**Øvelse 6: DC-motoren som positionsservo med digital Lag- regulator**.

1. **Øvelsesobjekt**

Øvelsesobjektet består af samme motorstand, som blev anvendt i øvelse 4.

Motor, tachometer, gear, ekstra inertibelastning og potentiometeret til måling af vinkeldrej­ning udgør reguleringsobjektet.

Tillige bruges et storagescope, funktionsgenerator, Power Amplifier, KPA, og Control box, Kp, hvortil tilsluttes et PSoC-modul til realisering af den digitale regulator. PSoC’en påmonteres et konvertermodul, så ±10Vpp signalniveauer kan behandles (se billede).



Effekttrin for DC-motor

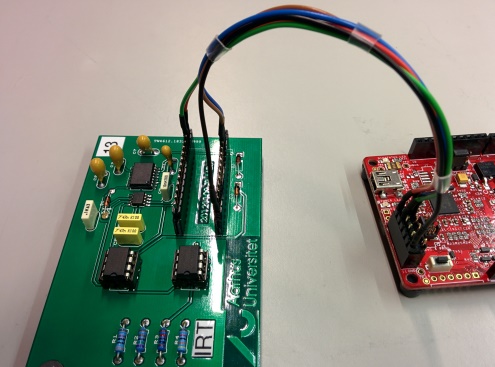
Ea motorspænding

10 turns potentiometer til måling af vinkeldrejning.



PSoC4 startkit

med interface board



**2. Formål.**

* At underbygge forståelsen af bogens gennemgang af digital regulering
* At få kendskab til, hvorledes en analog regulator kan realiseres digitalt (bilineær transformation).
* At få kendskab til, hvorledes tidsforsinkelser influerer på reguleringssystemer.
* At få kendskab til, hvorledes valg af samplingfrekvens påvirker reguleringssystemer.
* At indøve brugen af Matlab på tidsdiskrete (digitale) reguleringssystemer.
* At få forståelse for samspillet mellem teori, simulering og måling.

1. **Systemoversigt**

Controller box

Kp

Motor, gear belastning

m.m.

Gms(s)

Poten-

tiometer

Kpot

+

-

Ref.

Power Amplifier

KPA

PSoC

Ө

VӨ

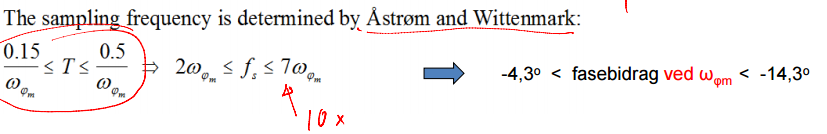
Vind

**4. Forberedelse.**

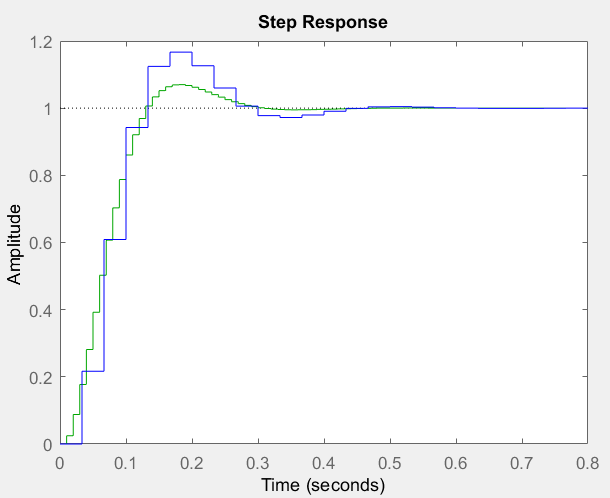
Power Amplifier og Motorstand med potentiometer til måling af vinkeldrejningen, har overføringsfunktionen

Med KPA = 3,8 og Control box: Kp=1 Gain x10, er systemets fasemarginfrekvens, ωφm = 15 rad/s.

* 1. Undersø­g hvilken indflydelse, det har på systemets stepresponse, at PSoC’en med forstærkningen 1 gg indsættes i reguleringssløjfen efter Control box’en.  
     Undersøg virkningen ved både den højest og lavest anbefalede samplingsfrekvens iflg. Åstrøm og Wittenmark.   
     Verificer undersøgelserne ud fra Bodeplot og stepresponse.  
     Den z-transformerede overførings­funktion af processen bestemmes som: G(z) = Z{Gh(s­)∙G(s)}.



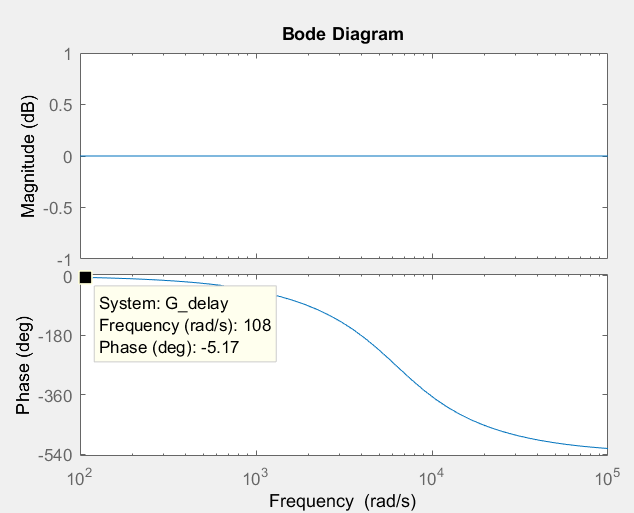
Zoh:



Blue is low frequency and green is high frequency

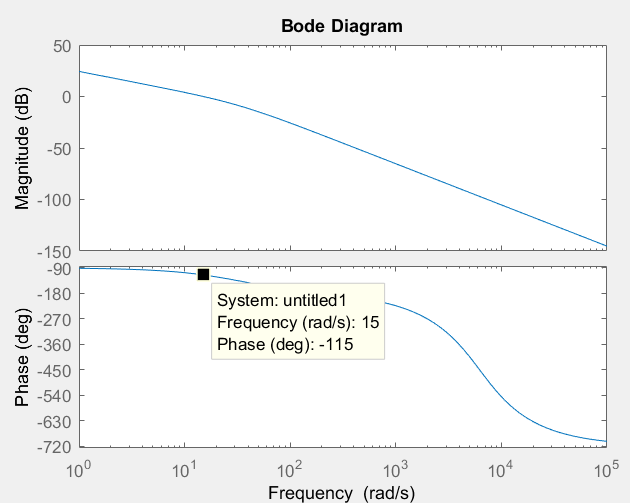
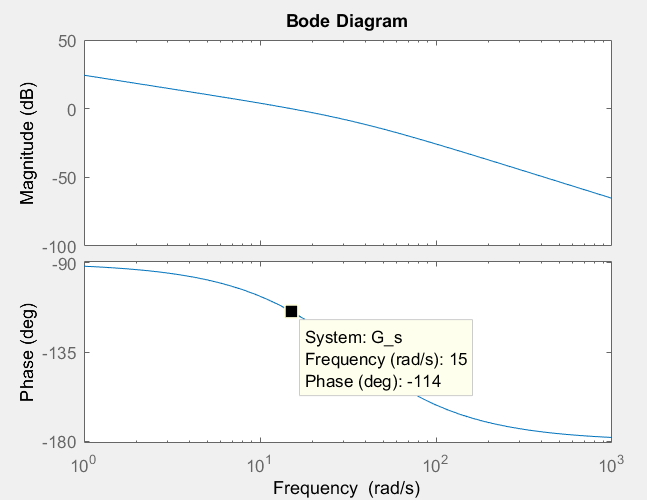
* 1. Vi vil efterfølgende forestille os, at der sammen med Motorstanden er en tidsforsinkelse på Td = 0,8 ms. Beregn fasebidraget fra denne tidsforsinkelse og sammenlign med åbensløjfe overføringsfunktionens frekvenskarakteristikker. Hvordan forventes systemet nu at reagere? Er det nødvendigt at medregne en sådan tidsforsinkelsen ?



Phase is almost 0 at the maxium samplingsfrequency of 100Hz, also 0.8ms is not a very high time delay if the samping time is 10ms.

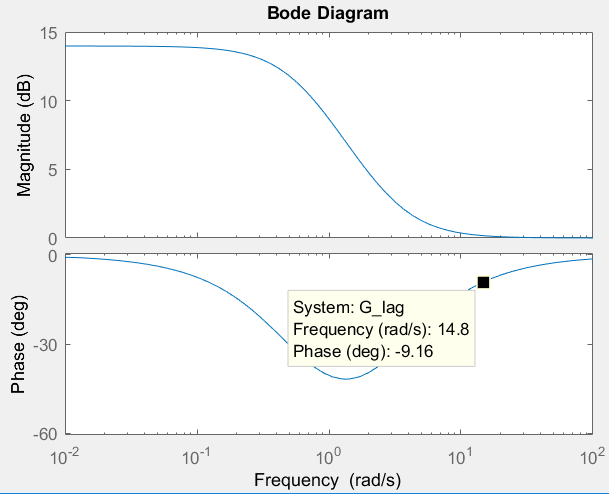
The system with this 0,8ms delay does not contribute much since the phase contribution is only 1 degrees as shown below

Before after 0.8ms delay



Følgende overføringsfunktion for en Lag-regulator anvendes efterfølgende: .

* 1. Verificer ud fra Bodeplot Lag-regulatorens forventede påvirkning af stationære- og dynamiske egenskaber for positionsservoen.



Lag-regulatoren overholder ikke det krav, vi tidligere har haft til den, nemlig at ”den skal placeres ved en frekvens der er 10 gange højere end fasemarginsfrekvensen”. Derfor kommer det negative fasebidrag til at have indflydelse på fasemarginen, hvilket resulterer i et større oversving. Det ses også af både simuleringen nedenfor, i opgave 4, og på scope-billederne i realiseringen.

* 1. Simuler lukket sløjfe systemets ramperespons (hældning a = 0,4). Simuler for den højest anbefalede samplingfrekvens uden Lag-regulator og med Lag-regulator.   
     Den digitale Lag-regulator bestemmes ved en bilineær transformation (Matlab/ ’Tustin’) af den analoge Lag-regulator Gc.  
     Den z-transformerede overførings­funktion af processen bestemmes som: G(z) = Z{Gh(s­)G(s)}, ved brug af Matlab, ‘ZOH’.

Nedestående Matlab kommandoer vil danne rampe-svaret for overføringsfunktionen Tz:  
t = 0:T:4;

x = sawtooth(4\*pi/4\*t,0.5);

y = lsim(Tz,0.2\*x);

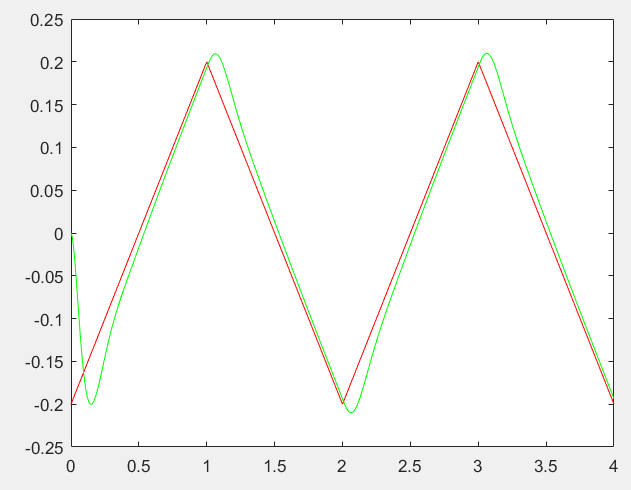
plot(t,0.2\*x,'r',t,y,'g');

Beregn den stationære fejl teoretisk og sammenlign med resultaterne fra simuleringen. Kommenter indsvingningsforløbet.

Vi beregner steady-state fejlen for systemet med formlen:

Den teoretiske steady-state fejl er altså 0.0606 før vi anvender Lag-regulatoren.

I simuleringen finder vi fejlen ved at aflæse forskellen mellem input og output på grafen. Her fås fejlen til 0.0242. Efter regulatoren fås fejlen i simuleringen til 0.0064, som er aflæst fra figuren nedenfor.



Forskel mellem ’tustin’ og ’zoh’ i MATLAB:

c2d med tustin-parametren anvendes til at konvertere analoge filtre til digitale filtre. Heri indgår tidsforsinkelsen ikke, og derfor er %OS også det samme i det digitale domæne.

c2d med zoh-parametren anvendes til at konvertere en analog proces til en digital proces. Heri indgår den forsinkelse, der er et resultat af 0. ordens holde-funktionen, og den rigtige gain indgår også.

**5. Øvelsen.**

Oversving beregnes, med følgende formel, ud fra aflæsninger fra scope-billeder:

**Vigtigt! Afbryd strømmen til motorstanden når PSoC’en programmeres!**

1. Kobbel systemet op som en analog positionsservo (vist i systemoversigten, blot uden PSoC’en). Kontroller systemets funktion med en et firkant signal, ±200 mV og 0,5 Hz, som reference.  
   Oversving < 5%

OSPeak: -108mV, steadystate: 96mV, startpunkt: 100mV

1. Undersøg nu hvilken indflydelse samplingfrekvensen (højeste og laveste) har på positionsservoens egenskaber, ved at indsætte PSoC-modulet umiddelbart foran Power Amplifier, KPA. Benyt samme firkantsignal som i a).  
   I denne situation skal forstærkningen gennem PSoC-modulet blot være 1.  
   **Om programmeringen af PSoC’en, se appendix: IRT\_EX6\_1.**

100Hz

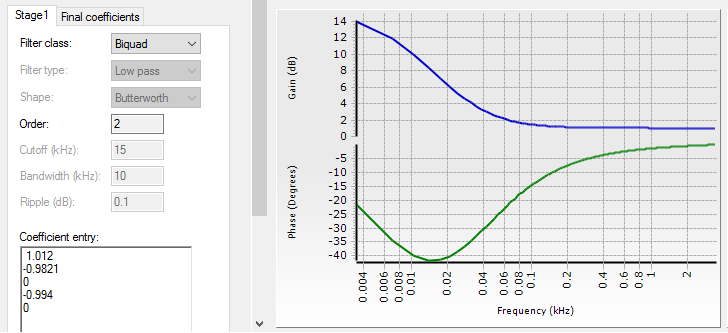
30Hz

Heraf ses det, at den langomme sampling giver et højere oversving, hvilket giver god mening, da der går længere tid imellem selve reguleringen, og dermed går der længere før systemet reagerer. => Højere OS.

1. Afprøv den digitale Lag-regulatorer fra forberedelsen. Sammenlign med Matlab simuleringer, og kommenter resultaterne (statiske og dynamiske egenskaber). Som rampeinput anvendes funktions­generatorens trekantsignal med en amplitude på ±200 mV og 0,5 Hz.  
   **Om programmeringen af PSoC’en, se appendix: IRT\_EX6\_2.**

På scopet ses, at der er et offset på målingen, som stammer fra usikerheder i kredsløbet. Disse skal vi se bort fra for at se hvad der sker med steady-state fejlen på scopet. Når vi indstiller de to grafer, så oversvinget af outputtet (det grønne) er lige stort på begge sider.

På figuren ses en logaritmisk afbildning af Lag-regulatorens amplitude- og fasekarakteristik.



Heraf ses, at karakteristikken stemmer overens med teorien for Lag-regulatoren, når vi indsætter vores koefficienter i PSoC Creator.

# Appendix, Brug af PSoC modulet i øvelse 6:

Selve PSoC’en er som det tidligere brugte udviklingskit. Har man et PSoC4 udviklingskit skal PSoC5 modulet tilgåes Skabelonen passer til disse. Dog skal der eftermonteres et konvertermodul, så ind- og udgangssignaler dækker området ±10V. Da modulet er lavet til PSoC3 og 5, skal et stik monteres på PSoC4 startkittet.

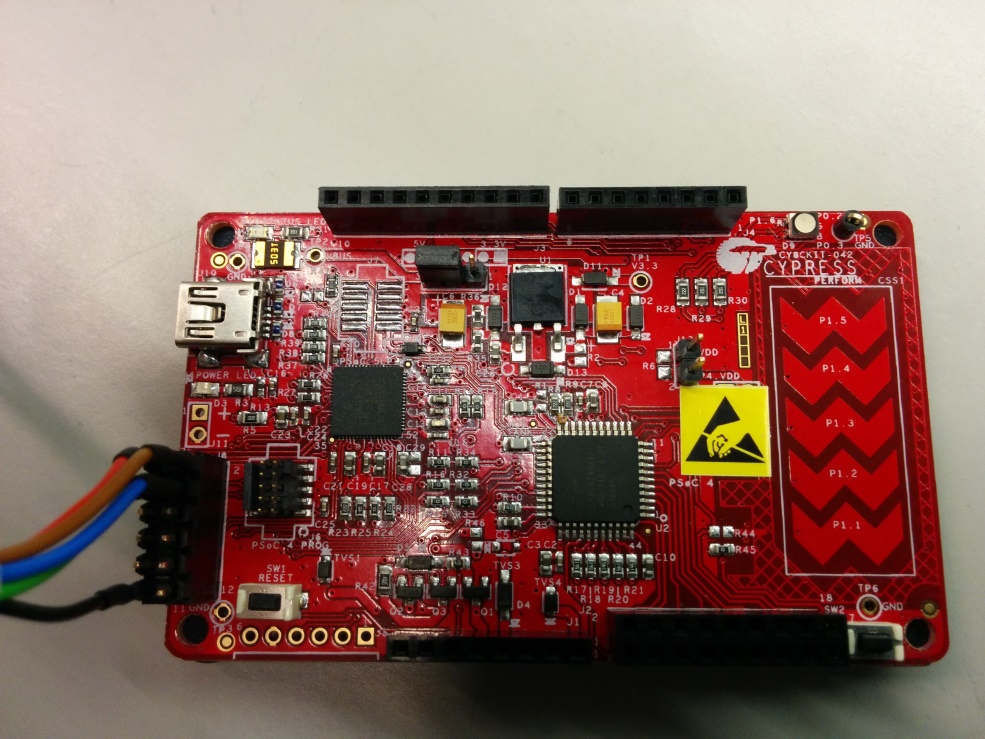
BNC stik til analogt ind / ud og USB stik til Labtop’en. Strømforsyningen sker gennem USB-stikket.

Fra BlackBoard hentes zip-filen E4IRT\_EX6.rar

Det indeholder 2 PSoC-projekter: IRT\_EX6\_1 og IRT\_EX6\_2.

**OBS: PSoC StarterKit skal sættes i 5-Volts mode for at benytte interfaceboarded! Dette gøres ved at sætte jumperne på PSoC StartKit som vist på figuren herunder. HUSK at PSoC skal være slukket når der flyttes på jumpere! Derudover slukkes der for PSoC4 modulet ved at fjerne jumperen, også vist herunder. Stikket fra interfaceboardet monteres ligeledes, som vist på billedet.**

**Kontrollér desuden at blokken Bootloadable peger korrekt på kit\_prog filerne, samt det er den korrekte bootlader host fil der bliver programmeret, som i kender det fra E3MSE.**

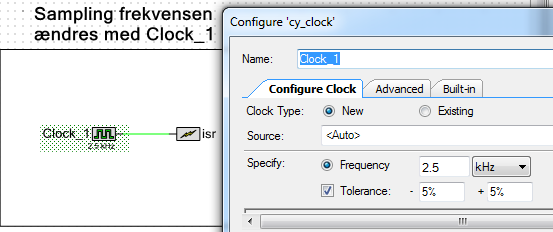
****

**IRT\_EX6\_1**

Benyttes i spm. b), hvor signalet blot samples og holdes. (1 gg forstærkning)

Denne skabelon sampler input signalet og sætter det direkte ud på output igen – et såkaldt *write through*.

Samplefrekvensen vælges i blokken **Clock\_1**, som værdien af clockens **frekvens.**



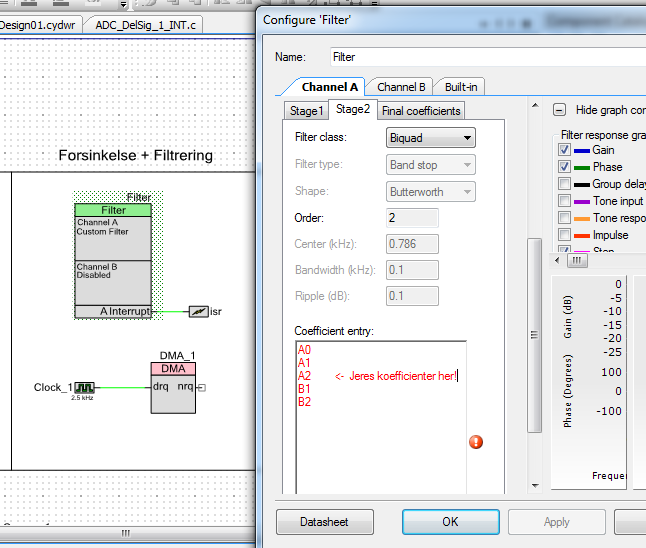
**IRT\_EX6\_2**

Benyttes i spm. c) til realisering af Lag-filteret.

Filteret er på PSoC’en realiseret som et standard 2.ordens IIR-filter på kanonisk form:

 A0 = 1; A2 = 0; B2 = 0 og et evt. K realiseres analogt.

Samplefrekvensen justeres som i skabelon 1.



|  |  |
| --- | --- |
| Stage1 | Stage2 |

|  |  |
| --- | --- |
| Stage1 | Stage2 |