

KAPITTEL 11

Mobile

Roboter



11.1 EN DRUE AV GANGEN

Har du noen gang forestilt deg hva som skjer hvis du gjør et virkelig godt arbeid på bachelor- eller masteroppgaven din?

Det tenkte sikkert ikke Lars Grimstad på da han skrev sin masteroppgave i 2014, som nå er gründer og CTO av Saga Robotics.

Pål Johan From som var veilederen til Lars på både masteren og doktorgraden, så antageligvis noe stort da de hamret sammen den første prototypen Thorvald I; og de to startet opp Saga Robotics i 2016 og lanserte Thorvald II. Pål som CEO og Lars som CTO. Kanskje du og jeg får den samme skjebnen snart?

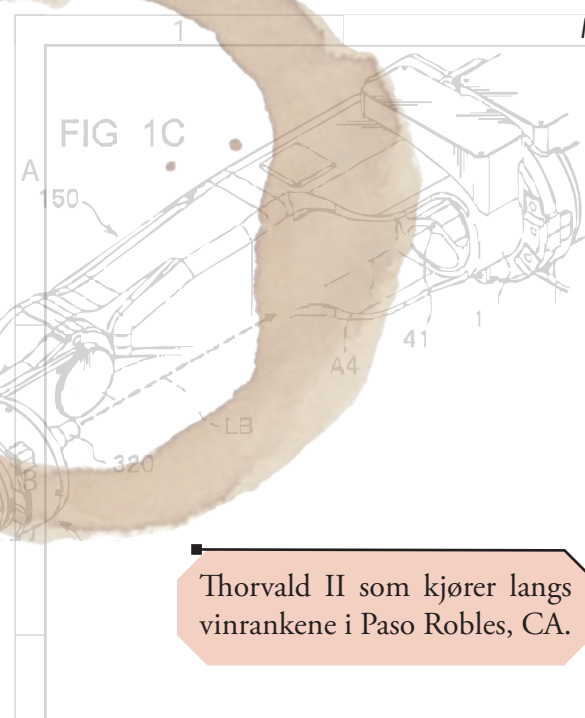
Saga Robotics produserer landbruksroboten Thorvald. Den startet med å plukke jordbær, men plukker også vindruer i California.

Thorvald II som behandler jordbær i Storbritania

Forestill deg nå at du står ved vinrankene i Paso Robles, California. Det er over 500 mål med jorde fylt med druer som du skal undersøke, beskytte mot insekter og sykdom, sortere og plukke.

Det er flere utfordringer rundt landbruksproduksjon. Lite tilgjengelig arbeidskraft, behov for kvalitetssikring og samtidig bærekraftig løsning. Dette med bærekraft er her ikke bare fokus på å ikke bidra til klimaendringer, men også å tilpasse produksjonen til klimaendringene. En siste utfordring er hvordan man kan sikre at de metodene man bruker for å få planten til å vokse, også gjør minimal skade for neste høsting. For eksempel, så vil kjemiske inngrep trekke inn i jorda og potensielt skade plantene for neste år.





Thorvald II som kjører langs vinrankene i Paso Robles, CA.



Sammen med planteforskere fra NMBU har du funnet ut at å lyse på plantene med intenst UV-C-lys tar du knekken på alle bakterier og insekt uten å skade plantene eller druene. Utfordringen er nå å bygge en robot som kan lyse på disse vinrankene.

Hittil har vi snakket om inverskinematikk, og kan regne ut hvordan hjulene skal heves og senkes for å holde roboten i riktig høyde. Du kan også bruke datasyn og kunstig intelligens til å identifisere druer og estimere druenes modenhet via robotens kamera-system.

Men hvordan klarer du å peile inn roboten til å følge kilometer med vinranker uten å kræsje? Du har et kart over landområdet, men det tar ikke høydemeter eller ulent terreng med i betraktningen. Det betyr at du ikke kan bruke kartet til å si "kjør 800 meter sørover for å komme til enden av ranken", fordi det er 800 meter i luftlinje. Teller du omdreininger på hjulene til du kommer til 800 meter, vil du ikke havne der du ønsker.

GPS kan benyttes, og med enkelte teknikker som vi skal se på i dette kapitlet kan vi få millimeter nøyaktighet. Men, GPS har veldig lav oppdateringsfrekvens, som gjør roboten blind i tidspunktene mellom målingen. Ikke minst får vi problemer hvis det kommer en solstorm som kan gjøre målingene unøyaktig i flere timer.

Hvordan vil du få roboten til å navigere seg nøyaktig gjennom vinrankene?



THORVALD

Roboten Thorvald har dype røtter i NMBU. Ikke bare er den utviklet ved universitetet, men navnet Thorvald ringer også en bjelle hos studentene. Thorvald er nemlig deponymet på en mannlig Ås-student, og Tora er deponypet for kvinner. Dette stammer fra UKA-revyen “På tå hev” i 1924, hvor sangen “Jeg og n’Thorvald og Johannes”, så ble da og er nå nasjonalsangen til NMBU.

Thorvald I ble først påbegynt i 2014 på NMBU. Den hadde fire individuelle hjul og en god del verktøy som kunne installeres på midten. Roboten ble brukt til mange scenarier for å finne ut hva den kommersielle roboten skulle gjøre.

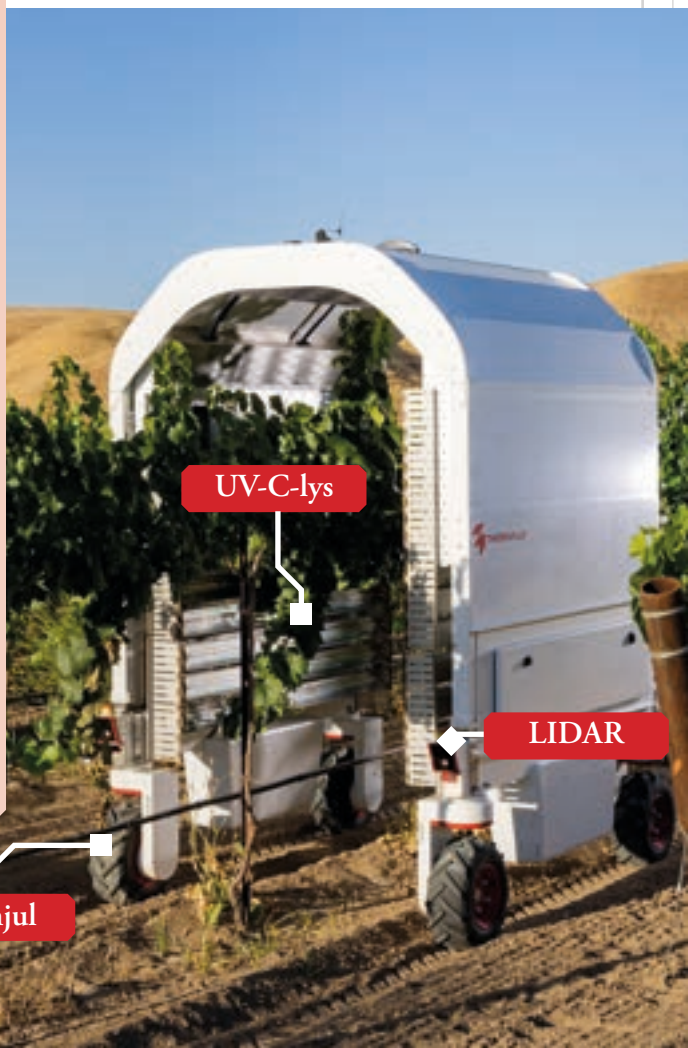
Thorvald II kom i 2016. Den har et høyt skrog så den kan kjøre over planter. Den har en modulær utforming slik at man kan tilpasse den til en rekke typer landbruksområder. Den kan klippe, plukke, så, og ikke minst lyse UV-C-lys på plantene for å hindre mugg. Hver modul har et enkelt grensesnitt som trekkes gjennom hele skroget slik at den kan konfigureres til de fleste situasjoner.

Thorvald III ble lansert i 2024 og er den nyeste iterasjonen av roboten. Den har det samme grensesnittet og samme tanken rundt modularitet, men med en utforming som gjør den enklere å produsere og lettere å vedlikeholde. Både software og firmware er lik på Thorvald II og III. Robotflåten med Thorvald III består nå av over 150 roboter.



Thorvald I

Thorvald II



11.2 HVOR ER JEG?

Vi har gjentatte ganger i boken snakket om viktigheten av fiksturer. En robot er både blind og døv, og kan ikke gjøre jobben sin uten å vite hvor gjenstander er i forhold til den. Vi startet med å se på hvordan vi kan fiksture gjenstander for å flytte de til faste posisjoner, så bruk av sensorer, slik som kamera for å kunne korrigere og være mer fleksibel. Men nå kommer den ultimate testen: hvordan skal en robot kunne håndtere at den ikke selv vet hvor den er?

BESTIKKREGNING

Bestikkregning er en gammel metode fra skipsnavigasjon, hvor man starter på en kjent posisjon, for så at man vet kursen og hvor langt man har reist, så kan man vite den nåværende posisjonen sin. Dette kan formuleres som

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta s_k \begin{bmatrix} \cos \theta_k \\ \sin \theta_k \end{bmatrix} \quad (11.1)$$

hvor \mathbf{p}_{k+1} er den nåværende posisjonen, \mathbf{p}_k er det kjente tidligere punktet, Δs_k og θ_k er henholdsvis distansen og kursen. k er en måling i tid, og $k+1$ er da en fremtidig posisjon gitt tidligere målinger og tilstander ved k . Dette gir ikke helt mening når man tenker at nåværende måling burde gi nåværende posisjon, ikke den neste posisjonen. Men man bruker gjerne k for forrige posisjon og k for nåværende måling for å kunne ha k på ene siden av ligningen og $k+1$ på andre, som gjør det enklere å beskrive matematisk.

Enkelte plasser vil du kunne komme over termen "dødregning" istedenfor bestikkregning. Dette er fra den engelske termen "dead reckoning", som kommer fra at loggen ikke flytter på seg "dead in the water".



En loggline er et instrument som ble brukt til å måle skipshastighet på. Loggen eller loggflyndren er en trekile, herav navnet "piece of log", som kastes over bord og fungerer som et anker og fast punkt. Loggen er festet til en line på en rull, hvor linen hadde knuter eller merker (knoper) med en best avstand. Når skipet seilte vekk fra loggen, så kunne man ta tiden på hvor mange knop som passerte over bauen, og man kunne regne om dette til nautiske mil per time.

Disse målingene ble skrevet ned i dekkdagbøker, såkalte loggbøker, og er opphavet til ordene loggføre, logging og andre varianter.

Det største usikkerhetsmomenter med bestikkregning er hvor nøyaktig du klarer å måle kurs og distanse. I klassisk skipsnavigasjon var det vanlig å bruke kompass, klokke og en loggline. Klokkene kunne gjerne bli upresise i løpet av en reise og logglinen tok ikke høyde for for eksempel havstrømmer, noe som fører til at målefeil akkumulerer og man ender opp med feil posisjon. Vi kan derfor korrigere ligning (11.1) ved å legge til målefeil og estimat.

$$\hat{\mathbf{p}}_{k+1} = \hat{\mathbf{p}}_k + \tilde{s}_k \begin{bmatrix} \cos \tilde{\theta}_k \\ \sin \tilde{\theta}_k \end{bmatrix} \quad (11.2)$$

hvor hatten $\hat{\mathbf{x}}$ beskriver et estimat og tilden $\tilde{\mathbf{x}}$ beskriver en måling med en viss unøyaktighet. Hvordan denne unøyaktigheten er beskrevet er ikke så viktig, hvertfall ikke nå.

Bestikk kommer fra det nederlandske ordet bestecken, som betyr å stake ut. Det er også her "å stake ut en kurs" kommer fra.

ODOMETRI

Når vi ikke er til havs, men har en kjørende robot, så er det mulig å bruke odometri for å gjøre bestikkregning. Et odometer (*fra gresk hodos som betyr vei og -meter*) er en distansemåler hvor man teller hvor mange omdreininger et hjul har. Hvis man vet diameteren på hjulet, vil man også kunne måle hvor langt det har rullet. Tripptellere eller kilometertellere på biler er eksempler på hvor odometri er brukt.

Men ett odometer vil kun kunne måle hvor langt ett hjul har rullet, som gjør at du kan måle distansen, men ikke kursen. Man kan derimot bruke to hjul og to odometer, og å bruke forskjellen mellom målingene til å regne ut kursen. Vi kan utvide ligning (11.1)

$$\begin{aligned} \mathbf{p}_{k+1} &= \mathbf{p}_k + \Delta s \begin{bmatrix} \cos(\theta_k + \frac{\Delta\theta}{2}) \\ \sin(\theta_k + \frac{\Delta\theta}{2}) \end{bmatrix} \\ \theta_{k+1} &= \theta_k + \Delta\theta \end{aligned}$$

hvor Δs og $\Delta\theta$ er målingene distansen og kursen siden sist måling. Vi kan regne ut begge målingene med ligningene

$$\Delta s = \frac{1}{2}r(\Delta\phi_H + \Delta\phi_V) \quad (11.3)$$

$$\Delta\theta = \frac{\Delta\phi_H - \Delta\phi_V}{b} \quad (11.4)$$

hvor r er radiusen på hjulene, b er sporvidden og $\Delta\phi_H$ og $\Delta\phi_V$ er vinkelmålingene fra høyre og venstre hjul. En robot med to hjul hvor man styrer begge hjulene uavhengig av hverandre kalles en *robot med differensialhjul*. Det vil si at hjulene kan modelleres som et differensialgir.

Hvis vår robot kjører med forskjellige hastigheter på hjulene så vil den kjøre i sirkel. Midtpunktet på roboten vil ha en hastighet på

$$v = R\omega \quad (11.5)$$

hvor R er radiusen på kurven roboten kjører langs og ω er rotasjonshastigheten langs sirkelen. Høyrehjulet og venstrehjulet vil begge også kjøre langs hver sin sirkel, den ene på innsiden av midtsirkelen og den andre på utsiden. Hastighetene deres vil være

$$v_H = (R + \frac{b}{2})\omega \quad v_V = (R - \frac{b}{2})\omega$$

Det gjør at vi nå kan finne R og ω uttrykt ved v_H og v_V .

$$\begin{aligned} v_H + v_V &= (R + \frac{b}{2})\omega + (R - \frac{b}{2})\omega \\ \Rightarrow R &= \frac{v_H + v_V}{2\omega} \end{aligned} \quad (11.6)$$

$$\begin{aligned} v_H - v_V &= (R + \frac{b}{2})\omega - (R - \frac{b}{2})\omega \\ &= b\omega \Rightarrow \omega = \frac{v_H - v_V}{b} \end{aligned} \quad (11.7)$$



Motorsykkelscenen i *Mission Impossible: Dead Reckoning Part 1* ble faktisk filmet i Norge. Faktisk i Hellesylt utenfor Ålesund. Vi spurte om Tom Cruise ville komme i bursdagen til bestemor siden han var veldig nært. Han svarte aldri.

Vi kan nå kombinere ligning (11.6) og (11.7) og få ut et uttrykk for kurveradiusen

$$R = \frac{b}{2} \frac{v_H + v_V}{v_H - v_V} \quad (11.8)$$

Her ser vi at hvis $v_H = v_V$ blir $R \rightarrow \infty$, og en kurve med uendelig radius er en rett strek. Roboten vil kjøre rett frem. Hvis $v_H = v_V$ så blir $R = 0$, altså en sirkel med null radius som er et punkt. Roboten vil snu på flekken.

La oss se hvordan vi kommer frem til ligning (11.3). Fra ligning (11.6) kan vi finne Δs ved

$$\begin{aligned} \Delta s &= v \Delta t = R \omega \Delta t = \frac{v_H + v_V}{2\omega} \omega \Delta t \\ &= \frac{v_H \Delta t + v_V \Delta t}{2} \\ &= \frac{v_H \Delta t + v_V \Delta t}{2} = \frac{\Delta s_H + \Delta s_V}{2} \quad (11.9) \end{aligned}$$

hvor Δs_H og Δs_V er distansen som hvert hjul har tilbakelagt siden sist måling.

Denne distansen er bestemt av vinkelen som hjulet har rotert og radiusen til hjulet. Altså

$$\Delta s_H = r \Delta \phi_H \quad \Delta s_V = r \Delta \phi_V \quad (11.10)$$

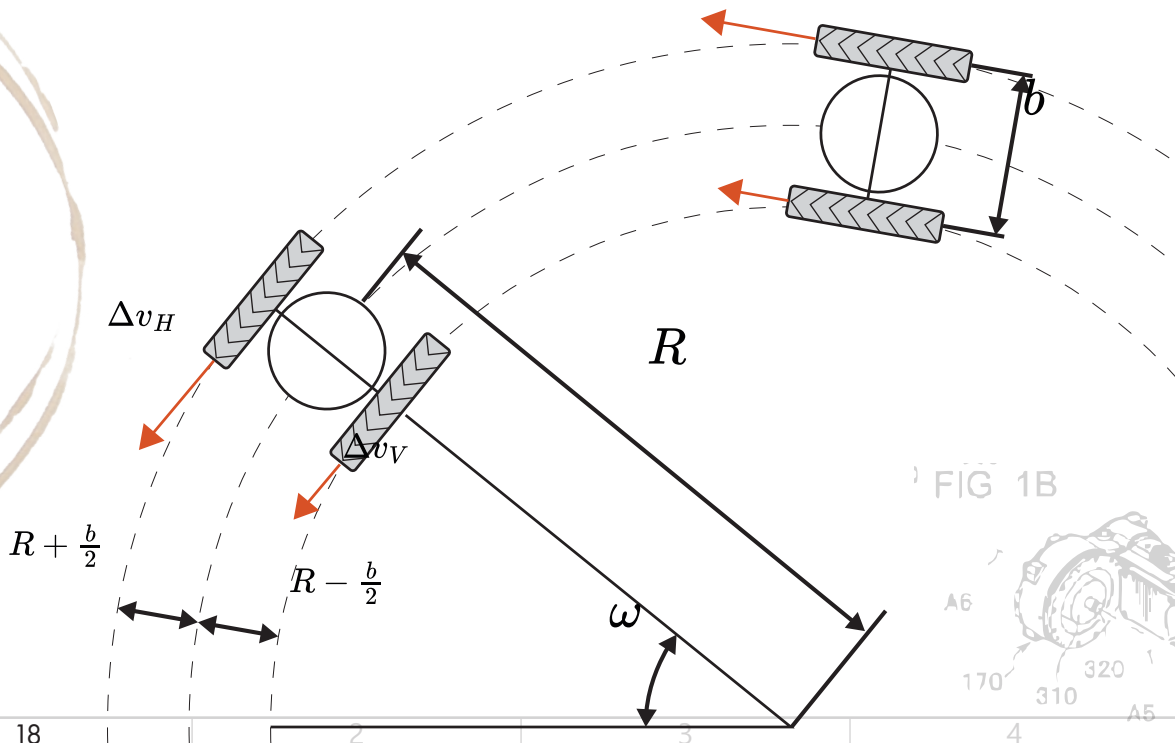
Som da gir oss ligning (11.3) hvis vi kombinerer med ligning (11.9).

For ligning (11.4) kan vi se at

$$\begin{aligned} \Delta \theta &= \omega \Delta t = \frac{v_H - v_V}{b} \Delta t \\ &= \frac{\Delta s_H - \Delta s_V}{b} \end{aligned}$$

Kombinerer vi dette med ligning (11.10) får vi ligning (11.4).

Kurveradiusen til en robot med differensialhjul. Her ser du også kurvene til begge hjulene til og hvordan de er relatert til sporvidden.



11.3 FEIL PÅ FEIL PÅ FEIL

Når vi nå har modellen for odometri, så er det en del forenklinger jeg har unnlatt å snakke om, og disse vil bite oss i rompa senere i kapittelet hvis ikke vi håndterer de nå.

Først, feilkildene i modellen kan deles inn i to kategorier: *systematiske feilkilder* er feilkilder som ligger i roboten vår, og *ikke-systematiske feilkilder* som er utenfor roboten vår.

De systematiske feilkildene i modellen er for eksempel hjulradiusen r og sporvidden b . Hvis vi har en feil på de verdiene vil vi akkumulere feil i hele modellen.

Eksempler på ikke-systematiske feilkilder er at det er glatt underlag og roboten sklir, noe som fører til at hjulene spinner mer enn hvor langt roboten beveger seg. Alle ytre faktorer som terreng, vær og vind, eller om roboten kræsjer eller hekter seg fast i noe vil føre til at posisjonen endrer seg, og vi kan hverken korrigere eller detektere om det skjer.

KALIBRERING AV ODOMETER

Når det kommer til systematiske feilkilder på en robot med to hjul, så er det hovedsakelig snakk om feil størrelse på hjulene, som fører til at roboten kjører kortere eller

lengre enn ønsket; forskjellig hjulstørrelse på høyre og venstre, som fører til at roboten vil dra til ene siden; og feil sporvidde, som vil få roboten til å over- eller undersvinge.

Kjører man en bestemt lengde fremover vil du kunne måle både om den faktiske og målte lengden er riktig og om den drar til ene siden. Kjører du rundt i en sirkel kan du bruke ligning R til å finne ut om sporlengden er riktig.

Man kan også kombinere dette i en *kvadrat-test*¹. Først kjører roboten en gitt lengde, hvor du måler den faktiske lengden s_{faktisk} mot den målt med odometri $s_{\text{målt}}$.

$$r_{\text{kalibrert}} = \frac{s_{\text{faktisk}}}{s_{\text{målt}}} r \quad (11.11)$$

Etter dette kjører roboten i et kvadrat mønster: Roboten kjører en gitt lengde fremover, svinger 90° med klokken, samme lengden en gang til og svinger 90°, og gjør dette frem til man har kjørt fem (eller flere)

¹ også kalt UMBmark eller University of Michigan Benchmark.

Vasco da Gama havnet i Kenya og ikke India i sin portugisiske karavell. Kan du liste opp systematiske og ikke-systematiske feilkilder, når han kun hadde kompass, stjerner, timeglass og loggline?



runder rundt et kvadrat. Så snur roboten og kjører samme ruten mot klokka i fem runder. For hver runde målet du hvor mye roboten avviker i posisjon og kurs fra startpunktet. En perfekt kalibrert robot skal ende opp på samme plassen.

Feilene kan vi beskrive som

$$\begin{aligned}\Delta x_{CW} &= \frac{1}{n} \sum_{i=0}^n \Delta x_{CW,i} \\ \Delta x_{CCW} &= \frac{1}{n} \sum_{i=0}^n \Delta x_{CCW,i} \\ \Delta y_{CW} &= \frac{1}{n} \sum_{i=0}^n \Delta y_{CW,i} \\ \Delta y_{CCW} &= \frac{1}{n} \sum_{i=0}^n \Delta y_{CCW,i}\end{aligned}$$

Vi kan kalibrere sporvidden b ved å bruke

$$\begin{aligned}\alpha &= \frac{\Delta x_{CW} + \Delta x_{CCW}}{-4L} \\ b_{\text{kalibrert}} &= \frac{\frac{\pi}{2}}{\frac{\pi}{2} + \alpha} b\end{aligned}$$

Vi kan regne ut kurveradiusen R ved at

$$\begin{aligned}\beta &= \frac{\Delta y_{CW} + \Delta y_{CCW}}{-4L} \\ R &= \frac{L}{2 \sin \frac{\beta}{2}}\end{aligned}$$

hvor L er lengden av kvadratet og β er vinkelen som roboten avviker.

Til slutt kan vi finne de kalibrerte radiusene

$$e_r = \frac{r_H}{r_V} = \frac{R + \frac{b}{2}}{R - \frac{b}{2}}$$

hvor e_r er forskjellen mellom den høyre og venstre hjulradius. De kalibrerte hjulradiusene blir

$$\begin{aligned}r_{H,\text{kalibrert}} &= \frac{2}{(1/e_r) + 1} r_{\text{kalibrert}} \\ r_{V,\text{kalibrert}} &= \frac{2}{e_r + 1} r_{\text{kalibrert}}\end{aligned}$$

Med disse kan vi finne de nøyaktige parametrene til roboten.

11.4 MODELLERING AV USIKKERHET

Selv om vi kan kalibrere en robot, så er det umulig å styre de ikke-systematiske feilkildene rundt roboten. Det betyr i praksis at uansett hvor mye vi prøver, så vil roboten drive av over tid uten noen form for absolutte målinger.

Vi kan ikke vite hvor mye feil som blir påført over tid. Eneste vi vet er at feilen akkumulerer. Dette betyr at usikkerheten øker over tid.

En måling gjort med et odometre kan modelleres som

$$\widetilde{\Delta\phi} = \Delta\phi + \nu_\phi$$

hvor ν_ϕ er prosessstøy. Prosessstøyen er en stokastisk variabel, altså at den ikke er regelmessig og at den ikke kan forutses. Vi modellerer den som en normalfordeling, som betyr at selv om den faktiske rotasjonen av et hjul er lik mellom hver måling, så vil den målte rotasjonen være ulik hver gang, og ulikheten vil ha en normalfordeling.

Med dette kan vi oppdatere ligningene for odometri til å inkludere støy

$$\widetilde{\Delta s} = \Delta s + \nu_s, \quad \widetilde{\Delta\theta} = \Delta\theta + \nu_\theta$$

og lage en tilstandsmodell for differensialhjul

$$\mathbf{p}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} \Delta s_k \\ \Delta\theta_k \end{bmatrix}$$

$$\mathbf{p}_{k+1} = f(\mathbf{p}_k, \mathbf{u}_k) + \mathbf{w}_k \quad (11.12)$$

hvor $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q})$, som er en måling av støymodell \mathcal{N} , og

$$\mathbf{Q} = \begin{bmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$$

er en kovariansmatrise som beskriver normalfordelingen til støyen.

Ligning (11.12) viser oss at selv om roboten starter på et punkt den kjenner nøyaktig, vil usikkerheten i posisjonen øke for hvert steg den tar. Kjører roboten rett frem, blir usikkerheten størst i kjøretretningen, som en ellipse som strekker seg foran roboten. Hvis den roterer, gjør det at ellipsen raskt sprer seg ut til begge sider. Over tid blir denne usikkerheten så stor at odometri alene ikke lenger er til å stole på.

Dette gir oss en viktig innsikt: odometri kan bare gi et kortsiktig estimat. Etter hvert må vi korrigere posisjonen med målinger som ikke driver over tid. GPS, radiosystemer og visuelle markører er eksempler på slike systemer. De fungerer som ankerpunkt som trekker roboten tilbake til en kjent posisjon og begrenser hvordan usikkerheten vokser.