

Øving 0

(ønsker tilbakemelding)

Oppgave 1

Hvilke fysiske størrelser må måles av Segwayen?

I en normal tilstand bør kjøretøyet stå slik at tyngdesenteret er rett over balansepunktet. Dette har både brukeren og Segwayen delt ansvar for. Segwayen sin del av ansvaret består av å måle *tilstanden* til kjøretøyet. Tilstanden som blir målt er vinkelen og vinkelhastigheten, og det måles av et spesielt type gyroskop. «Solid-state angular rate sensor» heter det, og det bygger på Coriolis effekten. Måten det gjøres er ved hjelp av sillikonpartikler. Når man setter disse partiklene i bevegelse ved hjelp av elektrisitet, så beveger de seg på en veldig forutsigbar måte (Harris 2001). Men så fort det systemet begynner å rotere vil de endre bane og ut ifra denne målbare endringen blir vinkelhastigheten og vinkelen målt.

Dette målet må konverteres av en *signalomsetter* for å få et signal som datamaskinen i Segwayen kan forholde seg til og arbeide videre med. Signalet blir behandlet av Segwayens datautstyr som hovedsakelig består av to kretskort med flere mikroprosessorer hver. Det som er interessant er at begge kortene i seg selv faktisk er kraftige nok til å operere maskinen på egenhånd, så hvis det ene kortet skulle feile vil det andre kortet ta over. En av de mest sentrale delene av Segwayens programmering er regulatoren som sørger for at hele maskinen kan stå oppreist. Regulatoren bruker inputet fra gyroskopet og en forhåndsbestemt referansevinkel for å regne ut hvor stort pådragssignal det må gi til motorene. I Segwayen brukes det en PID-regulator. PID-regulatorer har tre ledd; Proporsjonalitet, integrasjon og derivasjon. Proporsjonalitetsleddet gir et pådrag som er proporsjonalt med avviket mellom ønsket vinkel og målt vinkel. Problemet med en proporsjonal regulator er at den ikke kompenserer nok. Når avviket er stort vil den kompensere mye, men etter hvert som avviket blir mindre blir kompenseringen mindre også og tilstanden til systemet vil aldri bli helt riktig. For å kompensere «ekstra» slik at tilstanden faktisk blir lik den ønskede verdien brukes integreringsleddet. Problemet med integreringsleddet er at det overkompenserer så derfor brukes deriveringsleddet, som er litt vanskeligere å forstå. Ved å tids-derivere avviket for man hvor fort avviket endrer seg, og dette kan brukes til å kompensere for at integreringsdelen overkompenserer (Svensson (Dr.Ing.) 2013).

Regulatoren har en samplingshastighet på 100 Hz (Harris 2001), som vil si at den sjekker avviket mellom gyroskopet og referansen og justerer pådraget 100 ganger i sekundet. *Pådragsorganene*, eller motorene, er det som faktisk påvirker tilstanden til Segwayen. Hvert hjul har én elektrisk motor hver. Hjulene er uavhengige av hverandre som gjør at de også kan brukes til å svinge. Når man lener seg (og hele «rattet») mot venstre økes pådraget til det høyre hjulet slik at Segwayen svinger mot venstre. Hvis hastigheten er lav nok vil det venstre hjulet også gå baklengs. En slik hjulmekanisme gir Segwayen veldig spesielle kjøreegenskaper som egner seg spesielt godt i urbane strøk.

Det er flere *redundante* systemer i Segwayen, og det er et bevisst valg av sikkerhetsmessige årsaker. Den har fem gyroskoper selv om man kun trenger tre for å måle de nødvendige aksene (foran-bak, venstre-høyre (roll)) (Harris 2001). Den har som nevnt to kretskort i tilfellet det ene skulle feile. Hvis det skjer vil det andre kretskortet ta over og sørge for at maskinen stopper kontrollert og trygt (SegwayStore SE).

Oppgave 2

Main-filen inneholder denne koden:

```
%definerer mu og simuleringstid
mu=5;
t_sim = 30;
%initialverdier
x1_0 = 1;
x2_0 = 1;

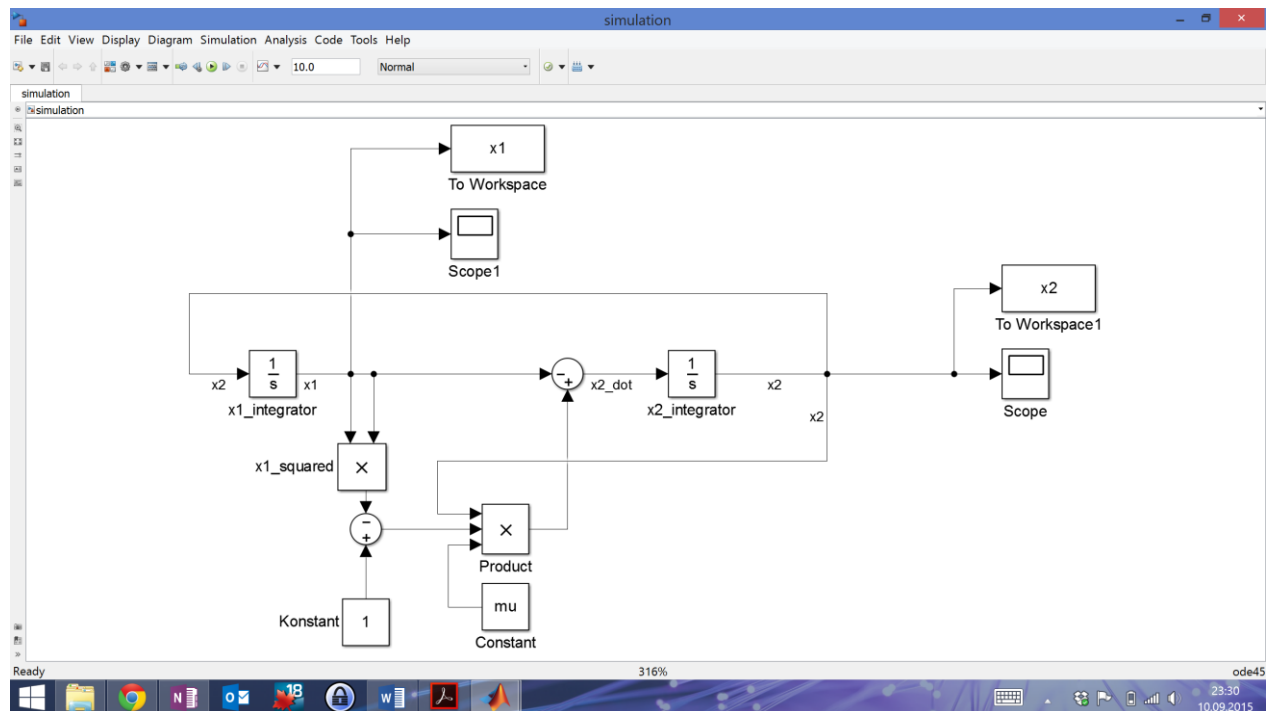
figure(1); clf(1);
hold on; grid on;

sim('simulation',t_sim);

%plotter data
plot(x1.data, x2.data);
legend_to_set=['x1_0 = ', num2str(x1_0), ' og x2_0 = ', num2str(x2_0)];

title('initialkondisjoner');
xlabel('x1'); ylabel('x2');
legend(legend_to_set, 'location', 'eastoutside');
```

Simulink-modellen ser slik ut:



Modell 1

Lagde en til fil (main_2) for å simulere flere initialtilstander i samme plott:

```
%definerer variablene
mu=5;
t_sim = 1000;
%initialverdier
x1_0_verdier = [-10,-9,-8,8,9,10];
x2_0_verdier = [10,5,-2,4,-10,3];

%formatering
formats = {'r','g','b','m','c','b'};

%legend, brukes mot slutten
legend_to_set = {};

figure(1); clf(1);
hold on; grid on;

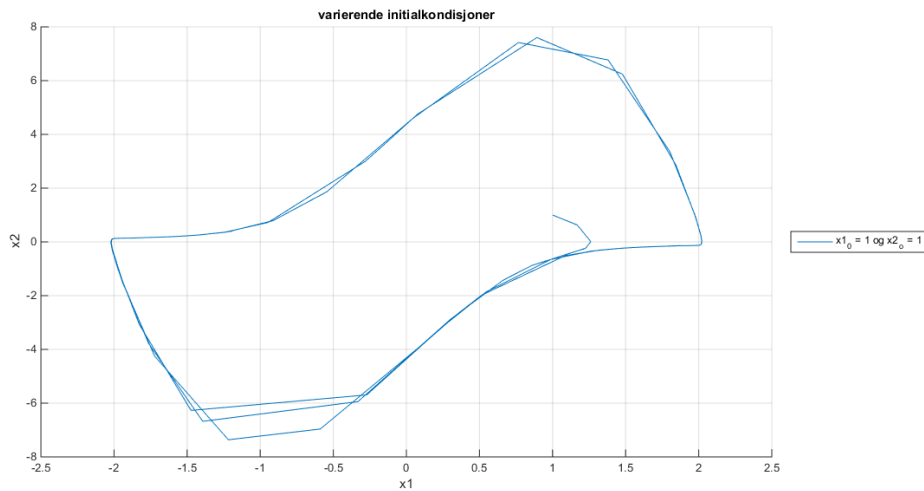
for i = 1:6
    x1_0 = x1_0_verdier(i);
    x2_0 = x2_0_verdier(i);

    sim('simulation',t_sim);

    %plotter data
    plot(x1.data, x2.data, formats{i});
    legend_to_set{i}=['x1_0 = ', num2str(x1_0), ' og x2_0 = ', num2str(x2_0)];
end

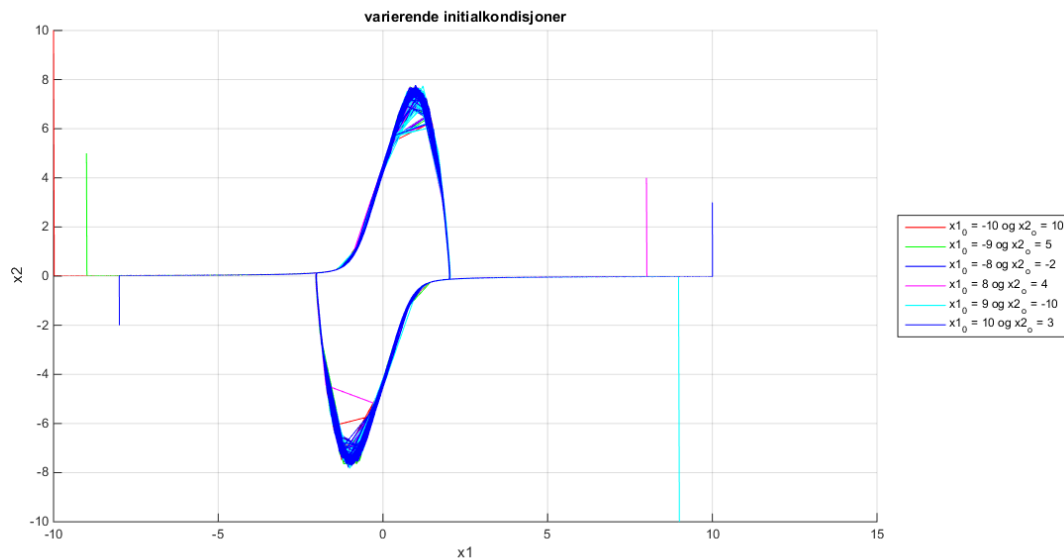
title('varierende initialkondisjoner');
xlabel('x1'); ylabel('x2');
legend(legend_to_set, 'location', 'eastoutside');
```

Main-filen gir dette plottet:



Figur 1

Main_2 simulerer mange forskjellige (tilfeldig valgte) initialtilstander og gir dette plottet:



Figur 2

I figur 1 ser vi at med initialtilstand (1,1) vil x_1 og x_2 konvergere fort mot en grensesvingning. Langs ordinat-aksen svinger det mellom ca. 8 og -8, og langs abschisse-aksen svinger det mellom ca. -2 og 2. I figur 2 ser vi at alle de forskjellige initialtilstandene vil konvergerer til samme grensesvingning, og at det er samme svingning som I figur 1. Jo større initialtilstandene er jo lenger tid tar det før grensesvingningen nås, men grensesvingningen vil alltid nås og den vil alltid være lik for alle initialtilstander (unntatt 0,0)

Referanseliste

Hachman, M. (2001) *The Technology Behind The Segway*.

<http://www.extremetech.com/extreme/72040-the-technology-behind-the-segway> [Hentet fra Internett 4.9.2015]

Harris, Tom. (2001) *How Segways Work*.

<http://science.howstuffworks.com/engineering/civil/ginger1.htm> [Hentet fra Internett 4.9.2015]

Svensson, M. C. (Dr. ing) (2013) *PID-regulatoren*.

<http://www.tu.no/automatisering/automatiseringsakademiet/kategorier/reguleringsteknikk/2013/03/30/pid-regulatoren> [Hentet fra internett 10.9.2015]

SegwayStore SE. *Segway – Hur fungerar det?* <http://www.segwaystore.se/om-segway.aspx> []