

Designspecifikation Minröjningsbandvagn

Version 1.0

Utgivare: Niklas Olleson
Datum: 9 oktober 2012



Status

Granskad	JC & JD	2012-10-01
Godkänd	JC	2012-10-09

Kursnamn: Reglerteknisk projektkurs, CDIO
Projektgrupp: MinenMarker (MM)
Kurskod: TSRT10
Projekt: Minröjningsbandvagn

E-post: minenmarker@googlegroups.com
Dokumentansvarig: Niklas Olleson
Ansvarigs e-post: nikol038@student.liu.se
Dokumentnamn: designspec-1.0.pdf

Projektidentitet

Gruppens e-post: minenmarker@googlegroups.com
Hemsida: <http://www.isy.liu.se/edu/projekt/reglerteknik/2012/minrojningsrobot/>
Beställare: Jonas Callmer (JC), Linköpings universitet
Kund: Torbjörn Crona, Saab Bofors Dynamics
Kursansvarig: Daniel Axehill och David Törnqvist, Linköpings universitet
Projektledare: Staffan Sjöqvist
Handledare: Björn Johansson, Carl Nordheim, Stefan Thorstensson, Saab Bofors Dynamics, Johan Dahlin (JD), ISY

Gruppmedlemmar

Namn	Ansvarsområde	Telefonnummer	E-post (@student.liu.se)
Staffan Sjöqvist (SS)	Projektledare	073-6857771	stasj403
Niklas Ollesson (NO)	Dokument- och Designansvarig	070-2902655	nikol038
Viktor Edman (VE)	Testansvarig	073-0223498	viked065
Fredrik Danielsson (FD)	Informationsansvarig	073-0302889	freda001
Jonas Wallin (JW)	Navigeringsansvarig	070-6764791	jonwa237
Josef Högberg (JH)	AI-ansvarig	070-7171131	josh0859
Jonathan Nilsson (JN)	Mjukvaruansvarig	072-0234148	jonni416

Dokumenthistorik

Version	Datum	Ändringar	Utförda av	Granskad
0.1	2012-09-28	Första utkastet	Alla	Alla
0.2	2012-10-05	Andra utkastet. En grundlig genomgång av hela dokumentet. I synnerhet har positionerings- & karteringsdelsystemet utökats och avsnittet om sensorer och hårdvara förbättrats.	Alla	SS, JH, FD
0.3	2012-10-08	Tredje utkastet. Ändrat stavfel, fel i dokumentstrukturen. Vissa bilder är ändrade och förtydningar gjorda på olika ställen, främst i ruttplanering och modeller	JW, VE	JW
1.0	2012-10-09	Godkänd av beställare	SS, NO	JC

Kursnamn: Reglerteknisk projektkurs, CDIO
Projektgrupp: MinenMarker (MM)
Kurskod: TSRT10
Projekt: Minröjningsbandvagn

E-post: minenmarker@googlegroups.com
Dokumentansvarig: Niklas Ollesson
Ansvarigs e-post: nikol038@student.liu.se
Dokumentnamn: designspec-1.0.pdf

Innehåll

1	Inledning	1
2	Systemöversikt	1
3	Gemensamma datastrukturer och programflöde	2
3.1	Programflöde	2
3.2	Huvudklasser	2
3.3	Trådning	3
3.3.1	Trådsynkronisering	4
3.4	Variabelkonventioner i dokument	4
3.5	Koordinatsystem	4
3.5.1	Globalt koordinatsystem	5
3.5.2	Lokalt koordinatsystem	5
3.5.3	Rutkoordinatsystem	5
3.5.4	Bandvagnskoordinatsystem	5
3.6	Bandvagnens positionshistorik	7
3.7	Vägpunkter	7
3.8	Sannolikhetskartor över operationsområdet	7
3.8.1	Avsökningssannolikhetskarta	7
3.8.2	Hindersannolikhetskarta	7
4	Delsystem: Basstation	7
4.1	Gränssnitt	7
4.2	Klassdiagram	8
5	Delsystem: Bandvagn	8
5.1	Underdelsystem: Sensorer	8
5.1.1	IMU	8
5.1.2	Odometrar	9
5.1.3	GPS	10
5.1.4	Ultraljudssensorer	11
5.2	Underdelsystem: Framdrivning	12
5.2.1	Hårdvara	12
5.2.2	Kommunikationsprotokoll	13
5.3	Underdelsystem: Mindetektering	13
5.4	Underdelsystem: Hinderdetektering	13
5.5	Modeller	15
5.5.1	Fordonsmodell platt värld	15
5.5.2	Mätmodell	16
5.5.3	Alternativ till fordons- och mätmodell	17
5.5.4	Mindetekteringsmodell	18
5.6	Underdelsystem: Positionering och kartering	19
5.6.1	Extended Kalman Filter (EKF)	20
5.6.2	Kartering	20
5.6.3	Kartuppdatering	21
5.7	Underdelsystem: Ruttplanering	23

5.7.1	Ruttplaneringskarta	23
5.7.2	Avsökningss algoritmer	25
5.7.3	Markering av objekt på ruttplaneringens interna karta	26
5.7.4	Hantering av hinder	26
5.7.5	Område utan GPS-täckning	28
5.7.6	A*-algoritmen	30
5.7.7	Avslutande av minsökning	30
5.8	Underdelsystem: Reglering	31
5.8.1	Reglera efter vägg	32
6	Kommunikation mellan basstation och bandvagn	32
7	Kodstandard	32
7.1	Stil	32
7.2	Namngivning	35
7.3	Riktlinjer	35
7.3.1	Klasser	36
7.4	Basstationen	36
7.5	Bandvagnen	36
7.5.1	Kompilering	36



Förkortning	Beskrivning
PWM	"Pulse Width Modulation"
IMU	"Inertial Measurement Unit"
ARM	"Advanced RISC Machine"-processor
RISC	"Reduced Instruction Set Computer"
UTF-8	"UCS Transformation Format—8-bit"
UCS	"Universal Character Set"
GNU	"GNU's Not Unix"
GPL	"General Public License"
GCC	"GNU Compiler Collection"

Tabell 1: Förkortningar

1 Inledning

Saab Bofors Dynamics vill utveckla en minröjningsbandvagn som ska kunna detektera minor. Detta projekt är en fortsättning på tidigare projekt som påbörjades våren 2009 av projektgruppen O'hara's med utveckling av specifikationer, kontrollprogram, nätverkskommunikation, navigationstekniker med mera. Under hösten 2009 fortsatte projektgruppen Carpe Locus arbetet och utrustade bandvagnen med fjärrstyrning samt motorreglering för framdrivning. Bandvagnen utrustades även med en GPS för att kunna följa en fördefinierad brytpunktsbana. Hösten 2010 fortsatte projektgruppen 8Yare som förde över tidigare funktionalitet till en industridator och installerade en stereokamera av typen Bumblebee 2. Det senaste projektet var hösten 2011 då projektgruppen iMAP utökade bandvagnens funktionalitet med att genom hjälp av en kamera och sensorer bygga upp en 3D-karta av ett område.

Målet med detta projekt är att detektera minor samt att arbeta med bandvagnens positionering. Dessutom är målet att man ska säkerställa att bandvagnen avsoekt ett visst område.

Designspecifikationen är en översikt över det system för autonom minröjning som ska utvecklas. Kraven som specificerar vad själva minröjningen består av och vad hela systemet ska klara av definieras i detalj i kravspecifikationen, se [12]. Designspecifikationen innehåller en kort beskrivning av alla delsystem som ingår i systemet.

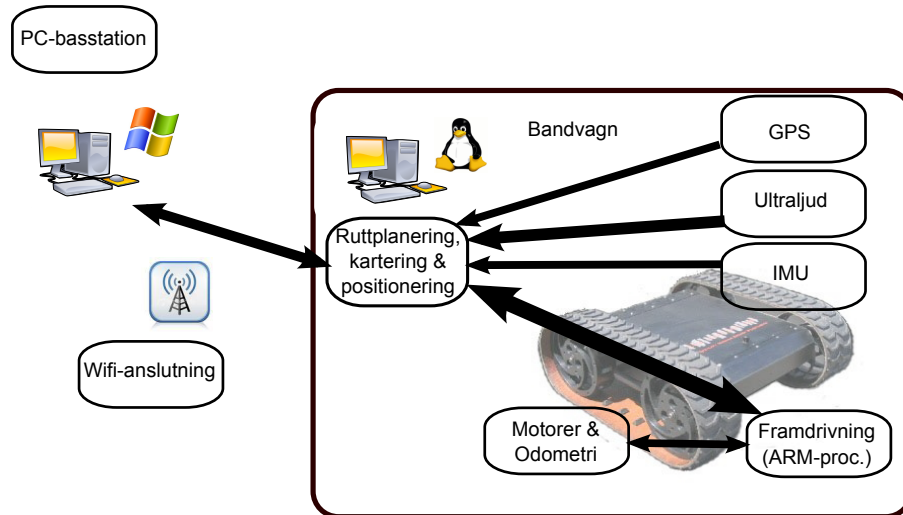
Designspecifikationen följer LIPS-mallen, se [1] och [2] för mer information.

I designspecifikationen används en del förenklade förkortningar och dessa definieras i tabell 1.

2 Systemöversikt

Minröjningsbandvagnen består av två huvudsystem, basstationen respektive bandvagnen, som i sin tur är uppdelade i mindre delsystem, se figur 1. Delsystemens syften och övergripande uppgifter är specificerade i detta dokument. Tanken med indelningen är att varje delsystem ska kunna konstrueras och testas individuellt i så stor mån som möjligt. Därefter ska dessa sättas ihop till större system och slutligen till ett system, det vill säga minröjningsbandvagnen.

Varje delsystem berör en eller flera fysiska komponenter och dessa presenteras också för varje delsystem. Notera att vissa delsystem delar vissa komponenter. I figur 1 visas en överblick över minröjningsbandvagnen med en skiss över informationsflödet mellan de olika delsystemen. Inspiration till figuren har hämtats från [4]



Figur 1: Skiss över hela minröjningsbandvagnen.

3 Gemensamma datastrukturer och programflöde

I följande stycken beskrivs programmets struktur och viktiga datatyper.

3.1 Programflöde

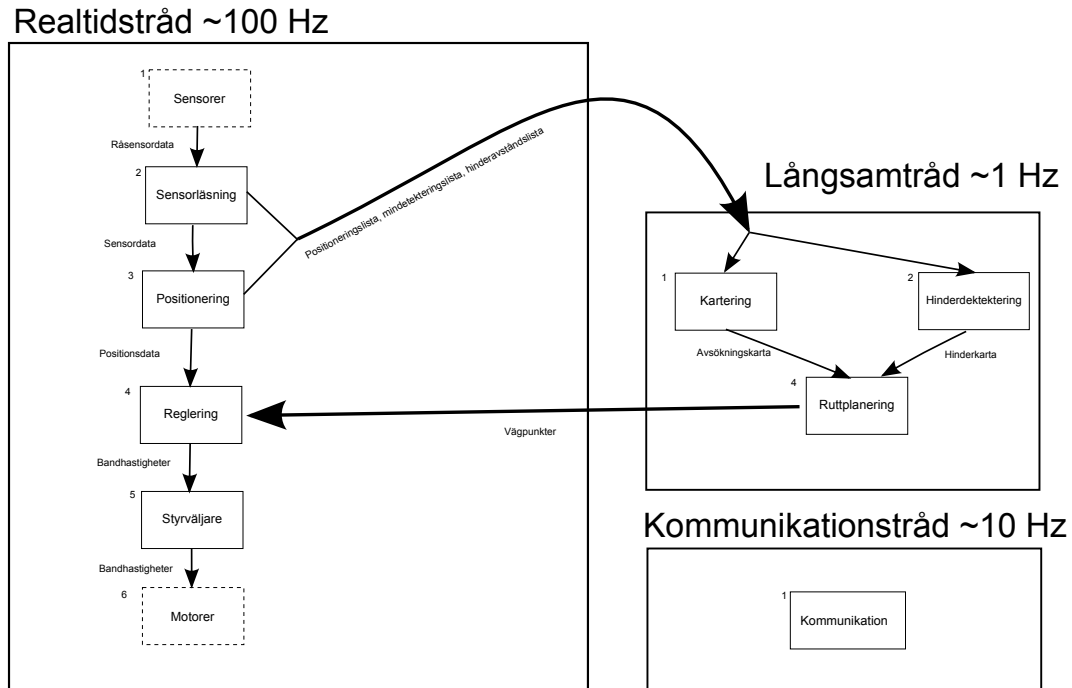
I figur 2 visas de viktigaste dataflödena och delsystemen i bandvagnen. De stora rutorna innehåller de delsystem som körs i respektive tråd. De små heldragna rutorna är de olika delsystemen och de streckade är hårdvara. Pilarna representerar data. All data ägs/skrivs från den sida som pilen pekar från och den läses från den sidan som pilen pekar till. Då pilar passerar trådgränser så kommer synkronisering att ske enligt stycke 3.3.1. Ordningen som de inbördes delsystemen exekveras i markeras med en siffra invid delsystemsrutorna.

Efter att sensordelssystemet i realtidstråden har exekverats så finns sensordata tillgänglig för alla andra delsystem att läsa. Dessa pilar är inte inritade i figur 2.

3.2 Huvudklasser

Alla delsystem kommer att modelleras som singleton-klasser. En singleton-klass är en klass som det endast finns en av. Klassnamnen kommer att vara följande:

- Sensorläsning \Rightarrow Sensor
- Positionering \Rightarrow Positioning
- Mindetektering \Rightarrow MapperMine
- Kartering \Rightarrow MapperPosition
- Hinderdetektering \Rightarrow MapperObstacle
- Kommunikation \Rightarrow Communication
- Reglering \Rightarrow Regulator



Figur 2: Dataflödet i bandvagnen.

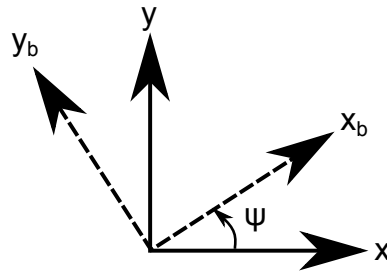
- Styrväljare ⇒ Steering
- Ruttplanering ⇒ RoutePlanner

De viktigaste databärande klasserna heter:

- Kartorna ⇒ Map
- Vägpunkter ⇒ Waypoint
- Larvbandhastigheter ⇒ WheelData
- Odometri data ⇒ OdometerData
- IMU-data ⇒ IMUData
- GPS-data ⇒ GPSData
- Avståndsdata ⇒ RangeData
- Positioneringsdata ⇒ StateData

3.3 Trådning

Det kommer att finnas tre trådar, en som körs i ungefär 100 Hz och ansvarar för allt som skall ske i realtid och beror på sensorvärden. I den andra körs det som inte behöver köras i realtid och som är mer tidskrävande. Till denna del hör mindetektering, kartering, hinderdetektering och ruttplanering. Den sista tråden är kommunikationstråden som sköter dataflyttningar mellan basstationen och bandvagnen. Den körs i ungefär 10 Hz.



Figur 3: Figuren visar på hur bandvagnen förhåller sig till det lokala koordinatsystemet.

3.3.1 Trådsynkronisering

Den data som skall synkas mellan trådarna är länkade listor med vägpunkter, positioneringsdata, hinderavståndsdata samt mindetekteringsdata. Dessa datastrukturer kommer att synkroniseras med "mutual exclusion locks" för att ingen skall läsa eller skriva från dem samtidigt. Hänsyn kommer att tas till att realtidstråden alltid kommer att kunna exekveras på utsatt tid.

3.4 Variabelkonventioner i dokument

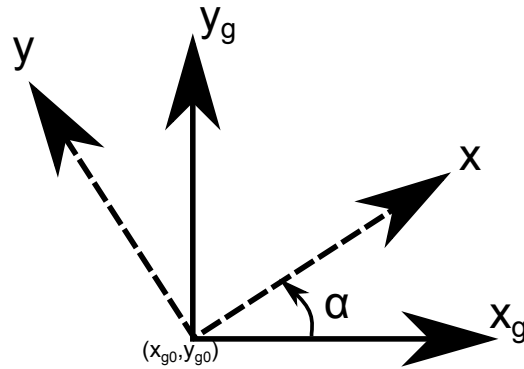
I dokumentet används variabler och dessa är enligt följande punktlista

- x , y och z är koordinater i det lokala koordinatsystemet
- i och j är koordinater i rutkoordinatsystemet
- x_b , y_b och z_b är koordinater i bandvagnens koordinatsystem
- \mathbf{x} är en tillståndsvektor
- x^i är element i i tillståndsvektorn \mathbf{x}
- $\hat{\mathbf{x}}$ är skattningen av tillståndsvektorn \mathbf{x}
- \hat{x}^i är element i i $\hat{\mathbf{x}}$
- X och Y är stokastiska variabler i med Gaussisk fördelning för positionsskattning av x och y
- x_s är positionen för en sensor
- x_h är positionen för ett hinder
- x_k är variabeln vid sampel nummer k
- ψ är bandvagnens riktning relativt det lokala koordinatsystemet, positiv riktning runt z-axeln. Se figur 3.
- σ_x^2 är variansen för en variabel x

3.5 Koordinatsystem

Systemet använder sig av fyra olika koordinatsystem.

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Ollesson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



Figur 4: Transformerering från det globala koordinatsystemet till det lokala.

3.5.1 Globalt koordinatsystem

Den GPS som används specificerar koordinatsystemet i latitud och longitud och det är detta system som kallas för globalt koordinatsystem.

3.5.2 Lokalt koordinatsystem

Det lokala koordinatsystemet är en roterad och translaterad modell av det globala koordinatsystemet. Detta koordinatsystem beror av användaren angivna området, med origo längst ner till vänster. Koordinataxlarna representeras som x åt höger, y uppåt och z ut ur kartan. Origo väljs som punkten längst ner till vänster på kartan som i figur 5.

Omräkning till det lokala koordinatsystemet kan göras enligt

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix} \begin{pmatrix} x_g - x_{g0} \\ y_g - y_{g0} \end{pmatrix} \quad (1)$$

givet det lokala koordinatsystemets origo-koordinater i det globala koordinatsystemet, (x_{g0}, y_{g0}) . I ekvationen anger α rotationen av det lokala koordinatsystemet relativt det globala enligt figur 4. Vinkeln α bestäms givet att hörnen på det av användaren angivna avsökningsområdet är kända. Givet att punkten nere till höger på avsökningsområdet är (x_{g1}, y_{g1}) bestäms α enligt

$$\tan(\alpha) = \frac{y_{g1} - y_{g0}}{x_{g1} - x_{g0}} \quad (2)$$

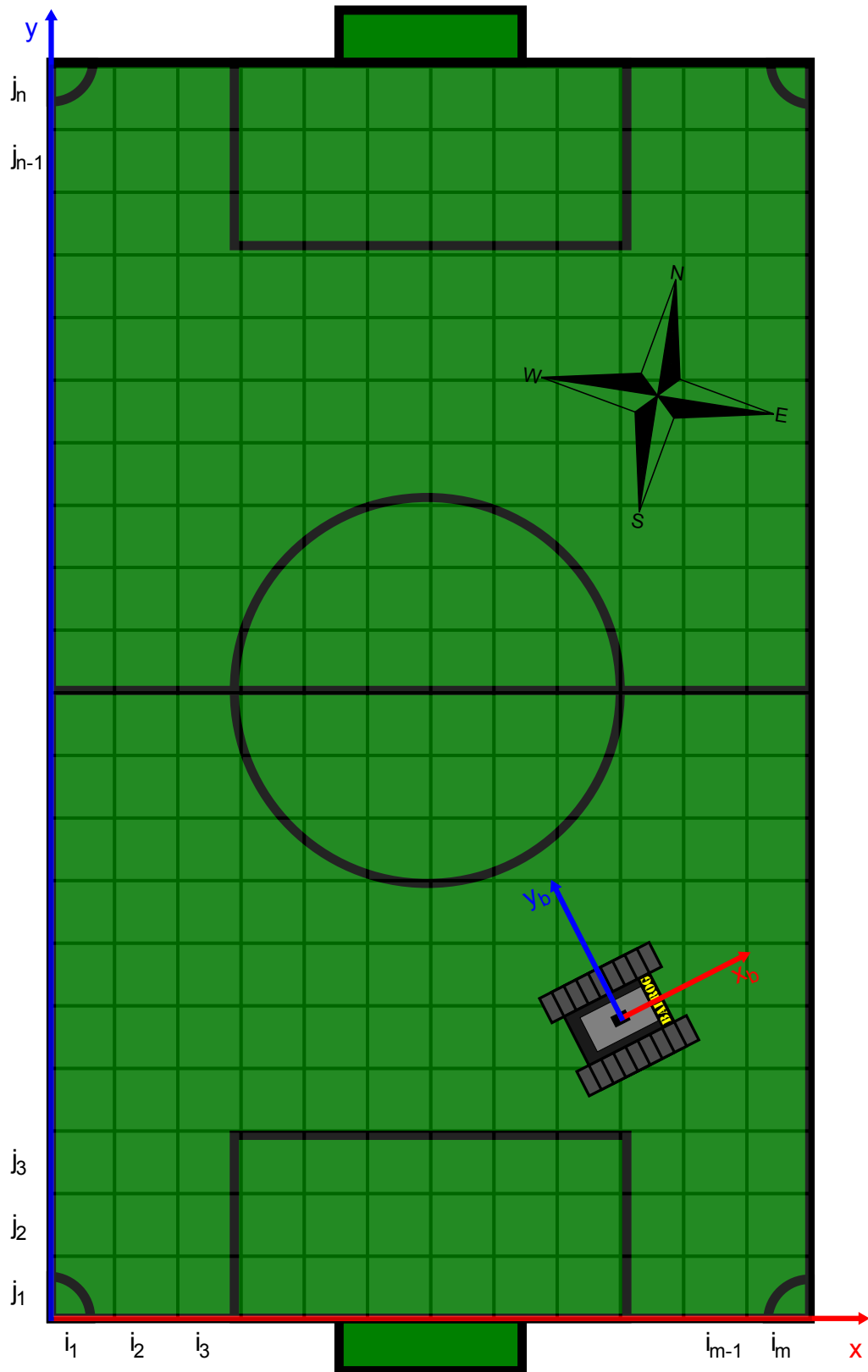
3.5.3 Rutkoordinatsystem

Rutkoordinatsystemet är det lokala koordinatsystemet diskretiserat, det vill säga uppdelat i rutor. Koordinataxlarna representeras som i åt höger och j uppåt. Origo väljs som punkten längst ner till vänster på kartan.

3.5.4 Bandvagnskordinatsystem

I bandvagnskordinatsystemet pekar x_b rakt framåt, y_b pekar åt vänster och z_b upp från bandvagnen. Origo specificeras som mitten av bandvagnen.

I figur 5 visas alla koordinatsystem i en och samma bild.



Figur 5: De fyra koordinatsystemen i en och samma bild där det globala koordinatsystemet representeras av en kompass.



3.6 Bandvagnens positionshistorik

All historik för bandvagnens position sparas i en länkad lista. Varje element i listan innehåller bandvagnens skattade position, hastighet, vinklar och vinkelhastigheter i kartkoordinatsystemet, med tillhörande kovariansskattningar. Till varje element hör även resultatet från mindetekteringen, det vill säga om bandvagnen är nära en mina eller inte, samt när skattningen gjordes.

3.7 Vägpunkter

Vägpunkter är positioner i det lokala koordinatsystemet som bandvagnen ska besöka. Vägpunkterna sparas som en länkad lista där varje element i listan innehåller vägpunktens koordinater uttryckt som x och y i rutkoordinatsystemet.

3.8 Sannolikhetskartor över operationsområdet

Två stycken sannolikhetskartor definieras som beskriver operationsområdet. Alla kartor representeras av en matris bestående av element som svarar mot en viss ruta av avsökningso-mrådet. Elementen innehåller flyttalsvärden som anger sannolikheten att det aktuella området är avsökt eller innehåller ett hinder.

3.8.1 Avsökningssannolikhetskarta

En karta över sannolikheten att en viss del av området är avsökt sparas som en matris med flyttalsvärden. I matrisen svarar 0 mot garanterat inte avsökt och 1 svarar mot garanterat avsökt. Varje element i matrisen representerar en yta av 25×25 centimeter av operationsområdet. Denna storlek kan emellertid justeras om inte önskad prestanda uppnås. Detta element anger således sannolikheten att motsvarande område har avsökts.

3.8.2 Hindersannolikhetskarta

En karta över sannolikheten att en viss del av området är ett hinder sparas som en matris med flyttalsvärden. I matrisen svarar 0 mot garanterat fritt område och 1 svarar mot garanterat hinder på området. Varje element i matrisen representerar en yta av 25×25 centimeter av operationsområdet. Detta element anger således sannolikheten att motsvarande område är ett hinder.

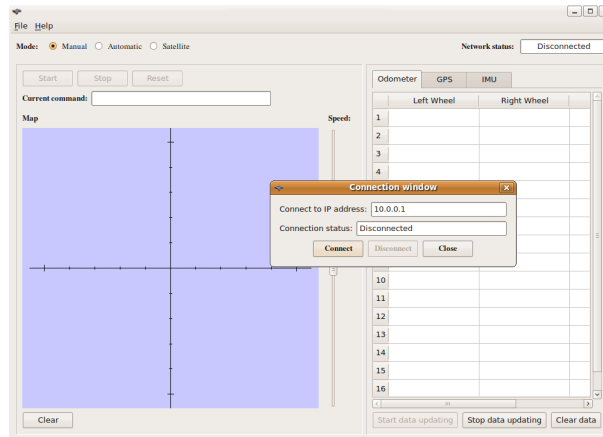
4 Delsystem: Basstation

Denna del av dokumentet beskriver vad som ingår i delsystemet basstation. Delsystemet består av en dator som kan kommunicera trådlöst med bandvagnen.

4.1 Gränssnitt

Användargränssnittet är i stort samma som använts av tidigare års grupper och kommer inte att beskrivas alltför detaljerat i detta dokument utan det går att läsa om det i tidigare års designspecifikation [5]. Grundstommen på användargränssnittet kan ses i figur 6.

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Ollesson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



Figur 6: Grafiska gränssnittet i basstationen

De skillnader som vår version av gränssnittet kommer att ha mot deras är att vi kommer att ta bort stöd för kamerahantering och bildanalys. Detta kommer att leda till att vi inte får något behov av externa bibliotek förutom Qt [7].

Gränssnittet har som funktion att ansluta till bandvagnen över nätverket och specificera avsökningsområdet som skall sökas efter minor. Därefter kommer kontinuerligt kartinformation att ritas ut i kartfönstret då bandvagnen söker av området.

Alternativt så styr användaren bandvagnen med piltangenterna och ett reglage för hastighet.

4.2 Klassdiagram

I figur 7 visas de klasser som finns i den befintliga mjukvaran. De klasser som är markerade med rött kommer att tas bort. De klasser som är markerad med gult kommer innehålla den största delen av våra modifikationer.

Kartgränssnittet kommer att utökas så att det kan visa sannolikhetskartor, där de sannolikhetskartor som vi genererar kommer att ritas ut.

5 Delsystem: Bandvagn

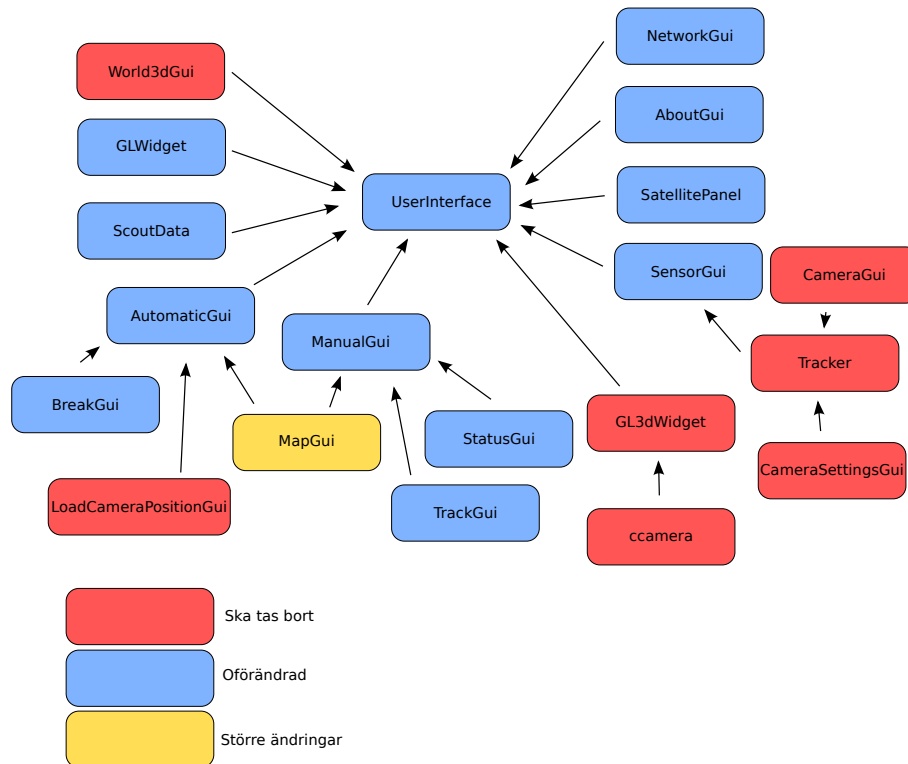
Denna del av dokumentet beskriver vad som ingår i delsystemet bandvagn.

5.1 Underdelsystem: Sensorer

Bandvagnen är utrustad med sensorer för att kunna navigera, detektera minor och skatta sin position. De sensorer som används specificeras nedan.

5.1.1 IMU

Från IMU:n läses accelerometer-, gyro- och magnetometerdata in. Data från IMU sparas i klassen IMUData. Data som sparas är acceleration i x_b -, y_b -, och z_b -led, vinkelhastighet



Figur 7: Klasserna i basstationens mjukvara.

runt x_b , y_b och z_b samt magnetfältstyrka i x_b , y_b och z_b enligt bandvagnens koordinat-system. Det sparas även en tidsstämpel för när data registrerades. IMU:n kommunicerar med ett serieprotokoll som kopplas till bandvagnen via USB.

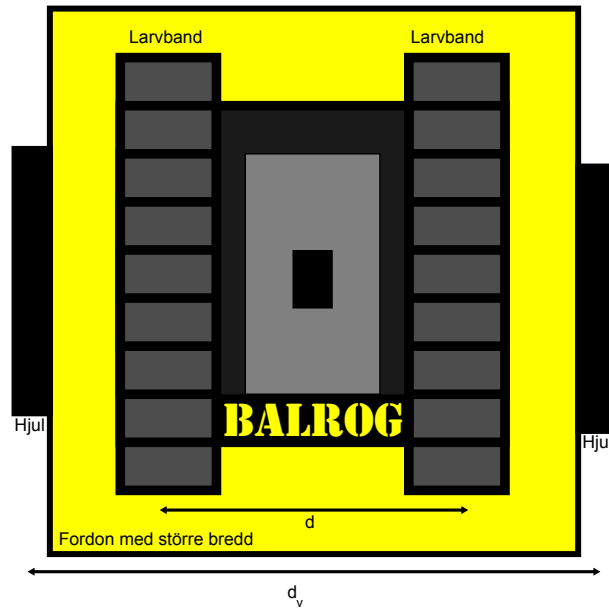
Den begränsning som finns för samplingstid av IMU-data är dels kommunikationsprotokoll över USB och dels hur ofta vi kan göra beräkningar för uppdatering av Kalmanfiltret. Då acceleration och vinkelhastigheter ändras snabbt vill vi sampla dessa ofta. Vi väljer således att utgå från en samplingshastighet för IMU-data på 50 Hz, som sedan kan justeras om det finns för mycket eller för lite beräkningskraft.

5.1.2 Odometrar

Odometerdata läses in från ARM-processorn, enligt kommunikationsprotokoll i tabell 3. Data från odometern sparas i klassen OdometerData. Data som sparas är en skattning av bandvagnens hastighet och vinkelhastighet givet hastighetsdata från odometern. Detta görs enligt nedan. Det sparas även en tidsstämpel för när data senast registrerades.

Odometern har 500 mätpunkter per varv på hjulet den mäter på. Detta get ett avstånd mellan mätningar på 1.1 millimeter, och alltså ett teoretiskt maximalt fel för odometermätningen på detsamma om man bortser från att banden kan slira mot underlaget.

Den begränsning som finns för samplingstid av odometrarna är framför allt att de uppdateras efter diskreta ändringar av larvbandens position. En samplingstid på 10 Hz används som utgångssamplingshastighet, och justeras sedan efter hand då beteendet hos odomet-rarna är mer väl utrett.



Figur 8: Modifierade modellen för vinkelhastighetsberäkning ur odometerdata för att kompensera för slir.

Odometermodell

Odometrarna mäter som nämnt avstånd som bandvagnens band har rört sig samt skattar deras hastighet. Denna hastighetsskattning räknas om till bandvagnens hastighet enligt

$$v = \frac{y_{o_r}^v + y_{o_l}^v}{2} \quad (3)$$

där $y_{o_l}^v$ och $y_{o_r}^v$ är bandvagnens respektive bands uppmätta hastighet. Enligt tidigare teknisk dokumentation för bandvagnen, se [4], behöver hänsyn här tas till slir hos bandvagnen då den svänger. Den lösning som angavs var att modellera bandvagnen som ett fordon med större bredd än den faktiska bandvagnen och med hjul istället för larvband. Detta illustreras i figur 8, och leder till en vinkelhastighet som

$$\omega = \frac{y_{o_r}^v - y_{o_l}^v}{d_v} \quad (4)$$

där v_l och v_r återigen är bandvagnens respektive bands hastighet och d_v är given enligt figur 8.

Från ekvationerna (3) och (4) kan larvbandens respektive hastigheter, givet bandvagnens hastighet och vinkelhastighet, räknas fram. Resultatet blir då

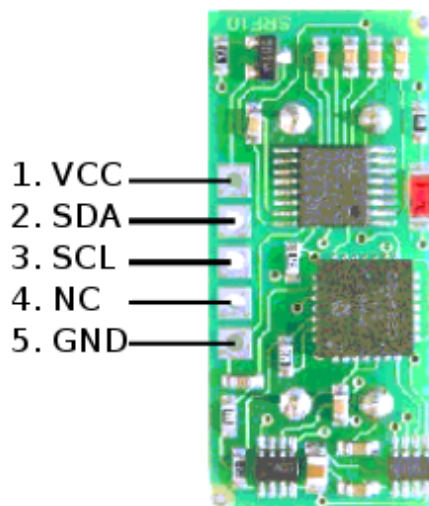
$$y_{o_r}^v = \frac{2v + d_v\omega}{2} \quad (5)$$

$$y_{o_l}^v = \frac{2v - d_v\omega}{2} \quad (6)$$

som ger larvbandens hastigheter.

5.1.3 GPS

Data sparas i det lokala kordinatsystemet. Även antal satelliter tillgängliga vid mätning samt en tidsstämpel för när data registrerades sparas. Omräkning till det lokala koordi-



Figur 9: Anslutningar på ultraljudssensorerna SRF10

Anslutning	Beskrivning
VCC	Drivspänning +5 Volt
SDA	I2C Data
SCL	I2C Klocka
NC	Används ej (Not Conncted)
GND	Jord

Tabell 2: Ultraljudssensorernas anslutningar

natsystemet kan göras enligt (1), givet det lokala koordinatsystemets koordinater i det globala. Den data som läses från GPS är i longitud och latitud, det vill säga med koordinataxlar parallella med vädersträcken.

Samplingshastigheten för GPS begränsas framför allt av att hårdvaran inte gör nya mätningar ofta. Utgångssamplingshastighet väljs till 5 Hz och justeras sedan efter att test gjorts av GPS där dess uppdateringsfrekvens noterats.

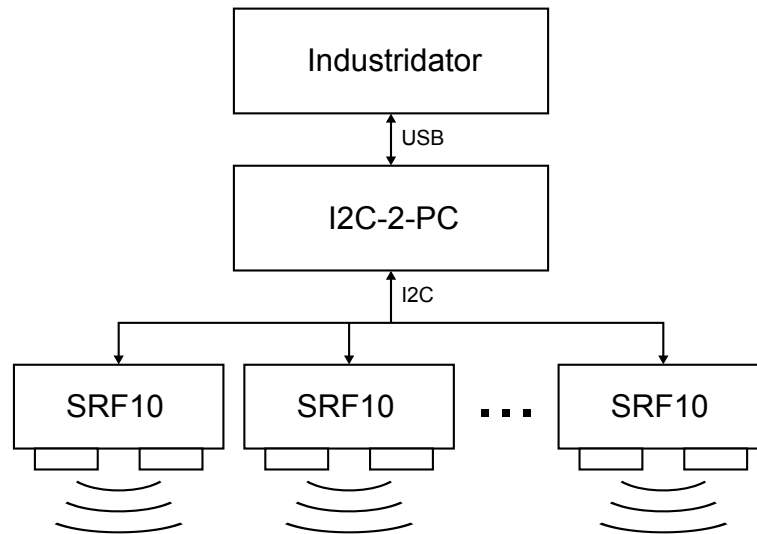
5.1.4 Ultraljudssensorer

Bandvagnen kommer att utrustas med ultraljudssensorer för att kunna upptäcka hinder. De sensorer som används heter SRF10 och har en räckvidd på mellan 6 cm och 6 meter.

Sensorgränssnitt

Dessa sensorer använder en kommunikationsbuss som heter I2C. I2C-bussen har stöd för att flera olika enheter som konfigurerats med olika adresser kommunicerar över samma fysiska ledare. Figur 10 visar kopplingen mellan sensorerna och industridatorn. Då industridatorn saknar ett gränssnitt för att kommunicera via I2C har vi valt att använda oss av en konverterare (I2C-2-PC) som hanterar I2C-kommunikationen via antingen USB eller RS-232 som datorn kan använda.

Ultraljudssensorerna har fem anslutningar som anges i figur 9 med förklaringar i tabell 2.



Figur 10: Inkoppling av ultraljudssensorer

Kommunikationsprotokoll

De olika ultraljudssensorerna kan ställas in på olika adresser för att kunna kommunicera på samma I2C-buss. Konverteraren kontrolleras genom ett enkelt textbaserat protokoll som beskrivs i detalj i dess datablad, se [9]. Ultraljudssensorn har fyra olika interna register som kan läsas och skrivas från I2C-bussen. För en mer detaljerad beskrivning av sensorns funktioner och gränssnitt hänvisas till den tekniska specifikationen för ultraljudssensorerna, se [8].

En avståndsmätning startas genom att skicka ett kommando till sensorns kommando-register. När mätningen är klar läses resultatet tillbaka från sensorns resultatregister. Resultatet av mätningen returneras i centimeter.

Ultraljudssensorernas samplingshastighet begränsas framför allt av hur snabbt sensorerna uppdaterar mätningar. En utgångssamplingsfrekvens på 5 Hz väljs, och justeras sedan efter att test gjorts av hur ultraljudssensorerna beter sig.

5.2 Underdelsystem: Framdrivning

Bandvagnen drivs av motorer kopplade till larvband och framdrivningsdelsystemet har till uppgift att hantera alla rörelser som bandvagnen kan tänkas göra. Framdrivningsdelsystemet ska reglera motorerna utifrån den önskade hastigheten från regleringsdelsystemet samt sensordata från odometrarna.

5.2.1 Hårdvara

Delsystemet för framdrivning består av ett utvecklingskort med en ARM-processor samt av två odometrar som är anslutna till motorerna och kopplade till utvecklingskortet. Mer information om hur hårdvaran fungerar finns beskrivet i designspecifikationen för projektgruppen Carpe Locus, se [3], som var den grupp som utvecklade framdrivningen på bandvagnen under ett tidigare projekt.



Uppgift	Fråga	Svar
Ställa in hastighet	sVXXYY	sVOK
Hämta odometerdata	upDi	LXXRYY
Hämta bandhastigheter	upVe	LXXRYY
Nollställ ARM-enheten	rese	reseOK

Tabell 3: Kommunikation mellan ARM-enheten och navigationsenheten.

5.2.2 Kommunikationsprotokoll

Från tidigare projekt finns kommunikation med framdrivningsdelsystemet och ARM-processorn, se [3]. Framdrivningsdelsystemet kommunicerar med regleringsdelsystemet som skickar börvärden för regulatorerna till framdrivningsdelsystemet. Framdrivningsdelsystemet kan skicka tillryggalagd sträcka för de två olika banden till regleringsdelsystemet. Framdrivningsdelsystemet skickar endast data när regleringsdelsystemet frågar efter data. En lista över de möjliga kommandon som kan sändas och mottas kan ses i tabell 3.

Till framdrivningsdelsystemet kan man skicka ett styrkommando som innehåller två hastigheter, en för vardera larvband. Det meddelande som skickas har följande utseende: 'sVXXYY'. XX är två bytes som anger vänster larvbands hastighet i millimeter per sekund och YY anger analogt hastigheten hos höger larvband. Mest signifikant bit är hastighetens tecken. Som svar får vi 'sVOK'(set velocity ok).

Framdrivningsdelsystemet returnerar odometerdata då den mottar kommandot 'upDi'. Meddelandet som returneras har följande format: 'LXXRYY'. XX är körd sträckändring sedan senaste mätning på vänster larvband i tiondels millimeter och YY är analogt sträckändringen i tiondels millimeter på höger larvband. Den mest signifikanta biten är teckenbit.

Det är även möjligt att nollställa variablerna för körd sträcka på larvbanden för framdrivningsdelsystemet genom att sända meddelandet 'rese'. Svaret blir då 'reseOK' (reset ok).

5.3 Underdelsystem: Mindetektering

Delsystemet mindetektering har som uppgift att detektera närliggande minor under områdets avsökning. Detekteringen av minor görs genom att data från magnetometern i IMU:n analyseras. Då denna magnetometer mäts vid magnetfältstyrka 1.2 Gauss (jämför jordens magnetfältstyrka på ~ 0.5 Gauss) kommer den mättas snabbt då den närmar sig en magnet. Detekteringen görs därför som ett beslut huruvida en mina är nära bandvagnen eller inte, givet om sensorn har mättats eller inte. Om tester visar att störningar kan mäta sensorn görs först en medianfiltrering av sensordata.

Systemet tar ett beslut om minan som detekterats är en ny eller gammal mina beroende på avståndet från bandvagnens skattade position till de minor som redan finns detekterade.

Minornas position skattas i samma EKF som tillstånden i fordonsmodellen. Se mer om mindetekteringsmodellen i avsnitt 5.5.4.

5.4 Underdelsystem: Hinderdetektering

Delsystemet hinderdetektering har som uppgift att detektera hinder. Hinder detekteras med ultraljudssensorerna som bandvagnen är utrustade med. Ett hinders position beskrivs

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Ollesson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



idealt enligt

$$\mathbf{x}_h = \mathbf{x}_{s_i} + f_{s_i}(\psi)r_i, \quad (7)$$

där \mathbf{x}_{s_i} är sensor i :s position i det lokala koordinatsystemet, $f_{s_i}(\psi)$ är en rotationsmatris, ψ är bandvagnens riktning och r_i är avståndet mellan sensor i och hinder. Detta är dock inte möjligt och därför kommer hindrens position att skattas genom

$$\begin{pmatrix} x_h \\ y_h \end{pmatrix} = \begin{pmatrix} \hat{x}_{s_i} \\ \hat{y}_{s_i} \end{pmatrix} + f_{s_i}(\hat{\psi})\mathbf{y}_{r_i}, \quad (8)$$

där $(\hat{x}_{s_i}, \hat{y}_{s_i})^T$ är den skattade positionen för sensor i som fås från bandvagnens skattade position tillsammans med en given offset från bandvagnens centrum. $\hat{\psi}$ är den skattade vinkeln som bandvagnen har relativt det lokala koordinatsystemet. \mathbf{y}_{r_i} är en mätning med ultraljudssensor i och $f_{s_i}(\hat{\psi})$ är en rotationsmatris som roterar mätningen i radiell led till det lokala koordinatsystemets (x, y) -koordinater. $f_{s_i}(\hat{\psi})$ beror av vilken sensor s_i som har gjort mätningen (då de sitter på olika sidor av bandvagnen) och av den skattade vinkeln $\hat{\psi}$.

\mathbf{y}_{r_i} är en 2×1 -vektor som består av ett nollskilt element i radiell led från sensorn sett, $\hat{\mathbf{r}}$, och ett nollelement i $\hat{\mathbf{r}}_{\perp}$ -led. Dessa riktningar blir olika beroende på vilken sida av bandvagnen som sensorn i fråga sitter på. Mätningen ser dock alltid ut enligt

$$\begin{aligned} \mathbf{y}_{r_i} &= \begin{pmatrix} r_i \\ 0 \end{pmatrix} (\hat{\mathbf{r}} \quad \hat{\mathbf{r}}_{\perp}) + \mathbf{e}_{r_i}, \\ \mathbf{e}_{r_i} &\sim N \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{r_i}^2 & 0 \\ 0 & \sigma_{r_{\perp,i}}^2 \end{bmatrix} \right). \end{aligned} \quad (9)$$

Ultraljudssensorerna mäter avståndet i en droppformad omgivning framför sig. Detta gör att mätningarna har en varians i både $\hat{\mathbf{r}}$ - och $\hat{\mathbf{r}}_{\perp}$ -led, se figur 11. Denna droppform gör även att variansen i $\hat{\mathbf{r}}_{\perp}$ -led beror av avståndet r_i mellan sensor och hinder. I och med detta så bör även ovanstående mätbrus vara korrelerade, men detta bortses från till en början för att inte erhålla en allt för avancerad modell. Däremot modelleras mätbruset i $\hat{\mathbf{r}}_{\perp}$ -led som en funktion av r_i , denna funktion approximeras dock som en konformad (linjär) funktion enligt

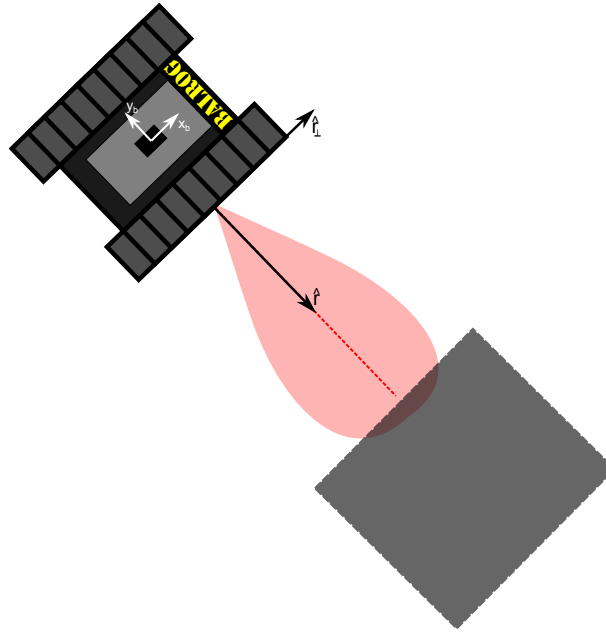
$$\sigma_{r_{\perp,i}}^2 = kr_i, \quad (10)$$

där k är en lutningskoefficient som behöver tas fram empiriskt för varje typ av ultraljudssensor som används. Variansen $\sigma_{r_i}^2$ antas vara konstant.

Funktionerna $f_{s_i}(\hat{\psi})$ kommer som sagt att bero av vilken sida en sensor sitter på och med vilken vinkel $\hat{\psi}$ som bandvagnen är vriden. Dessa är definierade enligt

$$\begin{aligned} f_{s_i,r}(\psi) &= \begin{pmatrix} \sin \psi & -\cos \psi \\ \cos \psi & \sin \psi \end{pmatrix}, \\ f_{s_i,l}(\psi) &= -f_{s_i,r}(\psi), \\ f_{s_i,f}(\psi) &= \begin{pmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{pmatrix}, \\ f_{s_i,t}(\psi) &= -f_{s_i,f}(\psi), \end{aligned} \quad (11)$$

där ψ är bandvagnens vinkel.



Figur 11: Illustration av hur en avståndsmätning ser ut för en mätning från bandvagnens högra sida.

Dessa rotationer kommer att göra om mätningarna från radiell led, och bruset för dessa, till de lokala (x, y) -koordinaterna. Detta kan exempelvis se ut som

$$\begin{aligned} \begin{pmatrix} x_h \\ y_h \end{pmatrix} &= \begin{pmatrix} \hat{x}_s \\ \hat{y}_s \end{pmatrix} + \begin{pmatrix} r \sin \hat{\psi} \\ -r \cos \hat{\psi} \end{pmatrix} + \mathbf{e}_h, \\ \mathbf{e}_h &\sim N \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_{r_i}^2 \sin \hat{\psi} + P_k^{xx} & \sigma_{r_{\perp,i}}^2 \cos \hat{\psi} + P_k^{xy} \\ -\sigma_{r_i}^2 \cos \hat{\psi} + P_k^{yx} & \sigma_{r_{\perp,i}}^2 \sin \hat{\psi} + P_k^{yy} \end{bmatrix} \right), \end{aligned} \quad (12)$$

för en sensor riktad åt höger.

Ur detta ovan kommer en hindr sannolikhetskarta att genereras på samma sätt som för positionen i avsnitt 5.6.2.

5.5 Modeller

Nedan specificeras de matematiska modeller som används för att beskriva delar av bandvagnens fysik.

5.5.1 Fordonsmodell platt värld

Bandvagnen modelleras med en tillståndsmo­dell i det lokala koordinatsystemet på formen

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k, \quad (13)$$

där \mathbf{x}_k är en tillståndsvektor vid sampel k . f är en olinjär funktion som beskriver bandvagnens dynamik och \mathbf{w}_k är processbruset. Tillståndsvektorn har följande form

$$\mathbf{x}_k = \begin{pmatrix} x_k \\ y_k \\ v_k \\ a_k \\ \psi_k \\ \omega_k \end{pmatrix}, \quad (14)$$

där x_k och y_k är bandvagnens position i lokala koordinatsystemet, v_k och a_k är hastighet respektive acceleration i bandvagnens färdriktning och ψ_k och ω_k är bandvagnens riktning samt vinkelhastighet i det lokala koordinatsystemet.

Hastigheten v och accelerationen a antas endast kunna vara riktade i bandvagnens färdriktning. För bandvagnens dynamik antas en modifierad konstant accelerationsmodell enligt

$$\mathbf{x}_{k+1} = \begin{pmatrix} x_k + Tv_k \cos(\psi_k + \frac{\omega_k T}{2}) + \frac{T^2}{2} a_k \cos(\psi_k + \frac{\omega_k T}{2}) \\ y_k + Tv_k \sin(\psi_k + \frac{\omega_k T}{2}) + \frac{T^2}{2} a_k \sin(\psi_k + \frac{\omega_k T}{2}) \\ v_k + Ta_k \\ a_k \\ \psi_k + T\omega_k \\ \omega_k \end{pmatrix} + \mathbf{w}_k, \quad (15)$$

där T är sampeltiden, eller snarare tiden mellan det senaste samplet och det nuvarande samplet. Processbruset \mathbf{w}_k antas vara gaussiskt fördelat med medelvärde 0 och kovariansmatris $Cov(\mathbf{w}_k) = Q_k$.

Q_k antas initialt, för enkelhets skull, vara en diagonalmatris enligt

$$Q_k = \begin{bmatrix} Q_k^{xx} & 0 & 0 & 0 & 0 & 0 \\ 0 & Q_k^{yy} & 0 & 0 & 0 & 0 \\ 0 & 0 & Q_k^{vv} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_k^{aa} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_k^{\psi\psi} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_k^{\omega\omega} \end{bmatrix}. \quad (16)$$

Eftersom modellen är en modifierad konstant accelerationsmodell så antas bruset komma in på de tillstånd som är högsta ordningen av derivator i modellen. I detta fall är det a och ω . Om man jämför med en linjär modell, se [6] kapitel 13.1.1, så resulterar detta i att om $T \leq 1$ så blir bruset desto mindre ju lägre ordning tillståndet har. Q_k^{xx} och Q_k^{yy} kommer alltså vara mindre än Q_k^{vv} som i sin tur är mindre än Q_k^{aa} . På samma sätt kommer $Q_k^{\psi\psi}$ att vara mindre än $Q_k^{\omega\omega}$.

5.5.2 Mätmodell

Detta avsnitt beskriver mätmodellen som används i avsnittet om EKF nedan. Den är definierad som

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{e}_k, \quad (17)$$

där y_k är mätvärden, h är en olinjär funktion som beskriver hur mätvärdena är relaterade

till tillstånden i fordonsmodellen i (14) och \mathbf{e}_k är mätbruset. Det som mäts beskrivs som

$$\mathbf{y}_k = \begin{pmatrix} y_k^x \\ y_k^y \\ y_k^a \\ y_k^\omega \\ y_{o_l,k}^v \\ y_{o_r,k}^v \end{pmatrix}, \quad (18)$$

där y_k^x och y_k^y är de positionsvärden som GPS ger ifrån sig omräknade till det lokala koordinatsystemet, y_k^a och y_k^ω är den från IMU uppmätta accelerationen respektive vinkelhastigheten och $y_{o_l,k}^v$, $y_{o_r,k}^v$ är den uppmätta hastigheten på respektive larvband.

Funktionen h relaterar dessa mätta värden till tillstånden enligt

$$h(\mathbf{x}_k) = \begin{pmatrix} x_k \\ y_k \\ a_k \\ \omega_k \\ \frac{2v_k + d_v \omega_k}{2} \\ \frac{2v_k - d_v \omega_k}{2} \end{pmatrix}, \quad (19)$$

där d_v är bandvagnens virtuella bredd och övriga variabler enligt tillståndsvektorn (14).

Mätbruset \mathbf{e}_k antas vara gaussiskt med medelvärde 0 och kovariansmatris $\text{Cov}(\mathbf{e}_k) = R_k$. Då x_k och y_k mäts med samma sensor (GPS) kan dessa antas vara korrelerade, vilket i sin tur innebär att korskovariansen mellan dessa är nollskild. Hastigheterna $y_{o_l,k}^v$ och $y_{o_r,k}^v$ mäts däremot med odometrar och dessa mätvärden kan då modelleras som okorrelerade med övriga. Kovariansmatrisen för mätvärdena enligt ovanstående mätmodell skulle alltså kunna skrivas på formen

$$R_k = \begin{bmatrix} R_k^{xx} & R_k^{xy} & 0 & 0 & 0 & 0 \\ R_k^{yx} & R_k^{yy} & 0 & 0 & 0 & 0 \\ 0 & 0 & R_k^{aa} & 0 & 0 & 0 \\ 0 & 0 & 0 & R_k^{\omega\omega} & 0 & 0 \\ 0 & 0 & 0 & 0 & R_k^{rr} & 0 \\ 0 & 0 & 0 & 0 & 0 & R_k^{ll} \end{bmatrix}. \quad (20)$$

Om innovationen ϵ är för stor i något element, till exempel om GPS försvinner, tas denna mätning bort ur filtreringen.

5.5.3 Alternativ till fordons- och mätmodell

Om antalet tillstånd ger en för beräkningstung modell kan accelerationen a respektive vinkelhastigheten ω i ψ -led väljas som styrsignaler från IMU istället för att skattas som tillstånd. Nu väljs dessa som tillstånd för att lättare kunna skapa en uppfattning om processbruset. Denna förändring skulle ge att varianserna Q_k^{xx} och Q_k^{yy} kommer att anses vara större än varianserna Q_k^{vv} och $Q_k^{\psi\psi}$. Detta kan motiveras av att x_{k+1} och y_{k+1} beror av två insignaler vardera, medan v_{k+1} och ψ_{k+1} beror bara av en var. Insignalerna kommer att vara brusiga och det är delvis detta som ska kompenseras för i det modellerade processbruset.

Detta skulle även innebära att kovariansen för mätningarna R_k skulle minskas till en 4×4 -matris.



5.5.4 Mindetekteringsmodell

Mindetekteringsmodellen är en utökning av tillståndsmodellen ovan där nya tillstånd tillkommer då en ny mina detekteras. I förväg kommer en radie r_m mätas upp för vilket avståndet bandvagnen måste befinna sig på från en mina för att kunna detektera den. I avsnittet om mindetekteringssystemet beskrivs två fall, ett där sensorn säger till om bandvagnen är nära en mina eller inte och ett fall då bandvagnen kan mäta hur nära den är en mina. I det senare fallet måste det också, precis som i det första fallet, finnas ett tröskelvärde för hur stort utslag sensorn ska ge för att det ska tolkas som att en mina är i närheten. Detta tröskelvärde kommer dock inte vara att sensorn är mättad vilket gäller i det första fallet.

Då detta tröskelvärde överstigs kommer bandvagnen att anta att den är nära en mina varpå en av de två följande sakerna sker:

Bandvagnens skattade position (\hat{x}_k, \hat{y}_k) ligger inom radien r_m från en redan detekterad mina i med skattad position $(\hat{m}_{x,k}^i, \hat{m}_{y,k}^i)$. Då uppdateras denna minas position med hjälp av EKF tillsammans med de andra tillstånden.

Annars, om bandvagnens skattade position ej är inom radien r_m från en detekterad mina, så kommer två nya tillstånd $m_{x,k}^j$ och $m_{y,k}^j$ läggas till i tillståndsmodellen. Dessa har nuvarande positionsskattning $\hat{m}_{x,k}^j = \hat{x}_k$ respektive $\hat{m}_{y,k}^j = \hat{y}_k$ som skattade värden och varianser som avspeglas av positionsskattningens skattade varians P_k^{pos} och radien r_m som minan kan befinna sig på relativt bandvagnens position. I det fallet där sensorn ej är mättad vid mindetektering kommer variansen att kunna vara aningen lägre och variera som en funktion av sensorns utslag istället för att vara en konstant.

Då n minor har hittats (med minst avståndet $2r_m$ ifrån varandra) kommer den modifierade tillståndsvektorn från fordonsmodellen se ut enligt

$$\mathbf{x}_k = \begin{pmatrix} x_k \\ y_k \\ v_k \\ a_k \\ \psi_k \\ \omega_k \\ m_{x,k}^1 \\ m_{y,k}^1 \\ \vdots \\ m_{x,k}^n \\ m_{y,k}^n \end{pmatrix}, \quad (21)$$

där $m_{x,k}^i$ och $m_{y,k}^i$ är x - respektive y -koordinater för minorna i det lokala koordinatsystemet.

Minorna antas, av uppenbara skäl, vara stationära och den utökade rörelsemodellen blir

därmed

$$\mathbf{x}_{k+1} = \begin{pmatrix} x_k + Tv_k \cos\left(\psi_k + \frac{\omega_k T}{2}\right) + \frac{T^2}{2} a_k \cos\left(\psi_k + \frac{\omega_k T}{2}\right) \\ y_k + Tv_k \sin\left(\psi_k + \frac{\omega_k T}{2}\right) + \frac{T^2}{2} a_k \sin\left(\psi_k + \frac{\omega_k T}{2}\right) \\ v_k + Ta_k \\ a_k \\ \psi_k + T\omega_k \\ \omega_k \\ m_{x,k}^1 \\ m_{y,k}^1 \\ \vdots \\ m_{x,k}^n \\ m_{y,k}^n \end{pmatrix} + \mathbf{w}_k. \quad (22)$$

Då en redan upptäckt mina, nummer i säg, passeras tillräckligt nära igen kommer den- nas positionsskattning och variansskattning uppdateras. För det fall då mindetekteringen ej har en mättad signal kommer variansen att kunna modelleras som en funktion av sensors utslag istället för som en konstant och därmed kan bättre positionsskattning för minorna erhållas. Minors skattningar uppdateras genom att modifiera mätmodellen så att positionsmätningarna mäter minpositionernas tillstånd också. Det som skiljer dem från positionernas skattningar i EKF kommer att vara en annan varians i Q_k . Den modifierade mätmodellen ser ut enligt

$$\mathbf{y}_k = \begin{pmatrix} y_k^x \\ y_k^y \\ y_k^a \\ y_k^\omega \\ y_{O_r,k}^v \\ y_{O_l,k}^v \\ 0 \\ 0 \end{pmatrix} = h(\mathbf{x}_k) = \begin{pmatrix} x_k \\ y_k \\ a_k \\ \omega_k \\ \frac{2v_k + d_v \omega_k}{2} \\ \frac{2v_k - d_v \omega_k}{2} \\ x_k - m_{x,k}^i \\ y_k - m_{y,k}^i \end{pmatrix}. \quad (23)$$

Vilket även innebär en utökning av mätkovariansen R_k . Elementen som tillkommer på grund av mindetekteringen kommer vara av ungefär samma storlek som det avstånd vi kan känna av minor.

Denna metod för att skatta minors position kommer endast att fungera då minorna inte ligger för tätt, närmare bestämt inte närmare varandra än $2r_m$ som nämnts ovan. Detta r_m kommer att, till stor del, bero av magneternas styrka. Då minorna ligger ”för tätt” kommer det bli konflikter i vilken minas position som ska uppdateras av EKF. Om detta inträffar kommer det att antas att det är en gemensam mina.

Mätningarna för minornas position kommer även innehålla en osäkerhet då bandvagnen troligtvis inte är rakt ovanför minan vid mätpunkten. Antag att mätningen är likformigt distribuerad på $-r$ till r där r är det avstånd som magnetometern kan registrera en mina. Denna fördelning antas i x - respektive y -riktning. Detta avstånd approximeras sedan vara lika med två standardavvikelse för en normalfördelning med väntevärde 0. Detta ger alltså varians för mätningarna enligt $r^2/4$.

5.6 Underdelssystem: Positionering och kartering

Delsystemet positionering och kartering har som uppgift att sensorfusionera de data som kommer från sensordelsystemet och med dessa bestämma bandvagnens position samt en

karta där hinder och dylikt finns med.

5.6.1 Extended Kalman Filter (EKF)

För att skatta bandvagnens tillstånd används ett utökat Kalmanfilter. EKF är en algoritm som använder sig av bandvagnens fordonsmodell och, i detta fall, mätvärden från GPS, odometri, accelerometrar och gyron för att skatta de olika tillstånden. För mer detaljer om EKF, se kapitel 8 i [6].

För att använda EKF behövs en tillståndsmodell och mätmodell enligt

$$\begin{aligned}\mathbf{x}_{k+1} &= f(\mathbf{x}_k) + \mathbf{w}_k, \\ \mathbf{y}_k &= h(\mathbf{x}_k) + \mathbf{e}_k.\end{aligned}\tag{24}$$

Funktionerna f och h är potentiellt olinjära funktioner som beskriver tillståndsmodellen respektive mätmodellen som används i EKF. Processbruset \mathbf{w}_k och mätbruset \mathbf{e}_k modelleras som gaussiska och additiva i de båda modellerna, därav att de inte ingår i f respektive h .

I EKF skattas de nästkommande tillstånden genom att först göra en mätuppdatering och sedan en tidsuppdatering. I mätuppdateringen skattas de nuvarande tillstånden $\hat{\mathbf{x}}_{k|k}$ och kovariansmatrisen $P_{k|k}$ givet nya mätvärden \mathbf{y}_k och de gamla skattade tillstånden $\hat{\mathbf{x}}_{k|k-1}$. Mätuppdateringarna blir således

$$\begin{aligned}S_k &= H_{k|k-1}P_{k|k-1}H_{k|k-1}^T + R_k, \\ K_k &= P_{k|k-1}H_{k|k-1}S_k^{-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})), \\ P_{k|k} &= P_{k|k-1} - K_kH_{k|k-1}P_{k|k-1},\end{aligned}\tag{25}$$

där S_k är innovationskovariansmatrisen, R_k kovariansmatrisen för mätbruset \mathbf{e}_k , $H_{k|k-1}$ jacobianen för h med avseende på $\hat{\mathbf{x}}_{k|k-1}$ och K_k kalmanförstärkningen. Jacobianen är definierad som

$$H_{k|k-1} = \left[\begin{array}{ccc} \frac{\partial h_1}{\partial x^1} & \dots & \frac{\partial h_1}{\partial x^n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial x^1} & \dots & \frac{\partial h_m}{\partial x^n} \end{array} \right]_{\hat{\mathbf{x}}_{k|k-1}}.\tag{26}$$

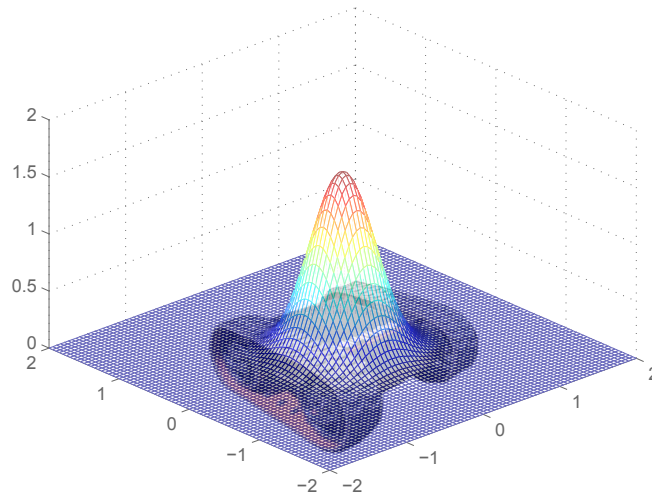
Efter mätuppdateringen sker en tidsuppdatering, där nästa tillstånd, $\hat{\mathbf{x}}_{k+1|k}$, samt kovariansmatris $P_{k+1|k}$ skattas. Detta görs enligt

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k} &= f(\hat{\mathbf{x}}_{k|k}), \\ P_{k+1|k} &= F_{k|k}P_{k|k}F_{k|k}^T + Q_k,\end{aligned}\tag{27}$$

där $F_{k|k}$ är jacobianen för funktionen f med avseende på $\hat{\mathbf{x}}_{k|k}$ och Q_k är kovariansmatrisen för processbruset \mathbf{w}_k .

5.6.2 Kartering

Med EKF-algoritmen som är beskriven ovan fås en skattning av tillstånden, \hat{x}_k , och en tillhörande skattning av dess kovariansmatris, P_k . Då bandvagnen åker genom området,



Figur 12: Tvådimensionell gaussfördelning

på det sätt som specificeras i avsnittet om ruttplanering, skattas bland annat dess position i den lokala kartans (x, y) -koordinater. Kovariansmatrisen P_k^{pos} för de skattade positionstillstånden blir enligt fordonsmodellen det övre vänstra hörnet i P_k

$$P_k = \begin{bmatrix} P_k^{xx} & P_k^{xy} & \cdots \\ P_k^{yx} & P_k^{yy} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} P_k^{pos} & \cdots \\ \vdots & \ddots \end{bmatrix} \quad (28)$$

där P_k^{pos} alltså är en 2×2 -matris. Alla tillstånd x_k^i antas vara Gaussiskt fördelade med skattat väntevärde \hat{x}_k^i . Sannolikhetsfördelningen för positionen, \mathbf{x}^{pos} , i en viss tidpunkt beskrivs då av en tvådimensionell Gaussisk sannolikhetsfördelningsfunktion

$$f_{X,Y}(\mathbf{x}^{pos}) = \frac{1}{2\pi} |P^{pos}|^{-\frac{1}{2}} e^{-\frac{1}{2}((\mathbf{x}^{pos} - \hat{\mathbf{x}}^{pos})^T P^{pos}^{-1} (\mathbf{x}^{pos} - \hat{\mathbf{x}}^{pos}))}, \quad (29)$$

där $\hat{\mathbf{x}}^{pos} = (\hat{x}, \hat{y})^T$ alltså är det skattade tillstånden för x respektive y och $\mathbf{x}^{pos} = (x, y)^T$. En illustration av denna funktion (för något värde $\hat{\mathbf{x}}^{pos}$ och P^{pos}) visas i figur 12.

Underdelssystemet för positionering och kartering ska med hjälp av EKF-skattningen av tillstånden och den antagna Gaussiska fördelningen för positionsskattningen enligt ovan markera sannolikheter för att rutorna i rutkoordinatsystemet är besökta. Hur detta görs beskrivs i nästa avsnitt.

Hinderdetekteringen har en lösning som bygger på samma metod och nästa avsnitt är därför döpt till "Kartuppdatering" då det gäller för flera systems kartor.

5.6.3 Kartuppdatering

Detta avsnitt handlar om uppdatering av sannolikhetskartor givet en viss sannolikhetsfördelning. Främst handlar det om bandvagnens position och denna del är skriven för bandvagnen men hinderkartan använder samma metod. Därför när det står "position" så kan detta bytas ut mot "hindrets position" och när det står "besökt en ruta" så kan



detta bytas ut mot ”detekterat ett hinder i en ruta”, etcetera. Sannolikhetsfördelningen för hindrens positioner kan läsas om i deras respektive avsnitt i detta dokument.

Bandvagnens (hindrets) position för varje rutkoordinat ska bestämmas med en sannolikhet som läggs i en sannolikhetskarta. Sannolikheten att man befinner sig i en viss ruta, (i, j) , vid ett visst tillfälle, k , betecknas med $p_k(i, j)$ och beräknas genom att integrera funktionen i ekvation (29) över denna ruta med en lämplig metod. Om värdet av funktionen $f_{X,Y}(\mathbf{x}^{pos})$ i en viss ruta understiger ett tröskelvärde f_{min} så betraktas sannolikheten att man varit i den rutan vid detta tillfälle som noll. Det vill säga, den betraktas som oberörd av den nuvarande positionsskattningen. Sannolikheten att man någon gång besökt en ruta (i, j) fram till tidpunkt N betecknas med $p_{1:N}(i, j)$. Denna beräknas enligt

$$p_{1:N}(i, j) = 1 - \prod_{n=1}^N (1 - p_n(i, j)). \quad (30)$$

Termerna $(1 - p_n(i, j))$ i produkten är sannolikheterna för att bandvagnen inte är i den berörda rutan vid ett visst tillfälle n . Då dessa multipliceras minskar därmed sannolikheten att bandvagnen inte besökt den aktuella rutan (i, j) fram till och med sampel N . Sannolikheten för att bandvagnen besökt denna ruta är då den komplementära sannolikheten, vilket kan ses i ekvationen. Då bandvagnen åker kommer denna sannolikhet att uppdateras per ruta och per sampel. Denna uppdatering kommer för en ruta, (i, j) , i sampel nummer k vara

$$p_{1:k}(i, j) = 1 - (1 - p_k(i, j))(1 - p_{1:k-1}(i, j)), \quad (31)$$

där $p_k(i, j)$ alltså är sannolikheten för att bandvagnen befinner sig i (i, j) i det nuvarande samplet k och $p_{1:k-1}(i, j)$ var sannolikheten att bandvagnen hade besökt denna ruta någon gång före sampel k .

Då en ruta betraktas som inte berörd i ett sampel k , det vill säga då $f_{X,Y}(\mathbf{x}^{pos}) < f_{min}$ för något värde på $(x, y) \in (i, j)$ så kommer sannolikheten för att denna ruta besöks i detta sampel sättas till noll enligt ovan. Detta betyder i ekvationerna (30) och (31) att $(1 - p_n(i, j)) = 1$ respektive $(1 - p_k(i, j)) = 1$ och innebär helt enkelt att det är sant att bandvagnen inte är i denna ruta under detta sampel.

Ett problem med denna metod är att den förutsätter oberoende händelser, det vill säga att $p_k(i, j)$ och $p_{k-1}(i, j)$ inte beror av varandra, vilket inte stämmer i detta fall. Om två, på varandra följande, skattningar görs då bandvagnen står still så kommer dessa skattningar, $\hat{\mathbf{x}}_k$ respektive $\hat{\mathbf{x}}_{k-1}$, att vara liknande och sannolikheten att man varit på berörda rutor kommer att öka utan att något nytt har hänt. Två tänkta åtgärder finns på detta problem.

Den första åtgärden är att begränsa de rutor man betraktar, inte bara med ett f_{min} som nämns ovan utan även med ett (x, y) -beroende. Med andra ord så ska de rutor som befinner sig för långt bort från väntevärdet (\hat{x}, \hat{y}) av den skattade positionen inte uppdateras.

Den andra åtgärden är att man inte uppdaterar sannolikhetskartan varje gång man uppdaterar tillståndsvektorn. Uppdateringsfrekvensen kan istället vara en funktion av ändringen i tillstånden.

En av ovanstående åtgärder eller någon kombination av dessa kan implementeras för att, åtminstone i någon grad, motverka att sannolikhetskartan uppdateras utan att någon ny information har tillkommit.



5.7 Underdelsystem: Ruttplanering

Delsystemet för ruttplanering ansvarar för att bestämma vägpunkter som bandvagnen ska färdas mellan på sin väg till ett givet mål. Bandvagnens aktuella position och dess målruta knyts således samman av vägpunkter. Mellan två vägpunkter ska bandvagnen kunna färdas rakt.

5.7.1 Ruttplaneringskarta

Nedan föreslås tre alternativ för ruttplaneringskartan. Det första alternativet är en binär karta där varje ruta endast har en egenskap - antingen utforskat område, avsökt område, hinder eller ej farbart område. I en utökad version av ruttplaneringen kan även mina vara en egenskap, på detta sätt kan en besökt mina granskas noggrannare senare. Det andra alternativet innebär en samplad och fusionerad version av sannolikhetskartorna som tilldelar varje ruta endast en egenskap. Det tredje alternativet är att endast sampla sannolikhetskartorna och att ha en mapp med kartor som nyttjas.

Alternativ 1 - binär karta

Varje ruta innehåller endast en unik egenskap. Egenskaperna är utforskat område, avsökt område, hinder och ej farbart område. En fördel med denna karta är att den är väldigt enkel att tänka sig och implementera. En nackdel är att trösklarna för om ett område är exempelvis avsökt behöver sättas lägre än i de andra kartorna för att inte riskera att hamna i en oändlighetsloop. Detta skulle dock kunna lösas genom att bandvagnen endast har en begränsad tid på sig att utforska operationsområdet och när denna tid har löpt ut ska den återvända till basen.

Området som ska utforskas representeras i ruttplaneringen som matrisen M enligt

$$M_{i,j} = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,j} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i,1} & m_{i,2} & \cdots & m_{i,j} \end{bmatrix},$$

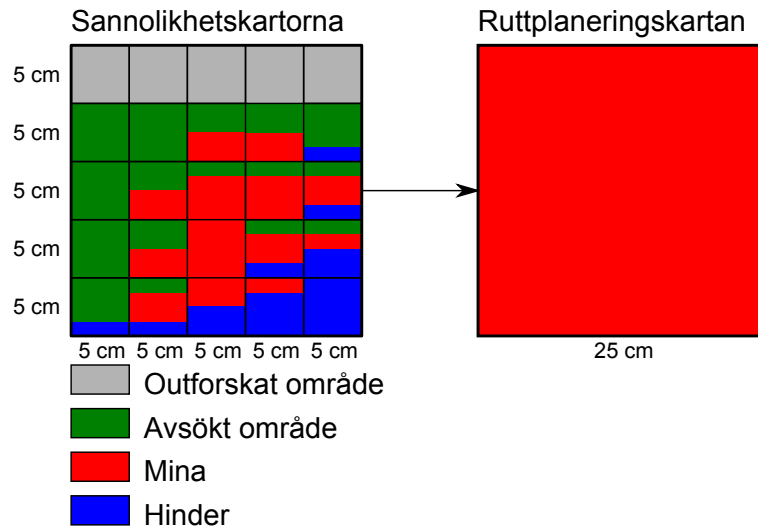
Varje element i matrisen kommer initialt att tilldelas värdet 0 som anger en låg säkerhet gällande att motsvarande ruta på operationsområdet har blivit avsökt. När en ruta med säkerhet har blivit avsökt tilldelas motsvarande element i matrisen värdet 1. Initialt ser således matrisen ut som

$$M_{4,4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

för ett område som är en kvadratmeter stort.

Minröjningsbandvagnen startar på element $m_{1,1}$ och ruttplaneringssystemet markerar element $m_{1,4}$ som nästa mål. När detta exekverats kan matrisen tänkas ha följande utseende beroende på värden från lokalisering och kartläggning som anger hur väl rutten följts

$$M_{4,4} = \begin{bmatrix} 0 & 0 & 0.9 & 0.8 \\ 0 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$



Figur 13: Kartkonvertering från sannolikhetskartorna till ruttplaneringskartan. Ovan visas en utökad version där även minorna finns utplacerade på en sannolikhetskarta.

Ju mer tiden går från senast kända position desto osäkrare bör positionsbestämningen bli. Detta innebär att närliggande rutor till de som är tänkta att utforskas kan ha besökts. Detta ger att deras osäkerhet minskar. Detta behandlar dock positionering och kartering.

Alternativ 2 - samplad och fusionerad karta

Ruttplaneringskartan är en samplad och fusionerad version av avsöchtsannolikhetskartan och hindersannolikhetskartan. Ruttplaneringskartan innehåller diskreta rutor á 0.25×0.25 meter. Dessa rutor placeras ut i det kartesiska kartkoordinatsystemet som representeras av en matris. Matrisen kommer typiskt att ha storleken 200×200 element då det område som ska utforskas är 50×50 meter. Eftersom varje ruta i ruttplaneringskartan är 25 gånger så stor som rutorna i sannolikhetskartorna kommer den egenskap som dominerar de 25 rutorna i sannolikhetskartorna att tilldelas motsvarande ruta i ruttplaneringskartan, se figur 13. Egenskapen att det finns ett hinder på området viktas högre än att området är avsökt. Detta eftersom ett område som klassats med egenskapen hinder även är ett avsökt område. I en utökad version där ruttplaneringen även för protokoll över minornas position viktas även dessa högre. Om den sammanvägda sannolikheten för att området är avsökt i de 25 rutorna i sannolikhetskartan är större än sannolikheterna för hinder och minor så kommer motsvarande ruta i ruttplaneringskartan tilldelas egenskapen avsökt område. Den sammanvägda sannolikheten sparas också till rutan i ruttplaneringskartan. Se figur 13 som visar samplingen av sannolikhetskartorna till ruttplaneringskartan.

Alternativ 3 - mapp med samplade kartor

Sannolikhetskartorna samplas endast enligt beskrivningen ovan och används simultant i ruttplaneringen. Dessutom läggs ytterligare två kartor till med egenskaperna ej farbart område och antal besök på området. Ruttplaneringskartan kommer alltså att innehålla en mapp med fyra kartor som beskriver olika egenskaper. Egenskaperna ges av



$$\begin{bmatrix} p(\text{Avsökt område}) \\ p(\text{Hinder}) \\ \text{Ej farbart område } [0/1] \\ \text{antal besök } [n] \end{bmatrix},$$

och deras kartor betecknas enligt

- M_1 = samplad avsöktsannolikhetskarta
- M_2 = samplad hindersannolikhetskarta
- M_3 = ej farbart område
- M_4 = antal besök

Mappen med kartor betecknas M_{mapp} och ges av

$$M_{mapp} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix}.$$

Alternativ 3 - mapp med samplade kartor kommer att implementeras först eftersom den är mest generell och behåller mest information från sannolikhetskartorna. Alternativ 2 - samplad och fusionerad karta kan dock vara enkel att bygga ut genom att lägga till fler kartor som viktas in. Den är också mer intuitiv eftersom varje ruta bestäms ha en egenskap - inte flera. Alternativ 1 - binär karta är den enklaste att föreställa sig och skulle kunna vara ett annat bra alternativ att starta med. Den binära kartan lider dock av att vi kan tilldela en ruta som avsökt trots att vi inte varit där. I de andra två kartorna säger vi det endast med en viss säkerhet.

5.7.2 Avsökningsalgoritmer

Nedan ges tre förslag på avsökningsalgoritmer. Den första ger en spiralformad avsökning, den andra en fram- och tillbakagående avsökning och den tredje nyttjar en förutbestämd rutt.

Alternativ 1 - spiralformad avsökning

Ruttplaneringen kommer att söka efter en utforskad ruta som i första hand ligger på samma rad eller i samma kolumn och så långt ifrån nuvarande position som möjligt. Detta tillvägagångssätt ger en spiralformad väg in till centrum av operationsområdet, då inga hinder föreligger. På detta sätt minimeras antalet svängar i tät följd.

Alternativ 2 - fram- och tillbakagående avsökning

En annan avsökningsstrategi är att söka av första raden, sedan andra raden och så vidare. Denna strategi ger två svängar i tät följd men sedan en konstant vägsträcka då inga hinder föreligger.



Alternativ 3 - förutbestämd rutt

Ett tredje alternativ att söka av området är att beräkna en rutt, som täcker in hela operationsområdet, innan bandvagnen lämnar basrutan. Hinder behandlas när de dyker upp genom att följa deras rand och sikta på att ta sig tillbaka till den förutbestämda ruttan så snart som möjligt. Ju längre bandvagnen kommer ifrån den förutbestämda ruttan desto mer straffas den. Denna avsökningsalgoritm kan liknas vid en linjeföljning.

Figur 14 visas hur bandvagnen använder sig av en spiralformad avsökning av operationsområdet.

Utvärdering av minröjningsbandvagnens kvaliteter

Efter fälttest kan minröjningsbandvagnens kvaliteter beträffande 90°-svängar kontra att köra rakt, bestämmas. Därefter kan en avsökningsstrategi väljas som innehåller flest moment som minröjningsbandvagnen klarar av bäst.

Till att börja med kommer alternativ 1, spiralformad avsökning, implementeras. Visar det sig att denna avsökningsstrategi ej fungerar tillfredsställande kommer alternativ 3, förutbestämd rutt, att implementeras.

5.7.3 Markering av objekt på ruttplaneringens interna karta

Hinder markeras med en brädgård, #, på ruttplaneringens interna karta, här finns en osäkerhetssiffra kvar som talar om hur säkra vi är på att hindret befinner sig just på den rutan. Områden som ej kan avsökas markeras med ett kryss, X. När minröjningsbandvagnen kommit fram till nästa vägpunkt kan kartan se ut som

$$M_{4,4} = \begin{bmatrix} 1 & 1 & 1 & 0.9 \\ 0 & 0 & 0.1 & 0.3 \\ 0 & 0 & 0.1 & 0.3 \\ 0 & 0 & 0.1 & 0.1 \end{bmatrix}.$$

I slutet av avsökningen av området kan kartan se ut som

$$M_{4,4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0.9 \\ \#1 & \#0.9 & \#0.9 & 0.9 \\ X0 & X0 & \#0.9 & 0.9 \end{bmatrix}.$$

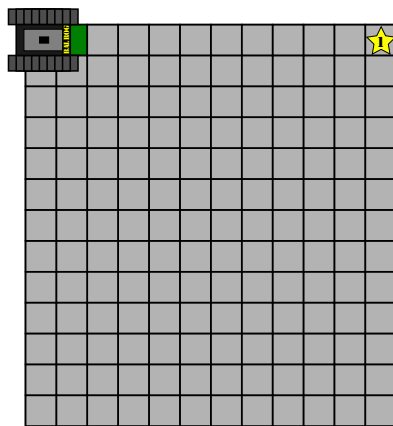
5.7.4 Hantering av hinder

Om ett hinder stöts på ska en ny rutt till målrumen genereras i första hand. I andra hand ska aktuell målruta utrangeras och en ny målruta väljas. I tredje hand ska hindret utforskas så att man vet var det finns på kartan och därmed kan undvika att åka in i det i ett senare skede.

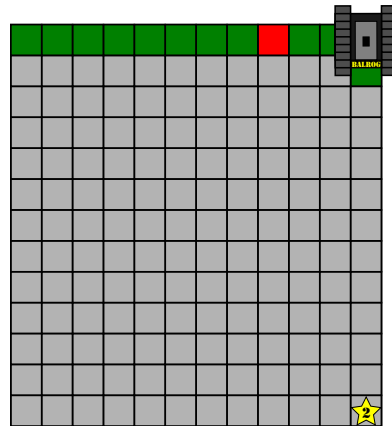
Alternativ 1 - ny rutt till målrumen

När ett hinder påträffas på området framför bandvagnen genereras en ny rutt till målrumen. Förhoppningsvis fungerar denna rutt bättre. Om inte så kommer endast en ny rutt att genereras till målrumen tills målrumen är nådd eller utrangeras på grund av att den ligger på ett ej farbart område.

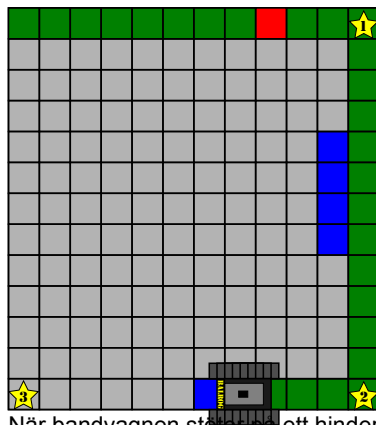
Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Ollesson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



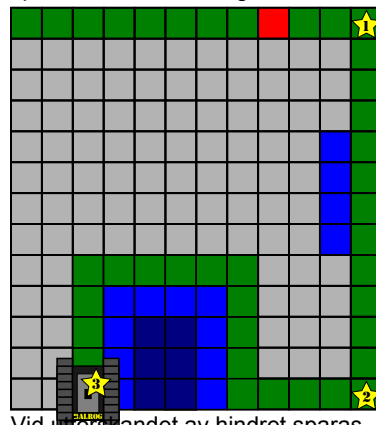
Ruttplaneringen sätter ut en målruta som bandvagnen Balrogg kan åka rakt till.



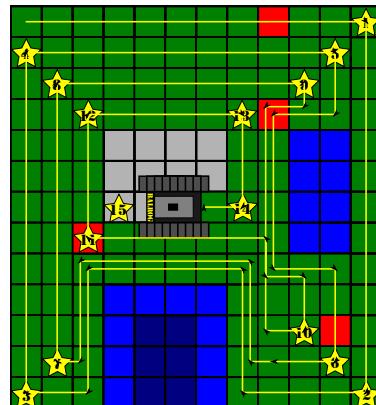
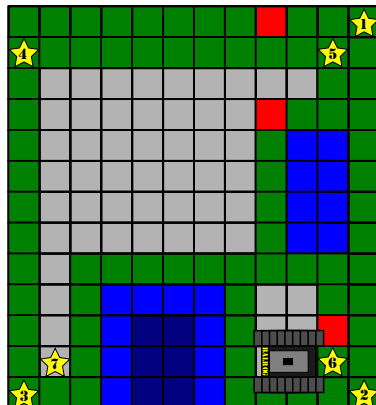
Nästa målruta sätts ut i nästa hörn av kartan, detta kommer att ge en spiralformad avsökning av området.



När bandvagnen stöter på ett hinder utforskar den det i positivt led i första hand och sedan i negativ led.



Vid utforskandet av hindret sparas målrutan undan. När hindret utforskats beräknas en ny rutt till målrutan.



- Utforskat område
- Avsökt område
- Mina Rutmål
- Hinder Ej farbart område

A*-algoritmen används för att hitta kortaste vägen mellan två rutor och används när operationsområdet är delvis utforskat. När alla rutor är avsökta planeras en rutt tillbaka till basrutan.

Figur 14: Ovan visas en spiralformad avsökning av operationsområdet.



Alternativ 2 - välj ny målruta

Det andra alternativet att hantera hinder är att utrangera aktuell målruta och rutten dit och istället generera en ny målruta med tillhörande rutt dit.

Alternativ 3 - utforska hindret

När hinderdetekteringsystemet signalerar att ett hinder finns i omgivningen av bandvagnen så kommer kartan att uppdateras. Om hindret finns i bandvagnens beräknade rutt, det vill säga att en kollision med hindret är oundvikligt om bandvagnen fortsätter efter den planerade rutten, så ska systemet gå in i ett "utforska hinder"-läge.

När ruttplaneringsdelsystemet går in i "utforska hinder"-läget börjar delsystemet med att utrangera de vägpunkter som finns i den länkade listan för vägpunkter. Samtidigt sparas den sista vägpunkten, målrutan, undan för att ruttplaneringsdelsystemet senare ska kunna beräkna en ny rutt till målrutan. Därefter utforskas hindret genom att, om möjligt, följa det i positiv led runt hindret tills bandvagnen är tillbaka till startrutan för utforskandet av hindret. Om hindret sträcker sig utanför operationsområdet ska bandvagnen utforska så mycket som möjligt av hindret tills den är vid gränsen av operationsområdet. Därefter ska bandvagnen utforska hindret åt andra hållet, det vill säga utforska hindret i negativ led så långt som möjligt. Om bandvagnen hamnar vid gränsen av operationsområdet även vid utforskandet av hindret i negativ led så ska ruttplaneringsdelsystemet planera en ny rutt till den undansparade målrutan. En ny rutt till den undansparade målrutan ska även planeras då bandvagnen utforskar hindret i positiv led och återkommer till startrutan för utforskandet av hindret.

För att följa ett hinder skickas ett kommando till regleringen om att reglera rutten efter hindret istället för efter vägpunkter.

Alternativ 4 - följ hindret

Detta alternativ hör till den tredje avsökningsalgoritmen där bandvagnen ska följa en förutbestämd rutt. Stöter bandvagnen på ett hinder ska den följa hindret i positiv led och ju längre bort ifrån den förutbestämda rutten som den kommer desto mer ska den straffas. Att hamna utanför operationsområdet straffas hårt vilket gör att bandvagnen håller sig inom operationsområdet.

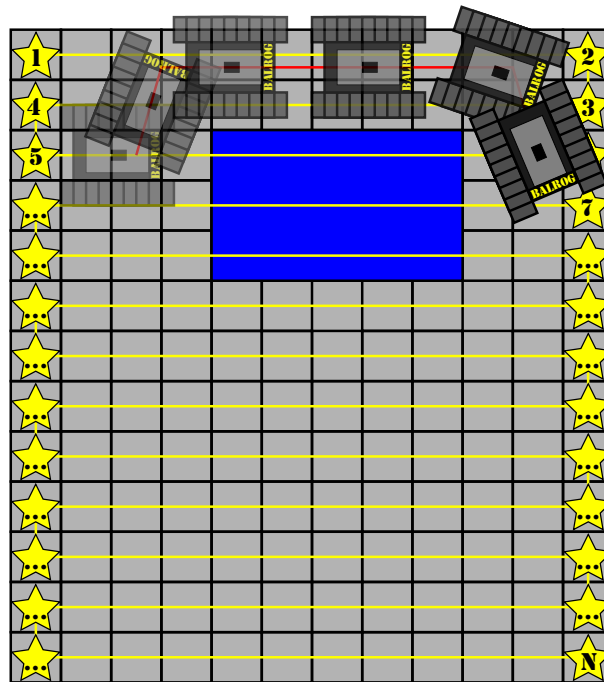
5.7.5 Område utan GPS-täckning

Ett område med GPS-täckning definieras som ett område där variansen understiger tröskelvärdena 1 meter i position eller 45° i vinkel. Nedan ges tre alternativ för att hantera situationer utan GPS-täckning. Det första alternativet handlar om att välja en ny rutt till målrutan och därmed besöka området senare då GPS-täckning kanske finns. Det andra alternativet är att välja en ny målruta. Det tredje alternativet förutsätter ett statistiskt och väldefinierat område utan GPS och blir därför mindre användbart eftersom området utan GPS-täckning troligtvis kommer att fluktuera.

Alternativ 1 - ny rutt till målrutan

Till att börja med sparas aktuell målruta undan och vägpunkterna utrangeras. En ny rutt till målrutan genereras och nya vägpunkter skickas till regleringen.

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Olsson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



Figur 15: Figuren visar hur bandvagnen hanterar ett hinder när den ska följa en förutbestämd rutt. Den ska följa den gula ruten, när detta ej är möjligt vid hindret reglerar den efter hindret och följer den röda ruten.

Eventuellt behöver bandvagnen börja med att ta sig ut ur aktuellt område innan en ny rutt planeras. Detta kan lösas genom att skicka ett kommando till regleringen som anger att bandvagnen ska backa tills GPS-täckning finns.

Alternativ 2 - välj ny målruta

Det andra alternativet är att direkt välja en ny målruta när variansen för position och vinkel överstiger tröskelvärdena som angavs ovan.

Alternativ 3 - utforska området

Då ett område saknar GPS-täckning ska bandvagnen identifiera detta område för att sedan utforska det från olika håll. När variansen för position och riktning är för stor går bandvagnen in i ett "utforska GPS-avsaknadsområde"-läge. Till att börja med sparas aktuell målruta undan för att användas igen efter att området utan GPS-täckning har identifierats. Sedan återvänder bandvagnen till senast kända position. Bandvagnen åker sedan till närmsta ruta i positiv led och fortsätter åka i samma riktning som den hade när den först tappade GPS-täckning för att på så sätt komma in i området utan GPS-täckning igen. Om den tappar GPS-täckning igen så vänder den om och åker ut igen och fortsätter till nästa rad eller kolonn i matrisen. Detta itereras tills bandvagnen kan åka fyra rutor in i det misstänkta området med bibehållen GPS-täckning. Vid detta tillfälle åker bandvagnen tillbaka till den rad eller kolumn där den senast tappade GPS-mottagning för att sedan utforska området från andra hållet.

I korta drag åker således bandvagnen in och ut ur området utan GPS-täckning för att lokalisera dess rand. När området utan GPS-täckning väl är lokaliserat utforskas det genom



att bandvagnen korsar det fram och tillbaka, från en given startruta med känd position till en målruta som ligger på andra sidan området utan GPS-täckning. Om bandvagnen lyckas ta sig till målrutan så bör den ha åkt över de rutor som förbinder start- och målrutan. Om den däremot hamnar på en annan känd ruta kan en trolig trajektoria beräknas som visar vilka rutor som troligtvis har besökts. Denna utforskning pågår tills alla motstående start- och målrutor kring området har besökts eller om området anses känt med 95 % säkerhet dessförinnan. Positionering- och karteringsdelsystemet uppdaterar sannolikheterna för att ett område har besökts eller ej. Operatören kan själv tänkas få ange med vilken säkerhet ett område ska sökas av.

Om området utan GPS-täckning sträcker sig utanför operationsområdet kan i bästa fall start- och målrutor på motstående sidor om området användas för en del av området. Om området däremot ligger i ett hörn av operationsområdet så kommer målrutorna att utgöras av de rutor som ligger i anslutning till gränsen av operationsområdet. I detta fall kan dödräkning av positionen användas för att veta när en målruta nåtts för att sedan först åka till nästa målruta som ligger i anslutning till bandvagnens aktuella ruta och sedan tillbaka till nästa startruta. En annan strategi är att avsöka området med cirkulära trajektorier med allt större radier. Denna avsökning utgår från en central del av randen till området utan GPS-täckning.

Om ett hinder stöts på i området utan GPS-täckning så utforskas detta först. På så sätt fås formen av hindret och denna form kan sedan fästas på kartan när en del av hindret positionerats tillräckligt väl. ”Utforska hinder”-läget har med andra ord högre prioritet än ”utforska GPS-avsaknadsområde”-läget. Högst prioritet är att befinna sig på operationsområdet.

I första hand kommer alternativ 1 att implementeras.

5.7.6 A^* -algoritmen

A^* -algoritmen är en vägsökningsalgoritm som används för att hitta en effektiv väg mellan två rutor. Se *Artificial Intelligence A Modern Approach*, [13], för mer information.

Den första rutan, startrutan, är den aktuella rutan som bandvagnen befinner sig på och den andra rutan är målrutan dit bandvagnen vill ta sig. A^* använder sig av heuristik i sökningen efter den bästa vägen. Funktionen som används i A^* utgörs av

$$f(x) = g(x) + h(x)$$

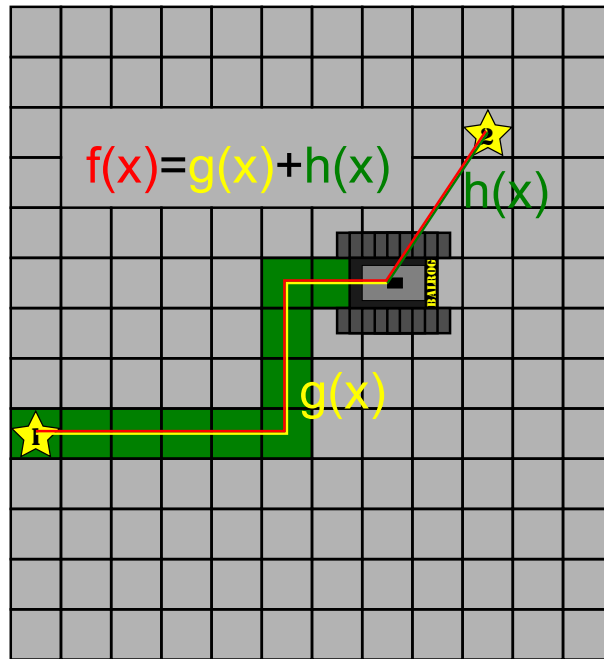
- $f(x)$ är den heuristiska funktionen som anger den nuvarande sträckan mellan startrutan och aktuell ruta plus en heuristisk skattning av sträckan mellan start- och målrutan. Denna funktion används för att välja i vilken ordning noderna (rutorna) i sökträdet ska besökas.
- $g(x)$ är sträckan mellan startrutan till aktuell ruta via den framräknade vägen
- $h(x)$ är en heuristisk underskattad sträcka mellan aktuell ruta och målrutan. Denna sätts lika med den euklidiska sträckan mellan aktuell ruta och målrutan.

En sträcka svarar mot en kostnad och A^* -algoritmen minimerar denna kostnad. Figur 16 visar hur A^* -algoritmen fungerar.

5.7.7 Avslutande av minsökning

Varje ruta har en räknare som anger antalet gånger ruttplaneringssystemet har sökt av den rutan genom att färdas över den. Hela området anses avsökt då alla rutor har besökts med

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinerMarker (MM)	Dokumentansvarig:	Niklas Olleson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



Figur 16: Målet är att ta sig mellan vägpunkt 1 och vägpunkt 2. Den heuristiska funktionen är den gula sträckan mellan startrutan och aktuell ruta plus den kortaste sträckan mellan aktuell ruta och målruta som är den gröna sträckan.

en säkerhet på 95 % (undantaget ej farbara områden) eller om alla rutor med en lägre avsökningssäkerhet besökts minst fem gånger. När hela området är avsökt enligt dessa kriterier ska ruttplaneringsdelsystemet bestämma bästa vägen tillbaka till startpositionen. Detta görs med A^* -algoritmen. Om minröjningsbandvagnen upprepade gånger försöker utforska en specifik ruta utan att sänka dess osäkerhet så markeras denna ruta med ett snabel-a, @.

5.8 Underdelsystem: Reglering

För att reglera bandvagnen används en PD-regulator. Syftet med regleringen är att minimera felet ϵ_ψ mellan önskad färdriktning och sann färdriktning. Bandvagnen kommer köra med en hastighet v som varierar beroende på hur stort reglerfelet ϵ_ψ och avståndet d till målet är.

Om avståndet till målet är större än en meter ställs hastigheten in enligt

$$v_k = \frac{v_{max}}{1.2} \left(1 - \frac{|\epsilon_{\psi,k}|}{\pi} \right)^4, \quad (32)$$

där v_{max} är bandvagnens maxhastighet. Detta innebär att ju större reglerfelet är desto långsammare kommer bandvagnen att köra. Om avståndet till målet är mindre än en meter ställs hastigheten in enligt

$$v_k = K_d d + v_{min}. \quad (33)$$

Det vill säga en enkel P-reglering med parameter K_d . Om reglerfelet överstiger 20° sätts $v = 0$.



Bandhastigheterna ställs in enligt

$$\begin{aligned}v_{l,k} &= v_k - K_\psi \epsilon_{\psi,k} - \frac{T_{d,\psi}}{T_s} (\epsilon_{\psi,k} - \epsilon_{\psi,k-1}), \\v_{r,k} &= v_k + K_\psi \epsilon_{\psi,k} + \frac{T_{d,\psi}}{T_s} (\epsilon_{\psi,k} - \epsilon_{\psi,k-1}),\end{aligned}\tag{34}$$

där K_ψ och $T_{d,\psi}$ är reglerparametrar som bestämmer hur reglerfel ska straffas. Den proportionella förstärkningen K_ψ anger hur mycket reglerfelet ska straffas medan deriveringsförstärkningen $T_{d,\psi}$ anger hur mycket förändringshastigheten mellan nuvarande och föregående reglerfel ska straffas.

5.8.1 Reglera efter vägg

Det här avsnittet beskriver vad som ska ske då delsystemet för ruttplanering meddelar att bandvagnen inte ska reglera till nästa vägpunkt utan istället reglera efter ett hinder.

Syftet är minimera ett fel ϵ_Δ som är skillnaden i avstånd till ett hinder mellan två sensorer på samma sida av bandvagnen. Detta görs genom att använda en PD-regulator på liknande sätt som avsnittet ovan men att använda ϵ_Δ istället för ϵ_ψ

$$\begin{aligned}v_{l,k} &= v_k - K_\Delta \epsilon_{\Delta,k} - \frac{T_{d,\Delta}}{T_s} (\epsilon_{\Delta,k} - \epsilon_{\Delta,k-1}), \\v_{r,k} &= v_k + K_\Delta \epsilon_{\Delta,k} + \frac{T_{d,\Delta}}{T_s} (\epsilon_{\Delta,k} - \epsilon_{\Delta,k-1}),\end{aligned}\tag{35}$$

där hastigheterna v_k sätts enligt ovan med undantag för då avståndet till hindret från en framsensor är mindre än 50 centimeter. Då sätts $v_k = 0$

6 Kommunikation mellan basstation och bandvagn

Basstationen kommunicerar med bandvagnen över WLAN via TCP/IP där data skickas i båda riktningarna. För att kommunicera med bandvagnen, som fungerar som server, ansluter basstationen till bandvagnen genom att ange bandvagnens IP-nummer. En tabell över de meddelanden som hanteras finns i tabell 4

7 Kodstandard

All kod kommer att skrivas i C++ de enda undantagen är koden på ARM-processorn som har skrivits av tidigare projektgrupper i C och simuleringskod som kommer att skrivas i MatLab.

7.1 Stil

Vi kommer att indentera med två mellanslag. Vi kommer aldrig att skriva rader som är längre än 110 tecken. Alla filer kommer vi att spara med teckenkodningen UTF-8. Vi kommer att licensiera vår kod under GNU GPLv3. All kod kommer vi att skriva på engelska.

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinerMarker (MM)	Dokumentansvarig:	Niklas Ollesson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



Id	Meddelandedyp
-1	Halt
0	Heartbeat
1	Manual
2	Automatic
3	LogMessage
4	Map
5	Waypoint
6	WheelData
7	OdometerData
8	IMUData
9	GPSData
10	RangeData
11	StateData
12	ErrorMessage

Tabell 4: Kommunikationsmeddelanden mellan bandvagn och basstation

I början av alla egenskrivna kodfiler kommer vi att skriva.

```

/*****/
/*
/* 888b d888 d8b 888b d888 888 */
/* 8888b d8888 Y8P 8888b d8888 888 */
/* 88888b.d88888 88888b.d88888 888 */
/* 888Y88888P888 888 88888b. d88b. 88888b. 888Y88888P888 88888b. 888d888 888 888 .d88b. 888d888 */
/* 888 Y888P 888 888 888 "88b d8P Y8b 888 "88b 888 Y888P 888 "88b 888P" 888 .88P d8P Y8b 888P" */
/* 888 Y8P 888 888 888 888 88888888 888 888 888 Y8P 888 .d888888 888 888888K 88888888 888 */
/* 888 " 888 888 888 888 Y8b. 888 888 888 " 888 888 888 888 888 "88b Y8b. 888 */
/* 888 888 888 888 888 "Y8888 888 888 888 888 "Y888888 888 888 888 "Y8888 888 */
/*
/*
/* filename.ext is part of MinenMarker.
/*
/* MinenMarker is free software: you can redistribute it and/or modify
/* it under the terms of the GNU General Public License as published by
/* the Free Software Foundation, either version 3 of the License, or
/* (at your option) any later version.
/*
/* MinenMarker is distributed in the hope that it will be useful,
/* but WITHOUT ANY WARRANTY; without even the implied warranty of
/* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
/* GNU General Public License for more details.
/*
/* You should have received a copy of the GNU General Public License
/* along with MinenMarker. If not, see <http://www.gnu.org/licenses/>.
/*
/* Description:
/* One line about what is in the file
/*
/*****/

```

I alla headerfiler kommer vi att skriva en "inclusion guard" på följande format.

```

#ifndef FILENAME_EXT
#define FILENAME_EXT

File contents

#endif /* FILENAME_EXT */

```

Kontrollstrukturer skriver vi med måsvinge på samma rad. Exempel på kontrollstrukturer är if-satser, for-loopar och while-loopar. Funktioner skriver vi med måsvinge på nästa.

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Olleson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



Skillnaden är till för att det ska gå snabbt för den som läser koden att avgöra om det är en funktion eller en kontrollstruktur det handlar om. Vi kommer att använda mellanrum mellan kontrollstrukturer och deras "condition" och mellan deras "condition" och måsvingen.

```
int fun(int a)
{
    statements;
}

if (condition) {
    statements;
} else {
    statements;
}

for (int i = 0; i < iterations; ++i) {
    statements;
}

switch(cond) {
    case 1:
        statements;
        break;
    case 2:
        statements;
        break;
}

while (condiditon) {
    statements;
}
```

När vi skriver uttryck kommer vi att använda oss av mellanrum mellan alla operatörer förutom gånger och delat då dessa operatörer har välkänt högre prioriteringsordning än andra operatörer. Vi använder oss av parenteser för att klargöra operatorordningen för alla operatörer förutom dem som vi är vana vid från matematiken.

```
bool r = (a && b) || c;
int a = a*b + c
```

När vi skriver pekar- eller referensdeklarationer så lämnar vi ett mellanrum på båda sidor om specialtecknet.

```
char * somer_text;
vector & a_vector;
```

Klasser skriver vi med måsvinge på samma rad som deklarationen. Först publicobjekt, sedan protected och sist private. Konstruktörer skriver vi alltid med initialiseringslistor. Endast konstruktörer med tomt kodblock skriver vi i headerfiler annars så skrivs de i sourcefilen. Alla metoder skriver vi i sourcefilen.

```
class A {
public:
```



```
A()
: a(1)
, b(2)
, c(3)
{}

private:
    int a;
    int b;
    int c;
}
```

7.2 Namngivning

Filer som endast innehåller funktioner kommer vi att döpa till

fileName.cpp samt fileName.h

Filer som innehåller en klass kommer vi att döpa till

ClassName.cpp samt ClassName.h

Klasser, metoder och klassvariabler döper vi enligt

```
class ClassName {
public:
    int MethodName();

private:
    int memberVariableName;
}
```

Globala variabler och funktioner döper vi enligt nedan. När vi deklarerar globala variabler kommer vi alltid att deklarerera dessa static så att de bara syns i den kodfil där de används. Vi kommer aldrig att använda globala variabler som sträcker sig över mer än en kodfil (kompileringsenhet).

```
static int varName = 10;

int longDescriptiveFunctionName()
{
    statements;
}
```

7.3 Riktlinjer

Vi kommer inte att använda oss av "namespaces", då vi avgör att även vid vidare utveckling är inte projektet stort nog för att det skall behövas.

Vi kommer inte att använda oss av "exceptions" då detta försvårar mycket av minneshanteringen och det blir svårare att använda externa bibliotek. Se [10] för en diskussion om anledningar.



7.3.1 Klasser

Om en klass inte skall gå att kopiera så kommer vi att deklarerat kopieringskonstruktron och kopieringstilldelningen privata så att de inte går att använda av misstag om den skall gå att kopiera så kommer vi att deklarerat dessa funktioner.

Konstruktorer kommer vi bara att använda till att initialisera medlemsvariabler med. Ytterligare initialisering kommer att göras i en "Init"-funktion.

Konstruktorer som tar en parameter kommer vi att markera som "explicit" för att undvika automatiska typkonverteringar.

7.4 Basstationen

Basstationen kommer att använda sig av mjukvarubiblioteket Qt [7] för att rita det grafiska gränssnittet och nätverkshantering.

7.5 Bandvagnen

Bandvagnen kommer att skrivas till Linux och använda Linux specifika bibliotek. För matematiska operatörer kommer vi att använda oss av biblioteket GSL [11].

7.5.1 Kompilering

Vi kommer att använda GNU GCC för att kompilera. Vi kommer att kompilera vår kod utan varningar med följande flaggor.

```
-Wall -Wextra -Werror -pedantic-error  
-Wno-unused-function -Wno-unused-parameter -Wno-unused-variable  
-Wno-long-long -fno-exceptions -fno-unwind-tables
```

Referenser

- [1] Tomas Svensson, Christian Krysanter, *Projektmodellen LIPs*. version 1.4, 2008.
- [2] Lipsmodellens hemsida, <http://www.liu.se/cul/resurser/lips/word-och-excel-mallar?l=sv>, 2012-09-09
- [3] Designspecifikation Autonom Bandvagn. Version 1.0, ISY Linköpings tekniska högskola, Linköping. http://www.isy.liu.se/edu/projekt/tsrt10/2009/bandvagn/doc/Designspecifikation_v1.0.pdf Norberg, Johan (2009).
- [4] Autonom bandvagn med stereokamera: Kravspecifikation Version 1.0, ISY Linköpings tekniska högskola, Linköping. http://www.isy.liu.se/edu/projekt/tsrt10/2010/bandvagn-2010/doc/kravspecifikation_v1.0.pdf Hanning, Gustav (2010).
- [5] Autonom bandvagn med stereokamera: Designspecifikation Version 1.0, ISY Linköpings tekniska högskola, Linköping. http://www.isy.liu.se/edu/projekt/tsrt10/2010/bandvagn-2010/doc/designspecifikation_v1.0.pdf Hanning, Gustav (2010).

Kursnamn:	Reglerteknisk projektkurs, CDIO	E-post:	minenmarker@googlegroups.com
Projektgrupp:	MinenMarker (MM)	Dokumentansvarig:	Niklas Ollesson
Kurskod:	TSRT10	Ansvarigs e-post:	nikol038@student.liu.se
Projekt:	Minröjningsbandvagn	Dokumentnamn:	designspec-1.0.pdf



- [6] Gustafsson, Fredrik (2010), *Statistical Sensor Fusion*. Edition 1:1, Studentlitteratur AB, Lund, ISBN 978-91-44-054189-6
- [7] The Qt Project, <http://qt-project.org/>, 2012-09-25
- [8] SRF10 Ultrasonic range finder Technical Specification, <http://www.robot-electronics.co.uk/htm/srf10tech.htm>, 2012-09-28
- [9] BL233B Datablad, www.i2cchip.com/pdfs/bl233_b.pdf, 2012-09-28
- [10] The google c++ code standard, <http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml>, 2012-09-25
- [11] The GNU Scientific Library, <http://www.gnu.org/software/gsl/>, 2012-09-25
- [12] Minröjningsbandvagn: Kravspecifikation Version 1.0 ISY Linköpings tekniska högskola, Linköping. <http://www.isy.liu.se/edu/projekt/reglerteknik/2012/minrojningsrobot/wp-content/uploads/2012/10/kravspecifikation.pdf> Ollesson, Niklas (2012)
- [13] Russel, Stuart; Norvig, Peter (2010), *Artificial Intelligence A Modern Approach* Third Edition, Pearson Education Inc, New Jersey, ISBN-13 978-0-13-207148-2