### Automatisk Kjøring

**Ansvarlig: Anders Svalestad**

**HENSIKT  
Hensikten er å kjøre automatisk gjennom løype ved bruk av lyssensor.**

**OPPBYGNING**

**Programmet tar i bruk lys sensor for å kjøre robot automatisk gjennom utdelt løypeprofil. Løype banen er tegnet med gråskala som går fra hvit, mot sort. Vi ønsker å kjøre roboten i senter av banen. Hensikten med den automatiske kjøringen er at programmet skal styre roboten med minst mulig avvik fra banesenter.**

**Først opprettes kobling mot NXT og Joystick, variabler og figurer initialiseres. Tiden siden oppstart legges inn i siste element i tidsvektoren og en beregner tiden på forrige løkkegjennomkjøring ved å ta tiden i starten av denne løkken å trekker fra tiden i starten av forrige løkke. Denne skal brukes som delta tid (x) i kalkulasjoner.**

tid(end+1)=toc;

deltaTid(end+1)=tid(end)-tid(end-1);

**Lyset leses inn og filtreres basert på siste lys verdi og siste filtrerte lys verdi. Filtrert lysverdi bruker for eksempel 60% av inlest lysverdi sammen med 40% av siste verdi.**

lys(end+1)=GetLight(SENSOR\_3);

lysFilt(end+1)=filtLys([lysFilt(end),lys(end)]);

**Avviket for lyset fra nullpunkt regnes ut. Avviket integreres og legges inn i en egen vektor, dette gjøres i en egen funksjon ved bruk av en delta tid og avviket fra nullpunkt som blir en slags f(t). En legger til siste verdi av integralet for å få hele integral summen. Tilsvarende integrasjon gjøres for en egen vektor, men denne bruker absolutt verdien av løkke integralet for dermed å få totalt avvik fra nullpunkt.**

avvikL(end+1)=lysFilt(end)-lysNp;

avvikA(end+1)=intFunk(tid,avvikL)+avvikA(end);

avvikA2(end+1)=abs(intFunk(tid,avvikL))+avvikA2(end);

**Derivasjonen gjøres så i en egen funksjon ved bruk av en delta tid og delta y som regnes fra endringen i lyset verdi. (sjekker også for å unngå ikke kalkulerbare verdier).**

deriv(end+1)=derivFunk(tid,avvikL)

**Retningen avgjøres i egen funksjon og legges i egen vektor, det må være en endring fra forrige retning, dersom endringen gir 0 skal retning beholdes lik den foregående. Verdien i konkurransen utregnes fra oppgitt formel og legges til slutten av en egen vektor.**

**En bruker retningen til å sette variabler for visning av dette noe som også er beskrevet i egen seksjon under. Pådraget til hver av motorene regnes ut i egen funksjon basert på joystickens filtrerte inn verdier og pådraget settes.**

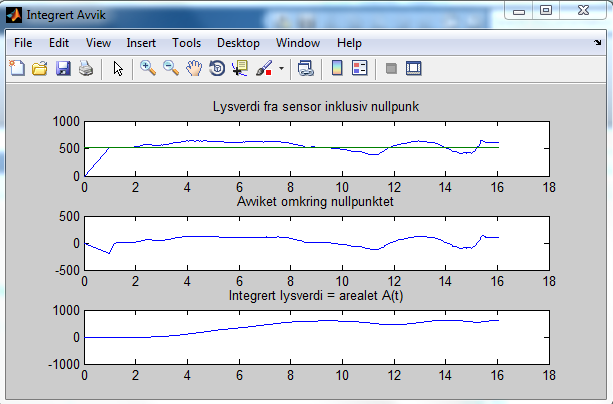
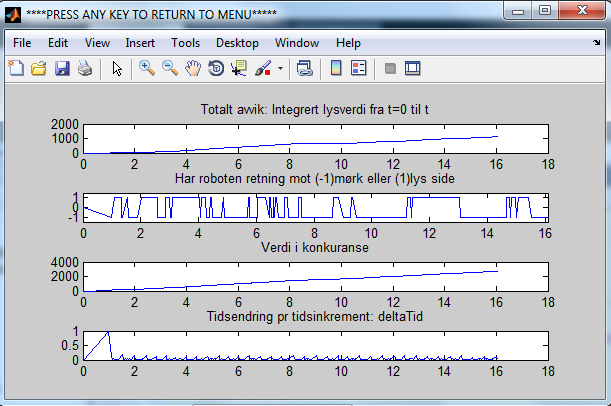
rettning(end+1)=retFunk(deriv(end),rettning(end))

verdi(end+1)=tid(end)\*100+avvikA2(end);

**Pådrag til hver motor beregnes ut fra lysverdiens avvik fra nullpunkt. En definerer en fast fart fremover. Pådraget til de to motorene blir deretter regulert ved å legge til/trekke fra en prosent av lysavvik fra 0 punkt.**

[paadragB(end+1),paadragC(end+1)] = Autofunc(avvikL(end));

**Når robot har fullført løype banen å går over på hvit papir, går lys verdien over en presatt verdi. (løkken avsluttes) Da stoppes motorene og sensor koblinger stegnes. Figurer fra kjøring stenges og nye mer detaljerte figurer basert på all data som har blitt tatt vare på tegnes. En løkke for å vente på at bruker skal kunne se på figurer settes før en kjøres til bruker trykker en knapp, da avsluttes programmet.**



**UTFORDRINGER**

**Lyssensoren har begrensninger å miljøet rundt påvirket resultatene i stor grad. For eksempel lysforhold i rommet du kjører, påvirker operasjon. Små justeringer av parametere må påregnes når miljø endres.**

**KONKLUSJON**

**Automatisk kjøring virker greit. Negativt at koden har en presatt makshastighet.**

**Løsningen er en relativ enkel pådragsregulering. Det finnes bedre, mer avanserte måter å løse oppgaven på som ville latt robot regulere hastighet , og kjørt løypen raskere.**