. S priloženým CD, ktorý obsahoval softvér Avrdude , AVRStudio a SbotManager.

<u>Robot Sbot 2.0</u>

Je to mobilný robot na, ktorom je osadených 5 infračervených senzorov na zisťovanie vzdialenosti Je poháňaný dvomi servomotormi. Pre účel tohto experimentu nám postačí zatiaľ táto definícia. Bližšia špecifikácia je v odstavci špecifikácia Sbot.



Robot Sbot 2.0

<u>Robotické laboratórium</u>

Robot sa pohybuje na ploche veľkosti 1x1metrov. Na ploche sa nachádza malá kartónová krabica, ktorá slúžila na experiment.

Sbot versus Khepera

Robot Khepera bol riadený joystickom, takže pohyb je plynulejší. Je osadený viacerými senzormi, takže výpočet premenných na určenie polohy predmetu je presnejší.

Robot Sbot je obmedzený svojou malou RAM pamäťou.

9.1 Bayesovský program

V tejto časti je popísaná podrobná realizácia bayesovského programu pre robot Sbot. Keďže úlohou robota bolo to isté čo Khepera. Obsah bayesovského programu je takmer totožný. Rozdiely boli samozrejme kvôli rôznym proporciám robotov a rozličnosti počtu senzorov.

špecifikácia

1.vhodné premenné:

Na splnenie nášho cieľa som si zadefinoval tri premenné, ktoré by mali dostatočne pokryť našu problematiku.

Vzd - bude reprezentovaná hodnotami od 0 do 7 čo symbolizujú vzdialenosť od najbližšieho predmetu k robotovi (7 najbližšie,0 najďalej).

Rot – bude reprezentovaná hodnotami od 0 po 17 a symbolizujú polohu najbližšieho predmetu k robotovi. (0 úplne naľavo,16 úplne napravo)

Uhl – bude reprezentovaná hodnotami od -3 po +3 a symbolizujú ako rýchlo sa má robot zatočiť.

Hodnota premennej Vzd sa získavala ako index senzora s maximálnou hodnota zo všetkých senzorov. Hodnota premennej Rot sa získavala od získavané od hodnôt senzora. A premenná Uhl bude nadobúdať hodnoty od hodnôt klávesnice, ktorá ovláda robota. Teraz si popíšeme vzájomne závislosti medzi premennými.

Premenné Vzd a Rot sú závisle len od predbežnej znalosti. Premenná Uhl bude závisieť od týchto dvoch, pretože práve premenná Uhl dáva do pohybu robota v závislosti od vzdialenosti a polohy od prekážky. Závislé podmienenosti premenných vyzerajú takto:

$$Vzd \mid \Delta \wedge \pi \quad , \quad Rot \mid \Delta \wedge \pi \quad , \quad Uhl \mid Vzd \wedge Rot \wedge \Delta \wedge \pi$$

2.dekompozícia:

Pravdepodobnostná distribúcia vhodných premenných bayesovského programu $P(Vzd \wedge Rot \wedge Uhl | \Delta \wedge \pi)$ po dekompozícií (použitím dvakrát pravidla konjukcie) je vzájomná distribúcia v tvare:

$$P(Vzd | \Delta \wedge \pi) \times P(Rot | \Delta \wedge \pi) \times P(Uhl | Vzd \wedge Rot \wedge \Delta \wedge \pi)$$

<u>3.parametrické formy:</u>

$P(Vzd | \Delta \wedge \pi)$ a $P(Rot | \Delta \wedge \pi)$ si zadefinujeme uniformou distribúciou, keďže nemáme žiadne apriori informácie o vzdialenosti a polohy predmetov.(pre každú hodnotu je rovnaká pravdepodobnosť)

Cieľom je aby robot tlačil objekt čo znamená že pre danú udalosť robota v prostredí existuje práve jedna hodnota jeho uhlovej rýchlosti. Teda forma $P(Uhl | Vzd \wedge Rot \wedge \Delta \wedge \pi)$ smeruje vždy k jednému bodu. Čo je najlepšie definované Normálnym rozdelením distribúcie. Takéto rozdelenie je definované dvomi hodnotami strednou hodnotou μ a smerodajnou odchýlkou δ Čo znamená, že pre každú dvojicu hodnôt Vzd a Rot existuje práve jedno Normálne rozdelenie pravdepodobnosti $P(Uhl | Vzd = vzd \wedge Rot = rot \wedge \Delta \wedge \pi)$. Definícia normálneho rozdelenia:

$$f(x_i) = (1/(\delta \sqrt{(2\pi)})\, e^{-((x_i - \mu)/(2\delta))})$$

4.otázka:

Z nášho cieľu vyplýva, že chceme vedieť ako sa má robot zatáčať. Teda potrebujeme vedieť aká je distribúcia Uhl pre konkrétne hodnoty premenných Vzd a Rot A tak naša premenná *hľadané* budú Uhl a premenná *známe* bude obsahovať Vzd a Rot .V tomto experimente sa nevyskytujú *skryté* premenné. Preto naša premenná bude definovaná distribúciou

$$P(Uhl|Vzd \wedge Rot \wedge \Delta \wedge \pi)$$

5.bayesovská inferencia:

Otázka je $P(Uhl|Vzd \wedge Rot \wedge \Delta \wedge \pi)$ použitím pravidiel [P.7] a [P.6] dostaneme:

$$P(Uhl|Vzd \wedge Rot \wedge \Delta \wedge \pi) = \frac{P(Uhl \wedge Vzd \wedge Rot \wedge \Delta \wedge \pi)}{P(Vzd \wedge Rot|\Delta \wedge \pi)} = \alpha * P(Uhl \wedge Vzd \wedge Rot \wedge \Delta \wedge \pi) =$$
$$\alpha[P(Vzd| \wedge \Delta\, \pi) \wedge P(Rot|\Delta\, \pi) \wedge P(Rot|Vzd \wedge Pol \wedge \Delta \wedge \pi)]$$

Keďže $P(Vzd|\Delta \wedge \pi)$ a $P(Rot|\Delta \wedge \pi)$ sú definované uniformnou parametrickou formou .Ich hodnoty sú vo všetkých prípadoch rovnaké tak ich môžeme vybrať zo zátvorky ako nejaké konštanty a dostaneme:

$$P(Uhl|Vzd \wedge Rot \wedge \Delta \wedge \pi) = \alpha * P(Uhl|Rot \wedge Vzd \wedge \Delta \wedge \pi)$$

Identifikácia

Máme distribúciu premennej Uhl , ktorá závisí od Vzd a Rot .A táto distribúcia je definovaná Normálnym rozdelením. Implementácia v identifikácií je nasledovná. Máme $[Vzd]*[Rot]$ dvojíc $(Vzd = vzd\ , Rot = rot)$ a pre tieto dvojice existuje práve jedna hodnota μ a δ . Robot pred prvým spustením identifikácie zaplní hodnoty veličín μ a δ hodnotami 0 a 10 . Keď táto situácia nastane tak sa tieto hodnoty vymažú a robot prijme hodnoty ktoré nameral. Počas identifikácie vypočíta hodnoty premenných $Vzd = vzd\ a\ Rot = rot$ a Z výstupu klávesnice mal uloženú poslednú hodnotu premennej Uhl.

Následne pre dvojicu (vzd , rot) vypočítal novú strednú hodnotu a odchýlku pre danú udalosť. Toto sa opakovalo pokiaľ robot nedostal na vstup na ukončenie.

Ukážka pseudo kódu : identifikačnej fázy:

```
VypocitajPremenne{
        vzd = VypocitajVzd(senzory[6]);
        rot = VypocitajRot(senzory[6]);
}

PocitajDistribuciuRot{
        pole[vzd][rot].µ = Vypocitajµ(uhl);
        pole[vzd][rot]. δ = Vypocitaj δ (uhl);
}

while(identifikácia je pravda){
        uhl = hodnota_klavesnice;
        VypocitajPremenne;
        PocitajDistribuciuRot;
}
```

<u>Fáza výberu</u>

Robot vypočíta hodnoty premenných $Vzd = vzd \; a \; Rot = rot$ a z dvojice (vzd , rot) z hodnôt μ a δ stochastickým výberom z normálneho rozdelenia vyberie hodnotu premennej $Uhl = uhl$. Táto hodnota sa pošle funkcii, ktorá na základe hodnoty $Uhl = uhl$ robotovi zadá uhlovú rýchlosť a robot ju vykoná.

Ukážka pseudo kódu:

```
RestituteRobot{
                uhl = VyberZGauss(pole[vzd][rot].µ,pole[vzd][rot].µ);
                PosliRobotovi(uhl);
        }
```

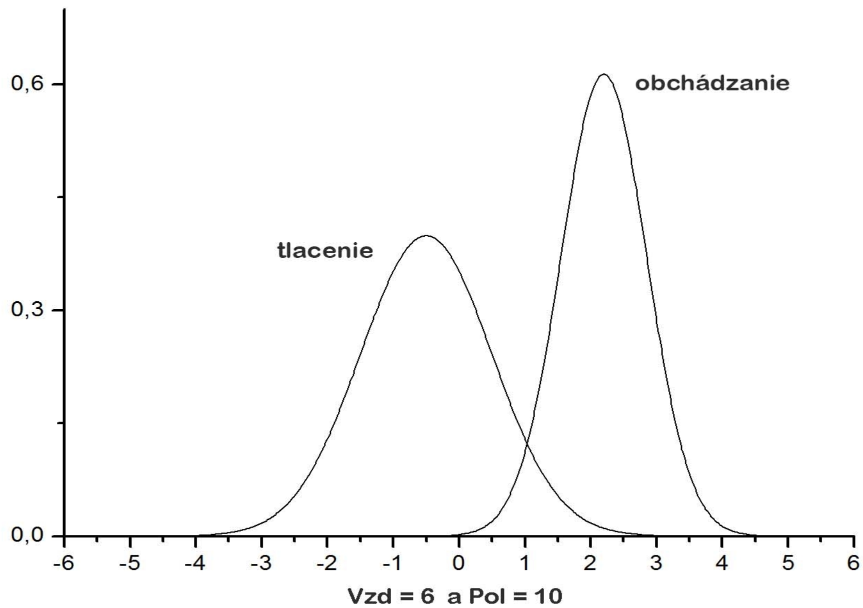
```
while(Fáza výberu je pravda){
        VypocitajPremenne;
        RestituteRobot;
}
```

9.2. Výsledky

Cieľom experimentu bolo naučiť správanie robota pomocou bayesovského programu. Keďže robot Sbot je obmedzený svojou pamäťou. Učenie vyvodzovalo pravdepodobnostné distribúcie len na základe svojich nameraných dát pre konkrétne hodnoty. Pre situácie, ktoré nenastali sa zadefinovalo normálne rozdelenie podobné uniformnému. V takom prípade bola stredná hodnota 0 a smerodajná odchýlka 10. Použitý bayesovský program zbieral dáta po nejakú dobu a potom sa spustila fáza výberu.

Prvou úlohou bolo naučiť robota tlačiť malú kartónovú krabicu. Výsledok učenia tlačenia je v prílohách vo videu č.1. Keďže ako som napísal v časti rôzne správania. Bayesovský program sa dá aplikovať pre rôzne správania a to tým že zmeníme jeho experimentálne dáta. Samozrejme takéto rozšírenie je obmedzené akú predbežnú znalosť máme definovanú. Druhým experimentom s rovnakou predbežnou znalosťou bolo obchádzanie predmetu. Predmetom bola taktiež kartónová krabica. Fáza učenia prebiehala tak, že som riadil robota okolo tejto krabice. Dĺžka toho učenia bola niekedy dlhšia ako bolo ukázané v práci [1]. Ale robot dokázal obchádzať predmet aj keď jeho experimentálne dáta neboli postačujúce. Výsledok učenia obchádzania je v prílohách vo videu č.2

.Obrázok 1 znázorňuje s akou pravdepodobnosťou sa rozhodoval pri obchádzaní a s akou pravdepodobnosťou pri tlačení pri rovnakej udalosti.



Obrázok č.1

Pri tlačení alebo obchádzaní prekážky vo fáze výberu sa robot niekedy dostal do situácie ktorú nepoznal a bolo vidno že ide zle v takom prípade bolo potrebné robota prepnúť do identifikačnej fázy a dané správanie ho naučiť. Výsledky experimentov môžem považovať za úspešné, pretože dokázali splniť zadaný cieľ.

Špecifikácia robota Sbot 2.0

Je jednoduchý robotický systém. Celkový možný počet pripojiteľných analógových senzorov je 8. Pre náš prípad je použitých 5 infračervených senzorov na meranie vzdialenosti.

Pohybuje sa pomocou 2 servomotorov,ktoré dávajú do pohybu kolesá. Ide teda o robota s pohybom typu „diffrential drive". Na hlavnej riadiacej doske je procesor Atmel ATMega128. Tento procesor obsahuje 128kb flash pamäť, do ktorej sa ukladá náš program a 4096B SRam pamäť. Robota môžeme nastaviť do dvoch módov: 1. bežiací mód 2. bootloader mód .Na zapnutie jedného z týchto dvoch módov slúži prepojka, ktorá sa osadzuje jumperom(nenasadený = bežiací mód , nasadený = bootloader mód,).

1.bežiací mód

Robot spustí nahratý program. Robota môžeme ovládať cez virtuálny sériový port, ktorý je vytvorený nad rádiovým spojením BlueTooth, buď pomocou nejakého terminálového programu(napr. Putty).

2.bootloader mód

Nahrá sa požadovaný program do pamäte robota. Program je napísaný v jazyku C. AVRStudio tento program skompiluje a vytvorí súbor v hex formáte. Na zavádzanie programu z počítača do pamäte robota sa používa bootloader. Prenos skompilovaného programu v hex formáte zabezpečuje softvér Avrdude.

10. Záver

Moja práca rozoberá postup tvorby Bayesovského programu. V úvode som sa oboznámil so základnými definíciami logiky a pravdepodobnosti. A následne som definoval a rozoberal všetky zložky bayesovského programu.

V mojom experimente bol ukázaný jednoduchý experiment, ktorý pracoval s malým počtom premenných. Kde otázka bola inferenciou a rôznymi aproximáciami zredukovaná na jednú distribúciu z ktorej sa vyvodzovala výsledná uhlová rýchlosť robota. Keďže táto úloha bola elementárna výsledky boli postačujúce. Robot bol schopný obchádzať prekážku alebo ju tlačiť. Problém zašumenosti senzorov nebol vzatý do úvahy a problém nejasnosti okolitého prostredia, ktorá sa redukuje použitím vhodným učiacim algoritmom bol vyriešený len pravdepodobnostným prístupom k nameraným hodnotám. Hodnoty ktoré nenastali boli implicitne nastavené. V prípade že máme komplexnejší model je dôležité aby robot svoje akcie vykonával s čo najväčšou váhou týchto dvoch problémov.

11. Použitá literatúra

[1] Lebeltel a spol. :: Bayesian Robot Programming. 2004,Kluwer Academic Publishers

Lebeltel, Diard.:: Bayesian Learning Experiments with Robot Khepera.

Lebeltel a spol.:: Bayesian Framework for Robotic programming

Bessier a spol.:: Bayesian Programming.