Aarhus universitet

Adaptive Control and Automation

7. Semester

ACA projekt

Gruppemedlemmer:	*AUID*
Daniel Tøttrup	au544366
Stinus Lykke Skovgaard	au520659

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

13. december 2018

Indhold

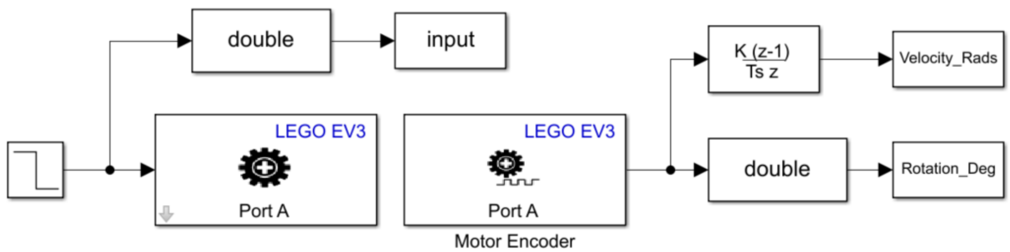
Figurer

1 Indledning

Vores første ide til projektet var at lave en "adaptiv følgebil". Tanken var at vores Lego bil kunne holde en bestemt afstand til et objekt i bevægelse, hvor svingende hastighed, skiftende hældning på underlaget, samt ændring af vægt monteret på vores Lego bil, ikke ville påvirke dens adfærd. Efter vi havde lavet test med udstyret til rådighed, besluttede vi os for at gå i en anden retning. Vi fik nemlig mere præcise målinger fra encoderen monteret inde i motoren, og ønskede derfor at gå videre den som vores sensor. Det betød dog også, at vores Lego bil ikke længere ville være i stand til at følge et objekt i bevægelse, da den ikke længere havde "øjne" i form af en afstandssensor. Ideen vi gik videre med, blev derfor en Lego bil der vil være i stand til at køre en specifik afstand, upåvirket af f.eks. vægt monteret på Lego bilen eller underlagets hældning. Derfor endte vores system i sidste ende med at bestå af en motor, med en indlagt encoder som sensor.

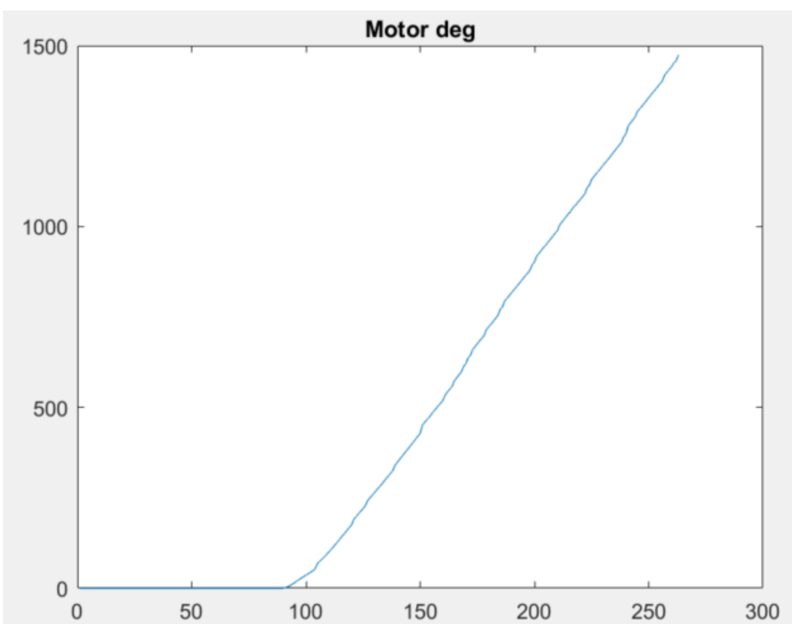
2 Identifikation af open-loop system

Efter vi har besluttet os for hvilket system vi ønskede at designe, og hvilke dele systemet skal bestå af, er det næste skridt at indsamle data fra vores motor via den indlagte encoder. Det gør vi ved at sende et step input ind på motoren. Vi aflæser to outputs, både det direkte output fra encoderen, som er rotationen i grader, samt den afledte som er hastigheden i grader. Opstillingen af denne måling i Simulink kan ses på Figure 1. Disse målinger gentog vi mange gange, for at få de bedst mulige målinger. Grunden til at vi ønsker de bedste mulige målinger er, så vi kan lave den bedst mulige repræsentation af vores system, så det system vi beregner på, er så tæt på det faktiske system som overhovedet muligt.

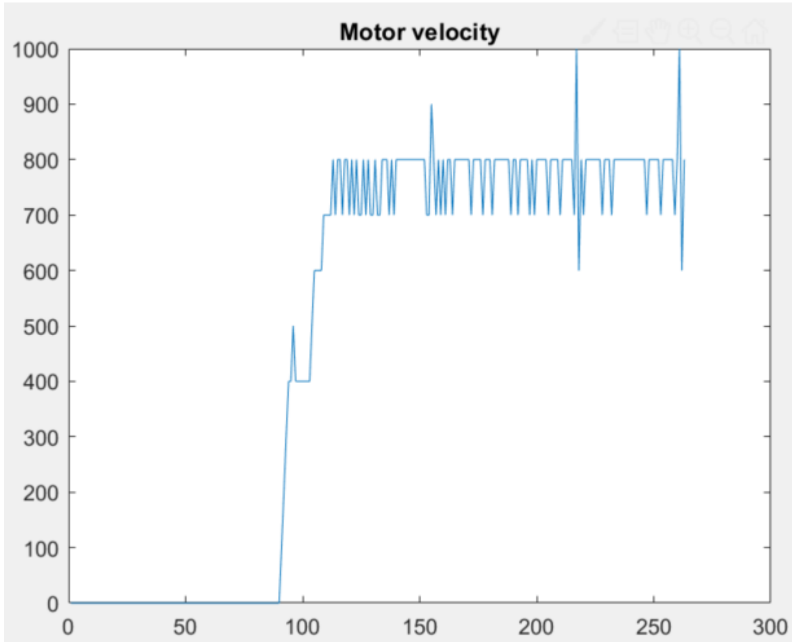


Figur 1: Simulink opstilling af måling

Grafen på Figure 2 repræsenterer motorens rotation i grader. Grafen på Figure 3 repræsenterer den afledte af positionen i grader, altså hastigheden i grader. Som man kan se på Figure 3 opfører den afledte sig som et første ordens system, dog med et lille udfald i accelerationen, men da den er så lille vælger vi at se bort fra denne. Da vores afledte opfører sig som et første ordens system må det betyde at vores system er et 2. ordens system.



Figur 2: Måling af motor vinkel i grader



Figur 3: Måling af motorens hastighed i grader

Vi besluttede os for at gå videre med disse målinger, og benyttede os af "tfest" funktionen i matlab til, ud fra vores data, at estimere en 2. ordens transfer funktionen.

$$\frac{74.34}{s^2 + 8.028s + 0.1365} \tag{1}$$

Vi ønsker at opstille vores system på state space form, da det gør det mulig at evaluere systemet ved hjælp af matricx algebra. State space er specielt brugbart i systemer med flere input og flere outputs, i kontrast til klassisk kontrol metoder, så som PID-regulering, som bygger på komplekse lapace og fourier ligninger som skal transformeres frem og tilbage mellem tids- og frekvensdomænet. En af de helt store fordele ved state space kontrol er dens anvendelighed til et bredt spekter af systemer, det kan både anvendes på linær og ikke linære, tidsafhængelige og tids uafhændelige, single-input single-output og multiple-input og multiple-output systemer. State space bliver repræsenteret ved bare to ligninger. Første ligning, som ses på Equation 2, giver forholdet mellem systemets nuværende state og inputtet til dens næste state. Equation 3 viser at systemets output er afhængig af systemets nuværende state.

$$\dot{x} = A * x(t) + B * u(t) \tag{2}$$

$$y = C * x(t) + D * u(t) \tag{3}$$

Hvor:
x er en vektor af alle state variabler
x' er en vektor af de afledte state variabler
A matrix er state matrixen
B matrix er input matrixen
C matrix er output matrixen

D matrix er feedforward matrixen

Derfor ønsker vi nu at få opstillet vores transfer funktion på state space form, det gør vi med matlab funktionen ”tf2ss”, som transformere vores transfer funktion til state space controllable canonical form. Denne state space repræsentation kan ses på Equation 4 og Equation 5. Vores D matrice er lige 0, og derfor ikke med i vores state space repræsentation.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.137 & -8.28 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} * u \tag{4}$$

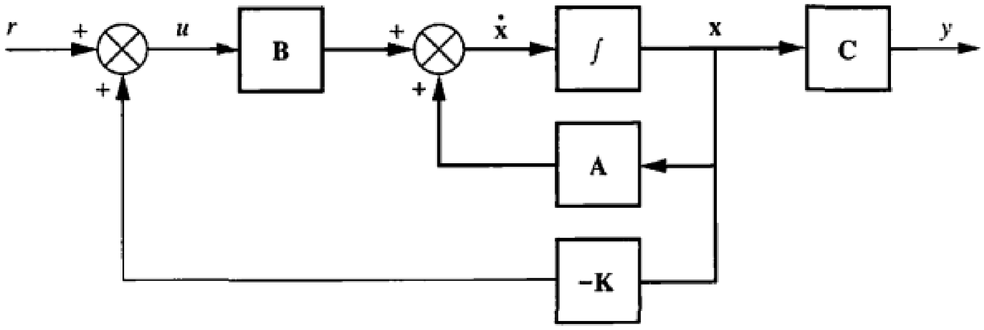
$$y = \begin{bmatrix} 74.340 & 0 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5}$$

3 Pol-placering

For at designe en lukket sløjfe controller for ens system, har vi gjort brug af pol-placerings princippet. Dette gøres ved at først at bestemme hvilken orden ens system har (første, anden osv.). Ud fra hvilken orden ens system har kan man lave en karakteristik ligning for denne. I vores tilfælde er det et anden-ordens system, hvilket for ligningen til at se således ud:

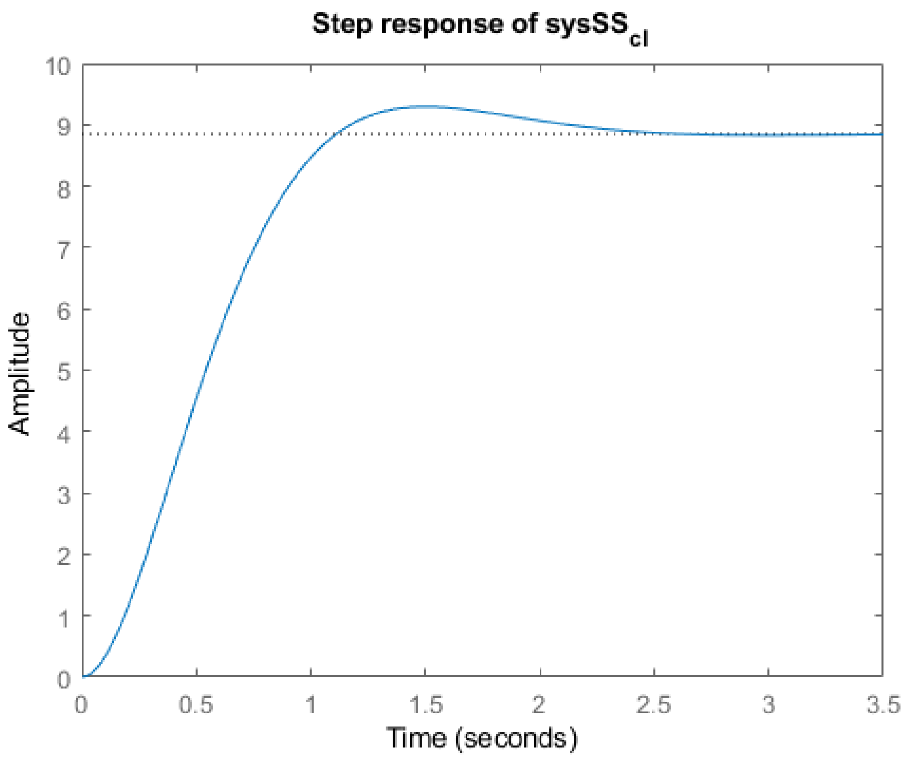
$$\frac{wn^2}{s^2 + 2 * \zeta * wn * s + wn^2} \tag{6}$$

Derefter skal vi bestemme nogle krav til systemet. Vi har valgt at vores overshoot skal ligge på 5% og vores settlingtime til 2 sekunder. Disse krav er forskellige fra system til system og bestemmes af hvad systemet skal bruges til. Når vi har valgt vores settlingtime og overshoot kan vi finde polerne for karakteristikligningen. Disse poler skal bruges til at designe vores controller, så den får den rette karakteristik, altså 5% overshoot og 2 sekunders settlingtime. Nedenunder kan man se blok opbygningen af et system med state-variable feedback.



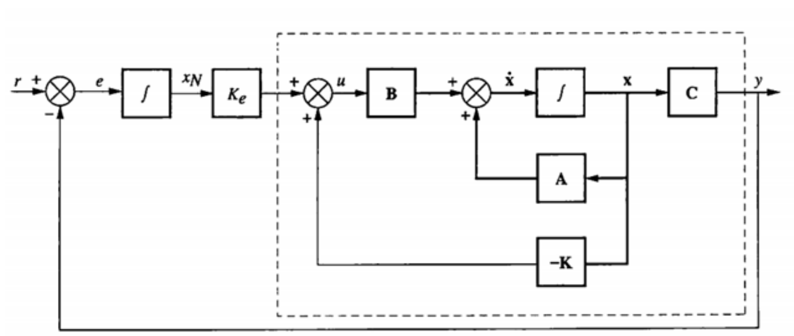
Figur 4: Blok diagram over vores controller

Vi gør brug af matlab funktionen place, som kan beregne vores state feedback matrix, K. Dette gør vi ved at indsætte state space variablerne A og B, som vi fandt frem til i starten og polerne fra vores karakteristik ligning. Dermed får vi beregnet vores feedback matrix så den passer med vores krav. Nu kan vi teste om vores controller fungerer efter hensigten ved at smide et stepinput ind i systemet og kigge på responsen. Dette kan ses nedenfor.



Figur 5: Steprespons for closed loop system

3.1 Steady state error



Figur 6: Matrix når tredje pol indsættes

For at fjerne steady state error skal en tredje pol indsættes

Man kan se på step responset at der er en stor steady state fejl. Denne fejl kan elimineres ved at indsætte endnu en pol i systemet. Denne pol skal gerne ligge så langt væk fra de eksisterende poler at det ikke har nogen indflydelse på systemets karakteristik. Til vores system vælger vi denne tredje pol til at ligge i -30.

Da vi tilføjer endnu en pol og dermed også gør det til et tredje ordens system, bliver vi nødt til at lave om på vores state space matricer A, B, C og D. De skal udvides så de kommer til at se ud som på billedet nedenfor.

$$\boxed{\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{BK}) & \mathbf{B}K_e \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_N \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r}$$

$$\boxed{y = [\mathbf{C} \quad 0] \begin{bmatrix} \mathbf{x} \\ x_N \end{bmatrix}}$$

Figur 7: Matrix når tredje pol indsættes

Denne matrix kan også laves på blokform.

Litteratur