TSEA56 - Kandidatprojekt i elektronik

Förstudie

Regleruppgift 2

Version 0.1

Adnan Berberovic, adnbe196, 931229-5570
Robert Oprea, robop806, 930508-9618

Beställare: Kent Palmkvist, kentp@isy.liu.se
Handledare ISY: Johan Löfberg, johanl@isy.liu.se
Handledare TEMA: Birgitte Saxstrup, birgitte.saxtrup@liu.se

5 mars 2015

Status

Granskad	Nikolaj Agafonov	2015-03-05
Godkänd		

PROJEKTIDENTITET

2015/VT, Undsättningsrobot Gr. 2
Linköpings tekniska högskola, ISY

Namn	Ansvar	Telefon	E-post
Nikolaj Agafonov	Dokumentansvarig (DA)	072-276 99 46	nikag669@student.liu.se
Adnan Berberovic	Projektledare (PL)	070-491 96 07	adnbe196@student.liu.se
Andreas Brorsson	Testansvarig (TA)	073-524 44 60	andbr981@student.liu.se
Fredrik Fridborn	Designansvarig Sensormodul (DSE)	073-585 52 01	frefr166@student.liu.se
Robert Oprea	Designansvarig Styrmodul (DST)	070-022 10 18	robop806@student.liu.se
Måns Skytt	Designansvarig Kommunikationsenhet (DK)	070-354 28 84	mansk700@student.liu.se

E-postlista för hela gruppen: adnbe196@student.liu.se

Kund: Kent Palmkvist, 581 83 Linköping, Kundtelefon: 013-28 13 47, kentp@isy.liu.se

Kursansvarig: Tomas Svensson, 013-28 13 68, tomass@isy.liu.se
Handledare: Olov Andersson, 013-28 26 58, Olov.Andersson@liu.se

Sammanfattning

Denna förstudie behandlar frågor som rör kartläggning, lokalisering och optimeringsproblem för kortaste väg och kopplas till kandidatprojektet i elektronik där svaren på problemen kommer att vara av ytterst relevans.

Kartläggning och lokalisering kommer att beskrivas av en metod kallad **SLAM**, *(simultaneous localisation and mapping)*. Problemet löses med stor valfrihet beroende av tillgängliga sensorer, beräkningskraft och miljön i fråga.

Optimeringsproblemet är av typen billigaste väg-problem och kommer att behandlas med *dijkstras algoritm*. Denna algoritm går ut på att i befintlig position hitta den billigaste vägen till den position som är av intresse.

Innehåll

Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2015-03-04	Första utkastet	ABe & RO	Nikolaj Agafonov

1 Inledning

Denna förstudie ämnar ge fördjupad kunskap inom några av de delar som ska implementeras i den robot som ska framtas i kandidatprojektet i elektronik. Roboten i fråga ska effektivt kunna avsöka och kartlägga en okänd omgivning, hitta ett bestämt mål, och sedan via den kortaste vägen navigera tillbaka till ursprungspositionen. Detta ställer många krav på robotens funktionalitet, och den ska klara av många olika moment. För att projektet ska kunna genomföras på bästa sätt utförs tre olika förstudier som ämnar fördjupa projektgruppens kunskaper inom olika områden, så att de har en god kunskapsgrund att stå på när väl roboten ska byggas.

2 Problemformulering

De problem som behandlas i denna studie är följande:

Problem 1: Avsökning och kartläggning av en omgivning

Problem 2: Beräkning av optimal väg

I samband med att dessa problem studeras kommer vissa metoder som löser dessa att beskrivas något ingående, tillräckligt för att kunna dra nytta av detta till kandidatprojektet i fråga.

3 Kunskapsbas

Som källor har boken *Optimeringslära* [1], samt artikeln *Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters* [2] använts. Boken kommer från studentlitteratur, och artikeln fanns på bibliotekets databas.

4 Avsökning och kartläggning av en omgivning

Problemet som betraktas här är att en autonom robot ska utforska och avsöka en okänd omgivning, samtidigt som den håller reda på sin egen position. Detta problem benämns allmänt som *simultaneous localisation and mapping* (SLAM). Allmänt kan SLAM-problemet delas upp i mindre delproblem såsom identifierandet av landmärken, dataassociation, uppskattning och uppdatering av position samt uppdatering av landmärken. Dessa delproblem kan lösas med stor valfrihet beroende på tillgängliga sensorer, beräkningskraft och speciellt den miljön man ämnar avsöka. Framöver kommer fokus läggas på avsökning i en inomhusmiljö bestående av ett rutnät. [2]

4.1 Problemformulering

Generellt kan SLAM-problemet beskrivas på följande sätt:

$$p(m_t, x_t | o_{1:t}) \tag{1}$$

där målet är att beräkna positionen x_t samt kartan m_t givet sensorobservationerna o_t under diskreta tidssteg t. Med hjälp av Bayes sats fås följande:

$$\begin{cases} p(x_t | o_{1:t}, m_t) \\ p(m_t | x_t, o_{1:t}). \end{cases} \tag{2}$$

I (2) uppstår en algebraisk loop där robotens position samt kartan beror av varandra. För att lösa detta problem används olika statistiska metoder, t.ex. kalmanfilter eller partikelfilter. Metoden som här kommer beskrivas använder sig av *Rao-Blackwellized partikelfilter* (RBPF). Den utgår från följande problem:

$$p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) \tag{3}$$

där m är kartan, $x_{1:t} = x_1, ..., x_t$ är robotens rörelsebana, $z_{1:t} = z_1, ..., z_t$ är sensorobservationer och $u_{1:t-1} = u_1, ..., u_{t-1}$ är distansmätningarna utförda av roboten.

4.2 Kartläggningsalgoritm

För att beräkna kartan skrivs (3) om som

$$p(m | x_{1:t}, z_{1:t}) \cdot p(x_{1:t} | z_{1:t}, u_{1:t-1}) \tag{4}$$

vilket tillåter att robotens bana beräknas först, och sedan beräknas kartan med hjälp av denna.

4.2.1 Beräkning av robotens position

För att beräkna robotens position kan ett partikelfilter appliceras. Varje partikel representerar en möjlig bana, och en karta associeras till varje partikel. Varje karta är framtagen utifrån mätdata samt den associerade banan. Den partikelfilteralgoritm som används är Rao-Blackwellized *sampling importance resampling* (SIR). Denna algoritm kan sammanfattas i följande steg:

1. **Sampling:** Nästa mängd partiklar $\{x_t^{(i)}\}$ fås genom sampling ur den föreslagna distributionen (eng. proposal distribution) π.

2. **Viktning:** Varje partikel tilldelas en vikt $w_t^{(i)}$ enligt

$$w_t^{(i)} = \frac{p(x_{1:t}^{(i)}|z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(i)}|z_{1:t}, u_{1:t-1})}. \tag{5}$$

3. **Omsampling:** Nya partiklar väljs ut med sannolikhet proportionell mot deras vikt. Efter omsampling har alla partiklar samma vikt.

4. **Kartskattning:** För varje partikel beräknas en skattning av motsvarande karta $p(m^{(i)}|x_{1:t}^{(i)}, z_{1:t})$ givet banan $x_{1:t}^{(i)}$ och tidigare sensorobservationer $z_{1:t}$.

5 Beräkning av optimal väg

Billigaste väg-problem är ett av de mest grundläggande problem som förekommer inom nätverksoptimering. Problemet som analyseras är att hitta den kortaste vägen från en punkt till en annan i ett känt eller delvis känt område. [1]

5.1 Definitioner

För att lösa och förstå detta problem behövs några definitioner.

- En *nod* är en tillståndspunkt. Betecknas n_i för något i.
- En *båge* är en väg mellan två noder. Dess kostnad betecknas med c_{ij}, dvs kostnaden från n_i till n_j.
- En *riktad* båge är en båge med bestämd riktning.
- En *cykel* är en uppsättning bågar som ger upphov till att en nod kan besökas flera gånger.
- *Nodpris* är kostnaden att ta sig till noden i fråga. Betecknas y_i.
- *Föregångare* är noden som besöktes precis innan nuvarande nod. Betecknas p_i.

5.2 Egenskaper hos billigaste väg-problem

I nätverksoptimering där en lösning av billigaste väg-problemet är önskat krävs att vissa förutsättningar uppfylls på nätverket. [1]

- Endast riktade bågar förekommer.
- Nod n_t är uppnåelig från nod n_s.
- Inga cykler med *negativ kostnad* förekommer.

Om en oriktad båge skulle förekomma ersätts den med två riktade bågar mellan noderna med samma kostnad.
Om nod n_t inte är uppnåelig från nod n_s saknar problemet lösning.
Om det finns cykler med en negativ kostnad har problemet en obegränsad lösning. [1]
Detta är i enlighet med det optimeringsproblem som uppstår i kandidatprojektet.

5.3 Bellmans ekvationer

Ett billigaste väg-problem kan formuleras med hjälp utav *Bellmans ekvationer*. När vi har ett känt antal, n, noder i nätverket kan ekvationerna skrivas:

$$\begin{cases} y_s = 0 \\ y_j = \min\limits_{i:(i,j)\in B} y_i + c_{ij}, \quad j = 2, .., n \end{cases} \tag{6}$$

där B är mängden av alla bågar i nätverket, c_{ij} är kostnaden på båge (i, j) och y_j är den billigaste vägen från startnoden n_s till nod n_j. [1]
Alla algoritmer som löser ett billigaste väg-problem försöker att systematiskt uppfylla Bellmans ekvationer.

5.4 Dijkstras algoritm

Dijkstras algoritm är en algoritm som löser just det beskrivna problemet. Denna algoritm bestämmer alltså en billigaste väg från en nod, n_s till en annan nod n_t, i ett nätverk med nodmängden N och bågmängden B.
Algoritmen kan sammanfattas enligt följande: [1]

Steg 0 Dela upp nodmängden i två mängder, Avsökta noder, $A = \emptyset$, och ej avsökta noder, $D = N$. Initiera startnoden med pris $y_s = 0$ och alla övriga noder med initialt nodpris $y_j = \infty$
Märk startnoden med $(p_s, y_s) = (-, 0)$ och övriga noder med $(-, \infty)$.

Steg 1 Sök efter den billigaste noden $n_i \in D$.

Steg 2 Avsök nod n_i, det vill säga undersök alla bågar $(i, j) \in B$ utgående från nod n_i. Om totalkostnaden fram till nuvarande nod n_i och bågkostnaden till nod n_j tillsammans blir mindre än nodpriset y_j så finns en billigare väg från n_s till n_j som går igenom n_i hittats. Märk n_j med nya värden, $(p_j, y_j) = (i, y_i + c_{ij})$.

Steg 3 Flytta över den avsöka noden n_i från mängden D till A.

Steg 4 Om alla noder har avsökts har en billigaste väg funnits. Annars gå till Steg 1 och repetera.

5.4.1 Komplexitet

I dijkstras algoritm kommer avsökningen av en nod att, i värsta fall, innebära att n-1 bågar undersöks, på totalt n noder. Detta innebär att komplexiteten uppskattas till att vara proportionell mot n^2. [1]

6 Slutsats

De problem som beskrevs i kapitel 2 Problemformulering och metoder till att lösa dessa problem har diskuterats i den utsträckning att lösningarna är tillämpningsbara till det kandidatprojekt denna studie är avsedd för.

6.1 Avsökning och kartläggning av en omgivning

Ingen slutsatts har ännu dragits.

6.2 Beräkning av optimal väg

Den optimeringsalgoritm som har valts är tillräcklig för att lösa det optimeringsproblem som uppkommer i kandidatprojektet. Den väg som ska optimeras är inte tillräckligt stor för att komplexiteten ska medföra någon märkbar skillnad i tidsåtgång, dessutom kommer man alltid fram till kortaste vägen. Denna metod är också till synes inte allt för komplicerad att implementera, varför denna metod har valts som lösning till optimeringsproblemet i fråga.

Referenser

[1] J. L. et al., *Optimeringslära*, 3. utg. Linköping: Studentlitteratur, 2003, kap. 8.4, Billigaste väg-problem, s. 190–202, ISBN: 978-91-44-05314-1.

[2] G. G. et al., "Improved techniques for grid mapping with rao-blackwellized particle filters", *IEEE Transactions on Robotics*, 2007, ISSN: 15523098. DOI: 10.1109/TRO.2006.889486. URL: http://www.informatik.uni-freiburg.de/.