

EKSAMEN

Emnekode: ITD30005	Emnenavn: Industriell IT
Dato: 5.12.2018	Eksamenstid: 4 timer
	Robert Roppestad

Løsningsforslag.



Oppgave 1. (7%)

Overvåking: Innsamling av informasjon. Til bruk for beslutninger, sikkerhet.
Samler data fra sensorer.

Bør få med at **styring** er måling og påvirkning vanligvis **uten** bruk av aktiv tilbakekobling. I styring er det ofte mange digitale signaler: Eks: Trafikklys, Heis-styring, styring av produksjonslinjer, overvåking, osv....

Regulering betyr at man ønsker å få utgangen (resultatet) mest mulig likt en referanse (setpunkt). Regulering er aktiv bruk av **tilbakekobling**.
Eks: Regulering av temperaturen i et rom. Hastigheten på en bil (cruisekontroll). Autopilot. Nivået i en tank. Det finnes masse eksempler på hvor det benyttes regulering i dagliglivet.

Oppgave 2. (6%)

Det kommer av at det er en rekke sensorer og aktuatorer som er digitale (av/på) i tekniske systemer. Alarmer, overvåking, digitale sensorer.

Ventiler, motorer, en rekke givere for å detektere tilstedeværelse eller ikke.

Analoge elementer benyttes blant annet i reg.systemer. Man ønsker å måle eller styre til et nivå mellom 0% og 100%.

Nivåregulering, autopiloter, hastighetsregulering, temperaturregulering osv.....

Man har rene analoge sensorer for å måle noe helt «riktig».

Utførte undersøkelser viser at digitale signaler kan utgjøre mer enn 80% av alle signaler i et industrielt system.

Oppgave 3. (7%)

1. Eliminere feilen, få $e(t) \rightarrow 0$.
Få utgangen til å følge referansen best mulig.
2. Eliminere innvirkningen av en forstyrrelse $v(t)$ på utgangen $x(t)$.
Helst **0 (null) innvirkning** på utgangen.
3. Reguleringssystemet må **stabilisere** seg etter en påvirkning.
Falle til ro etter en påvirkning.
4. Minst mulig påvirket av variasjoner i **prosess-parameterne**.

Avvik: $e(t) = r(t) - y(t)$ Feilen, avviket i systemet

Pådrag: Tilsiktet påvirkning. $u(t)$

Tilstand: Karakteristisk størrrelse(r) som beskriver prosessen/systemet.

Som f.eks temperaturen i et rom. $x(t)$. Blir også kalt **prosessvariabel**.

Forstyrrelse: Utilsiktet påvirkning fra omgivelsene eller last endring. $v(t)$

Oppgave 4. (6%)

En matematisk modell er en beskrivelse av sammenhenger i systemer/prosesser.

Beskriver et system/prosesser med matematiske ligninger som gir oss muligheter for analyse og design.

Beskriver f.eks sammenhengen mellom påvirkninger og responser vha matematiske uttrykk.

Kan utvikles vha:

1. Basert på fysiske lover og balanser (fysisk modellering)
2. Eksperimentell analyse ved bruk av eksperimenter og databehandling. (System identifikasjon).

Hva modeller kan benyttes til:

1. For å simulere systemer
2. For analyse/design av styringssystemer
3. For å trene operatører i simulatorer
4. For optimal design av systemer
5. For å lage realistisk dynamikk i spill

Oppgave 5. (6 %)

1.

Vi ser at det er et 2. ordens system. Har 2 reelle poler.

Systemet er stabilt ettersom polene har negativ realdel.

Tidskonstantene er: $T_1 = 1$ og $T_2 = 0.5$ Forholdsvis små tidskonstanter.

En sprangrespons på et slikt system vil svinge seg pent inn mot en stasjonær verdi. Men det er ikke spurt etter hva en respons vil bli. Systemet har en forsterkning på 0.5. (Ikke spurt etter).

2.

Kryss-multipliserer ut høyere og venstre side og får:

$$X(s)[s^2 + 3s + 2] = U(s)$$

Vi vet at $s^2 X(s)$ betyr at det er den 2. deriverte osv..

Kan derfor overføre ligningen raskt til tidsplanet.

Vi går rett over i tidsplan og finner diff.ligningen som beskriver modellen.

$$\ddot{X}(t) + 3\dot{X}(t) + 2X(t) = U(t)$$

Oppgave 6. (6 %)

Frekvensresponsen uttrykker hvordan et system overfører sinussignaler med varierende frekvens. Et system vil både forsterke og fasedreie et sinussignal. Vi kan da som vist i figuren plote hvordan systemets forsterkning varierer med frekvensen, og hvordan systemets faseforskyvning varierer med frekvensen. Frekvensresponsen kan sies å være et slags fingeravtrykk for systemet.

Vi kan lese rett av figuren at: *Her må det godtas litt slingringsmonn.*

10 rad/s gir: Forsterkning -20dB som er 0.1. Det betyr at kun 10% av signalet (amplitude 0,1 Volt) slipper gjennom. Faseforskyvningen leser vi av til ca 85 grader.

1000 rad/s gir: Forsterkning 0dB som er 1. Det betyr at 100% av signalet slipper gjennom. Faseforskyvningen leser vi av til ca 10 grader.

Dette er et typisk **høypassfilter**. Demper lave frekvenser og slipper de høye rett gjennom.

Oppgave 7 (6%)

Stabilitetsmarginer angir hvor mye reguleringsystemet har å gå på før det blir ustabil. Må ha en viss forsterkningsmargin og fasemargin for at systemet skal oppføre seg «pent» / "godt" stabilt.

Bli **marginene for store** blir reg.systemet **tregt**. Langsom innsvingning med mulighet for **stor** dynamisk og stasjonær feil.

Motsatt hvis **marginene er små** vil reg.systemet bli veldig **hurtig/følsomt**, og dermed gi oscillasjoner i utgangen. Det vil også kunne oscillere så mye at responsen ikke er anvendelig.

Kan bli ustabil.

Pidtune() er en Matlab-funksjon som benytter **åpen sløyfetransferfunksjon A(s)** som input.

Den vil gi oss P, I og D parametere som gir et reg.system som avveier krav til stabilitet, nøyaktighet og hurtighet. Sikrer fornuftig stabilitetsmarginer.

Forenkler jobben med å finne gode P, I og D verdier.

Output er : Kp, Ki og Kd som inngår i en PID-algoritme.

Vil som standard oppfylle en fasemargin på 60 grader.

Oppgave 8. (8%)

1.

Da bør **K** velges som $K_{krit}/2 = 30/2 = 15$ (Det betyr 6dB f-margin)

2.

Bør velge en **PI-regulator**. Må tune inn T_i som bidrar til å fjerne stasjonær feil.

T_i må ikke velges for liten, men stå i forhold til prosessen tidskonstanter.

Er T_i for stor har den dårlig virkning.

3.

Her bør nevnes

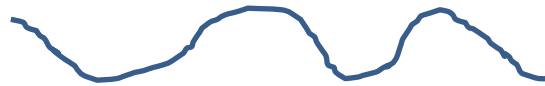
- Sprangrespons test metode.
Utfører et sprang. Beregner parametere ut fra responsen.
- Relè metoden. Forklare den kort.
- Ziegler-Nichols metode. Egner seg som en manuell metode...
- Mange andre metoder. Finn Haugens, osv...

4.

Som navnet tilsier vil utgangen fra regulatoren enten være av eller på.
Når utgangen når sitt setpunkt slås pådraget av. Når utgangen er lavere slås pådraget på.
Dermed vil utgangen (resultatet) hele tiden svinge rundt ønsket verdi.
Vi får ingen stabil utgang.



Pådrag



Respons/utgangen

Oppgave 9 (4%)

Redundans kalles informasjon som **gjentar allerede etablert kunnskap** uten å tilføre noe nytt. Redundant informasjon kan derfor også kalles **overskudds-informasjon**.

Benyttes for å øke pålitelighet i systemer. Øket sikkerhet.

Redundans bygges ofte inn i systemer som krever **høy pålitelighet**.

I datasystemer kan **to eller flere datamaskiner** jobbe **parallelt** med samme oppgaver og **speile hverandre**. Dersom en av dem skulle «gå ned» så kan den andre ta over.

Redundans betyr at vi øker antall elementer/moduler i et systemer for å øke sikkerheten/pålitelighet.

I systemer som krever stor sikkerhet vil det være fornuftig å dublere måle-elementer og sentrale moduler for å opprettholde kontroll hvis noen elementer svikter.

I gassproduksjon vil man kunne benytte både 3 og 4 trykk-sensorer som måler samme trykk for å opprettholde pålitelighet/sikkerhet viss en sensor svikter.

Systemer hvor redundans er nødvendig er typisk prosesser som **ikke MÅ** feile..

Olje/gass, atomkraftverk, fly, selvgående kjøretøy osv...

Oppgave 10. (7%)

1.

Nøyaktighet er aldri bedre enn antall bit som benyttes for å holde den digitale verdien.

Aldri bedre enn : $\pm 0.5 \text{ LSB}$. Vanligvis oppnår en ikke bedre enn ± 1 til 2 LSB .

(LSB- minste signifikante bit)

Eks **12bit: 1/4096 (0.02%)**

Med spenning fra 0 -> 5 volt blir max nøyaktighet. **$5V/4096 = +/- 0.6 \text{ mV}$ ($+/- 0.5 \text{ LSB}$)**

A/D omformere kan ha linearitetsfeil, feil pga temperatursvingninger, feil pga av dårlig elektronikk, at komponenter eldes osv....

2.

Shannon samplingsteorem stadfester at man aldri kan gjenskape frekvenser som er høyere enn $0.5 * \text{samplingsfrekvensen}$. Kalles og Nyquist frekvensen.

I dette til betyr det at vi maks kan gjenskape frekvenser opp til 5000 Hz.

Frekvenser over $F_s/2$ vil nedfoldes og dukke opp som **falske** frekvenser i det samlede signalet. Dette kalles **alias-frekvenser**.

For å unngå problemet så kan man tenke seg at man sampler så hurtig at det ikke finnes frekvenser over $F_s/2$. Det vil normalt ikke være realistisk, så man må heller filtrere bort alle frekvenser over $F_s/2$ med et **lavpassfilter** (anti-aliasfilter) **før sampling**.

Oppgave 11. (7%)

1.

I svært mange sammenhenger har man behov for å filtrere signaler/bilder.....

Filtrering kan utføres analogt eller digitalt.

Hvorfor benyttes filtrering?

- Fjerne uønskede frekvenskomponenter.
- Framheve ønskede frekvenskomponenter.
- Samt en rekke andre oppgaver innen signalbehandling, avansert regulering, bildebehandling.

Eks

- Fjerne målestøy. Lavpassfiltrere.
- Fjerne 50Hz nettstøy.
- Fjerne frekvenser over $f_s/2$ før sampling. Hindre nedfolding.
- Framheve et gitt frekvensområde.

Analoge filtre konstrueres vha elektriske komponenter (aktive eller passive).

Ofte benyttes operasjonsforsterkere (aktiv komponent).

Analoge filtre trenger ingen CPU, men er ikke så fleksible som et digitalt filter.

Digitale filtre er **diskrete varianter av de analoge**.

Programmerbare og lett å modifisere. De krever CPU-tid.

2.

MA-filteret er et enkelt **lavpassfilter**. **Glatte et signal**.

Virker ved at man beregner **gjennomsnittsverdien** av de **n** siste samplingene.

Får da et MA-filter (Moving Average filter, bevegelig middelværdi).

Lager en algoritme som tar middelverdien av de n siste samplingene.
Lett å programmere.

$$\rightarrow U_k = \frac{1}{n} (E_k + E_{k-1} + \dots + E_{k-n+1})$$

E_k – samplet signal som skal filtreres

U_k – filtrert signal

Oppgave 12. (9%)

1.

Regner ut den diskrete (digitale) varianten av filteret. Benytter Eulers metode.
Løsningen innebærer at filtrert utgang X_k er avhengig av gamle verdi X_{k-1} og U_k .
 U_k er signalet som skal filtreres.

$$a \dot{X} + X = U$$

$$\Rightarrow a \frac{X_k - X_{k-1}}{T} + X_k = U_k$$

$$a X_k - a X_{k-1} + T X_k = T U_k$$

$$X_k [a + T] = a X_{k-1} + T U_k$$

$$X_k = \frac{a}{a + T} X_{k-1} + \frac{T}{a + T} U_k$$

$a \gg 1$ a stor \Rightarrow kraftig filtrering
 $a \approx 0$ a liten \Rightarrow liten filtrering

T - samplingstiden
 $a = \frac{1}{\omega_f}$, ω_f - cut off frekvens

2.

T er samplingstiden. Den må stå i forhold til filterets cut-off frekvens (tidskonstant $a = 1/\omega_f$).
For stor T vil medføre at filteret kjøres for langsomt, og det vil bli en dårlig tilnærming til den analoge varianten (utgangspunktet). T bør være mindre eller lik $0.1 \cdot a$.
 a er en faktor som avgjør hvilke frekvenser som skal kuttes.
Rent praktisk vil a være: $1/\omega_f$ der ω_f er filterets cut-off frekvens.

Hvis a er tilnærmet 0 vil det være liten filtrering av frekvenser.

Hvis $a \gg 1$ vil filteret ha stor virkning og filtrere bort høyere frekvenser.

Oppgave 13. (5%)

ESP32 er benyttet i lab.

Fordeler

- Billig.
- Liten. Veier kun noen titalls gram.
- Strømgjerrig
- Har både analog og digital IO, Men begrenset antall.
- Har forholdsvis stort internt minne. Rask CPU.
- Støtter en rekke bus-systemer
- Støtter Bluetooth og WiFi
- Mange som utvikler «åpne» løsninger til plattformen.
- Lett å få til trådløs kommunikasjon. Web-server løsninger.
- Passer godt inn i IOT (Internet of thing) utvikling.

Ulemper

- Begrenset IO-støtte
- Kan ikke takle store oppgaver med mye IO. Ikke så lett å skalere opp.
- Ikke gjennomprøvd driftssikkerhet.
- Ikke tilpasset industrielle løsninger som PLS er dedikert til.
- Har ikke utvalget i IO-moduler som PLS'er har.
-

ESP32 og PLS'er vil følgelig kunne dekke ulike type oppgaver i styring, overvåking.

Oppgave 14. (16%)**Del 1.**

Her kan SR-vippe, Timere og triggere benyttes.

Mulig med mange ulike løsninger.

Viktig at man sikrer at utganger ikke styres ut flere steder i programmet.

Løses greit ved å benytte 4 eller 5 tilstander som gjennomløpes sekvensielt.

Løsningen som vises baserer seg på det.

At alle lys skal slås av når start/stopp bryter er **av** kan sikres med OG-blokker på utgangene som styrer lysene. Kan også være lurt og la lyskrysset parkeres seg i sekvens 1 når det slås av.. Gjøres ved å utføre OG mellom Restart og startlyskryss.

En grei timer er TP-timer. Men TOF timer vil også funke greit.

Se FBD-løsningen som følger. Dette kan løses på mange ulike måter.

PS!! Løsningen viser både lyskryss nord_syd samt øst_vest.

Den benytter også en tilstand 0 hvor alle lys er røde noen sekunder ved oppstart.

Del 2.

--

Når en fotgjenger trykker på knappen lagrer vi det vha en SR-vippe med utgang C! i figuren.. Utgangen ligger da høyt. Kjører dette signalet og rødt signalet (tilstand 1) inn på en OG-blokk. Når begge er sanne får fotgjengere grønt. Når det grønne fotgjenger lyset slås av benytter vi en F_trig for å resette SR-vippa for fotgjengere. Sikrer også at hvis fotgjengerbryter IKKE er trykket så vil tilstand 1 gi rødt lys for fotgjengere.

Mulig løsning. Testet i Codesys.

