UNIVERSITETET I AGDER

Optimalisering av rørhåndteringskran

Thorbjørn Alvsvåg

Veileder

Morten Kjeld Ebbesen

Masteroppgaven er gjennomført som ledd i utdanningen ved Universitetet i Agder og er godkjent som del av denne utdanningen. Denne godkjenningen innebærer ikke at universitetet innestår for de metoder som er anvendt og de konklusjoner som er trukket.

Universitetet i Agder, 2013

Fakultet for teknologi og realfag

Institutt for ingeniørvitenskap

Oppsummering

Oppgaven tar for seg utviklingen av et matlab program for bruk til konsept-generering av en Pipe Deck Pipe Handler kran. Programmet er designet for å håndtere en analyse av kranens dynamikk. Reaksjonskreftene til kranen i bevegelse danner grunnlaget for en grundigere analyse og evaluering av kranens oppbygning. Kranen er optimalisert med hensyn på kjøretid og mekanisk design.

Kinematikken til kranen er basert på matematiske begrensninger for ledd og drivere. Posisjonsligninger er ulineære slik at disse må løses ved bruk av Newton Raphson metode. Hastighet og akselerasjonsanalyse er basert på Jacobian matrise utviklet fra begrensnings-ligningen. Til slutt er reaksjonskrefter beregnet ved bruk av Lagrangian Multiplier.

Olje og gassindustrien har i stor grad bruk for kompliserte maskiner for å utføre arbeidsoppgaver. Disse maskinene kan variere i utforming etter kundens behov. Dette gjør at utviklingsarbeidet fra kunde til kunde er stort, selv om arbeidsoppgaven til maskinen er relativt lik. Problemet gjør at det er stor interesse i å designe et program for å ta seg av en grunnleggende analyse i tidlig utviklingsfase. En slik databasert analyse vil hjelpe til med å ta beslutninger om design raskere.

Innholdsliste

Symbolforklaring

Notasjoner:

En skalar er representert ved kursiv liten bokstav: a

En vektor er representert ved fet liten bokstav: **a**

En matrise er representert ved fet stor bokstav: **A**

Overstrekning

˙	Førstederivert med hensyn på tid.
¨	Andrederivert med hensyn på tid.
~	Skew matrise

Hevet tekst

‘	Lokale koordinater

Senket tekst

i	nummerert legeme

Forkortelser

PDPH = Pipe Deck Pipe Handler

Figurliste

1 Introduksjon

Innenfor all industri brukes maskiner til ulike formål. Maskinene er designet for å utføre ulike oppgaver, og må ofte skreddersys for å møte kundens behov. Selv om det ofte er små forandringer i bruksmål kan det likevel være store endringer i design. Dette gjør at mange arbeidstimer brukes til å beregne og bestemme seg for ny design av et produkt. Et verktøy som kan bidra til å bestemme design i tidlig fase vil være både økonomisk og tidsmessig besparende. Det er dette som er utgangspunktet for Aker Solutions ønske om å lage et program for deres Pipe Deck Pipe Handler kran. Programmet skal analysere kranens dynamiske reaksjonskrefter. Dette vil gi nyttig kunnskap om kranens krav til design. Basert på den dynamiske analysen optimeres kranen med hensyn på kjørehastighet og mekanisk design.

Et mekanisk system som gjennomgår en analyse kan deles opp i to hovedparter, statisk og dynamisk analyse. I den statiske analysen ser man på et system uten å ta hensyn til hvordan systemet endrer seg over tid. I den dynamiske analysen ser vi på et system i bevegelse og hvordan dette endrer seg over tid. Den dynamiske analysen kan også deles inn i to hovedgrupper, kinematikk og kinetikk. Kinematikk er læren om bevegelse uten tanke for kreftene som forårsaker bevegelsen. Kinetikk er læren om bevegelse med hensyn til krefter. Kinetikken er derfor basert på kinematikk studien (Nikravesh 2008). Gjennom denne rapporten benyttes både kinematikken og kinetikken for å danne en databasert dynamisk analyse av PDPH kranen

I kapittel 2 forklares PDPH kranens funksjon og oppbygging. Kranens anvendte kinematikk beskrives i kapittel 3. Kapittel 4 presenterer posisjonsanalysen av kranen med bindingsligninger for krankinematikken og Newton Raphson metode for å løse disse ligningene. Kapittel 5 og 6 presenterer hastighet og akselerasjonssanalyse av kranen, og introduserer Jacobian matrisen for systemet. Reaksjonskrefter beregnes gjennom bruk av invers Jakobian matrise og Lagranges multiplikator og presenteres i kapittel 7. I kapittel 8 utføres en statisk analyse av kranens reaksjonskrefter for validering av modell. Simuleringsresultater for posisjon, hastighet, akselerasjon samt reaksjonskrefter valideres ved hjelp av figurer i kapittel 9. I kapittel 10 optimeres kranen med hensyn på kjøretid og konstruksjon. Kapittel 11 omhandler en løsning for å initialverdiproblemer for posisjon, og tilslutt oppsumeres oppgaven gjennom en konklusjon i kapittel 12.

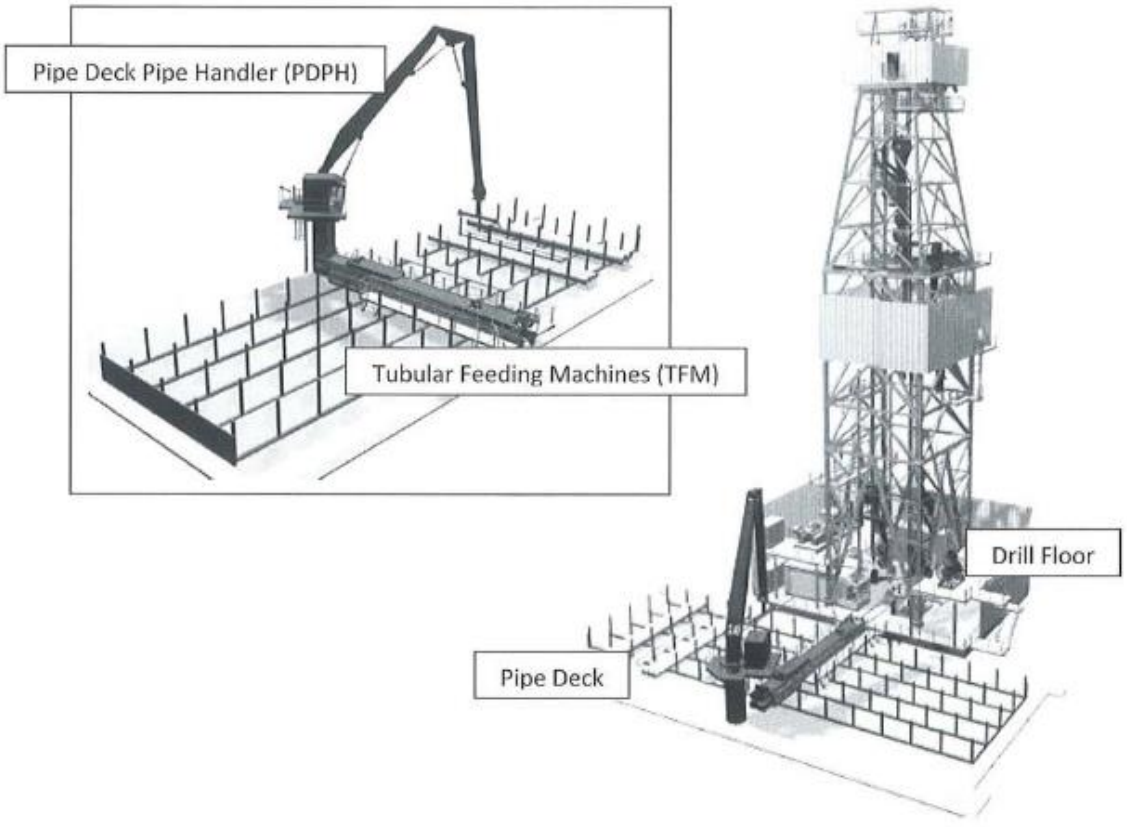
Alle beregninger er gjort i Matlab. Derfor er grunnleggende kunnskap om programmet nødvendig for å lese oppgaven.

1.1 Aker Solutions

Aker Solutions, som er oppdragsgiver for dette prosjektet, er en av verdens ledende leverandører av produkter, systemer og service for oljeindustrien. Deres kunnskap og teknologi strekker seg fra oljereservoaret til produksjon gjennom et helt livsløp for et oljefelt. I dag har selskapet ca. 25 000 ansatte i mer enn 30 land på alle kontinenter. (Solution, 2013)

2 Systembeskrivelse

Den aktuelle PDPH kranen er utviklet og konstruert av Aker Solutions. Dens formål er å transportere "drill pipes" mellom boreriggens "pipe deck" og riggens "tubular feeding machine" som tar seg av den videre transporten. PDPH kranen forsynes med olje gjennom boreriggens "ring line" system. Hydrauliske sylindere og motorer for rotasjon kontrolleres manuelt ved hjelp av fjernkontroll. Figuren under viser PDPH kranen på en oljerigg.

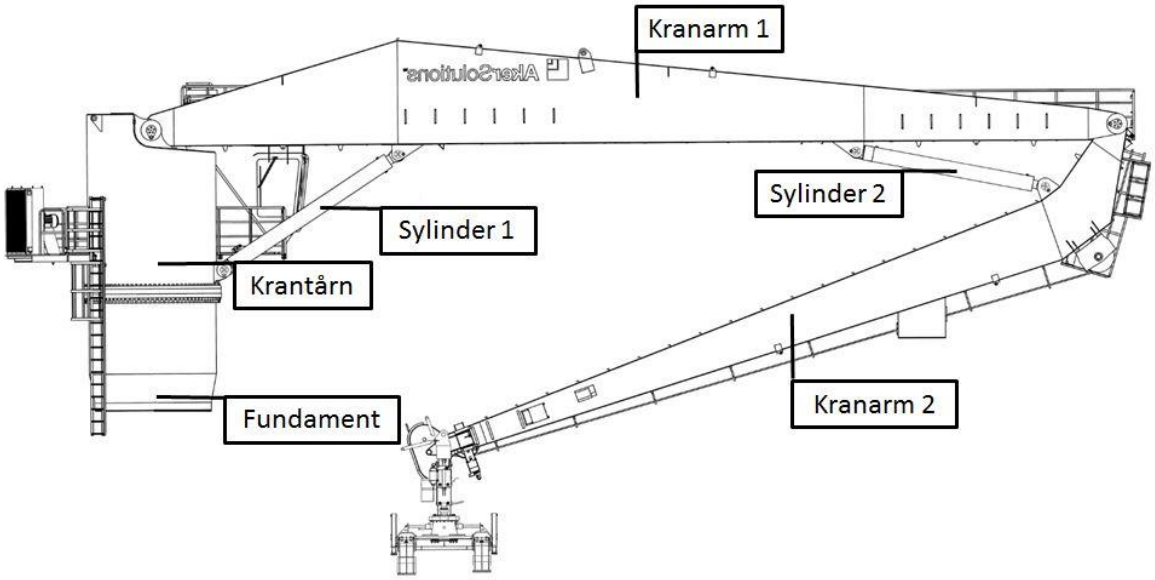


Figur 2.1 Pipe Deck Pipe Handler

PDPH kranen består av følgene hoveddeler:

- Fundament
- Styrehus
- Krantårn
- Kranarm 1
- Kranarm 2
- Sylinder 1
- Sylinder 2

Disse delene er vist i Figur 2.2:



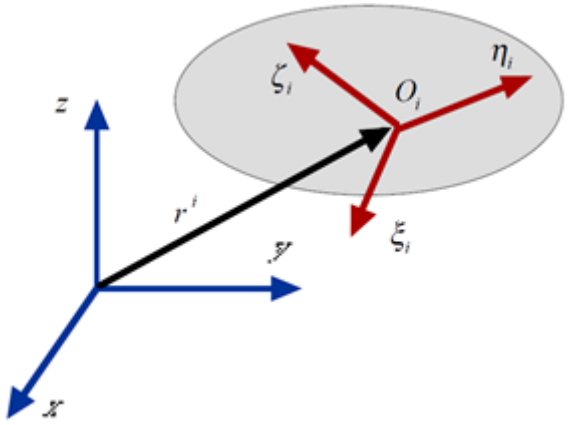
Figur 2.2 Pipe Deck Pipe Handler oversikt

Krantårnet kan rotere om z-aksen og er festet til fundamentet ved hjelp av glidelager. Dette gjør at kranen kan rotere om senteret til fundamentet ved hjelp av hydrauliske motorer og girutveksling. Kranarm 1 og kranarm 2 kan rotere om sine fester og styres ved hjelp av hydrauliske sylindere. Dette gjør kranen fleksibel med et stort operasjonsområde.

3 Kinematikk

I en kinematisk analyse ser vi på et system uten å ta i betraktning kreftene som vil forårsake systemets bevegelse. Et mekanisk system kan ses på som en samling legemer som beveger seg relativt til hverandre (Nikravesh, 2013).

3.1 Translasjon og rotasjon av vektorer



Figur 3.1 Globalt og lokalt koordinatsystem (Nikravesh 2013)

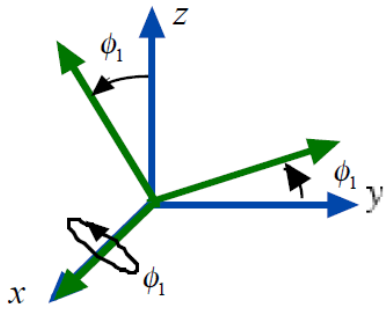
Som Figur 3.1 viser definerer man et stasjonert globalt koordinatsystem (x,y,z), og et lokalt koordinatsystem (ξ, η, ζ) som er låst til legemet. Vektor $\boldsymbol{r}^i$ angir legemets posisjon i globalt koordinatsystem.

$$\boldsymbol{r}^i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (1)$$

Rotasjonsrelasjonen mellom globalt og lokalt koordinatsystem beskrives ved hjelp av Eulers transformasjons matrise **A**. Denne matrisen transformerer vektorers representasjon fra det lokale koordinatsystemet over til det globalt koordinatsystemet. Transformasjonsmatrisen er produktet av tre transformasjoner. Transformasjon rundt koordinatakse x, $\boldsymbol{A_x}$, transformasjon rundt ny koordinatakse y, $\boldsymbol{A_y}$, og til slutt transformasjon rundt ny koordinatakse z, $\boldsymbol{A_z}$,

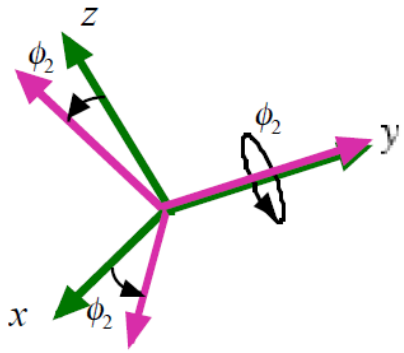
Hvor

$$\boldsymbol{A_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_1) & -\sin(\phi_1) \\ 0 & \sin(\phi_1) & \cos(\phi_1) \end{bmatrix} \quad (2)$$



Figur 3.2 transformasjon om x-aksen (Nikravesh 2013)

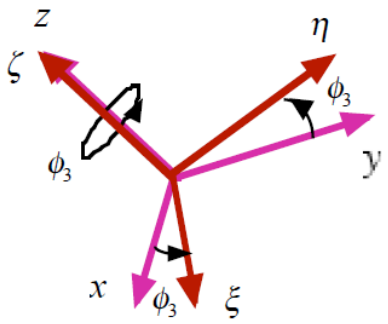
$$\boldsymbol{A_y} = \begin{bmatrix} \cos(\phi_2) & 0 & \sin(\phi_2) \\ 0 & 1 & 0 \\ -\sin(\phi_2) & 0 & \cos(\phi_2) \end{bmatrix} \quad (3)$$



Figur 3.3 transformasjon om y-aksen (Nikravesh 2013)

(4)

$$\boldsymbol{A_z} = \begin{bmatrix} \cos(\phi_3) & -\sin(\phi_3) & 0 \\ \sin(\phi_3) & \cos(\phi_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Figur 3.4 transformasjon om z-aksen (Nikravesh 2013)

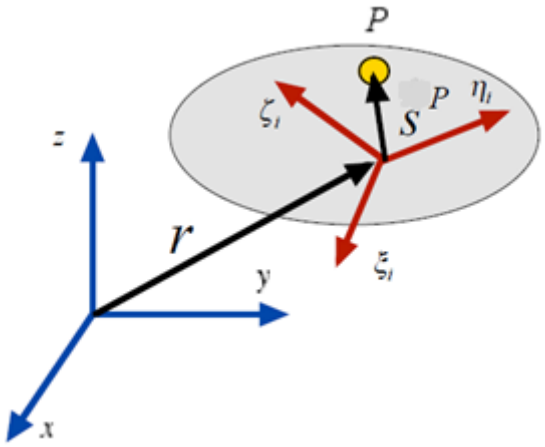
Produktet av transformasjonsmatrisene gir **A** matrisen.

$$\boldsymbol{A} = \boldsymbol{A_z A_y A_x} = \begin{bmatrix} \mathrm{c}\phi_2 c\phi_3 & -\mathrm{c}\phi_2 s\phi_3 & s\phi_2 \\ \mathrm{c}\phi_1 \mathrm{s}\phi_3 + \mathrm{s}\phi_1 \mathrm{s}\phi_2 \mathrm{c}\phi_3 & \mathrm{c}\phi_1 \mathrm{c}\phi_3 - \mathrm{s}\phi_1 \mathrm{s}\phi_2 \mathrm{s}\phi_3 & -\mathrm{s}\phi_1 \mathrm{c}\phi_2 \\ \mathrm{s}\phi_1 \mathrm{s}\phi_3 - \mathrm{c}\phi_1 \mathrm{s}\phi_2 \mathrm{c}\phi_3 & \mathrm{s}\phi_1 \mathrm{c}\phi_3 + \mathrm{c}\phi_1 \mathrm{s}\phi_2 \mathrm{s}\phi_3 & \mathrm{c}\phi_1 \mathrm{c}\phi_2 \end{bmatrix} \quad (5)$$

Hvor:

c = Cosinus.

s = Sinus.



Figur 3.5 Vektorbeskrivelse til punkt P for legemet (Nikravesh 2013).

Figur 3.5 viser en vektor, $\vec{S}^P$ som peker ut fra det lokale koordinatsystemet. Avstander oppgis som vektorer representert i det lokale koordinatsystem og gjør at vektorkoordinat for legemer er konstante. Dette er fordi det lokale koordinatsystemet er fiksert til legemet, og alltid følger dets rotasjon og translasjon. En definert vektor i det lokale koordinatsystemet kan representeres i globalt koordinatsystem ved hjelp av transformasjonsmatrisen, A.

$$\boldsymbol{s^P = As^{P'}} \tag{6}$$

Merket vektor er vektor beskrevet i lokalt koordinatsystem. Vektoren, $\boldsymbol{r^P}$, som leder til punktet P kan defineres:

$$\boldsymbol{r^P = r + As^{P'}} \tag{7}$$

Det er også av interesse å studere vektorens hastighet og akselerasjon. Dette gjøres ved å derivere posisjonsligningen. Den tidsderiverte transformasjonsmatrise defineres:

$$\boldsymbol{\dot{A} = A\tilde{\omega}'} \tag{8}$$

Hvor $\boldsymbol{\tilde{\omega}'}$ er skewmatrisen til vinkelhastighetene $\boldsymbol{\omega'}$. (Nikravesh 2013)

$$\dot{\boldsymbol{\varphi}}' = \boldsymbol{\omega}' = \begin{bmatrix} \omega_{\xi_i} \\ \omega_{\eta_i} \\ \omega_{\zeta_i} \end{bmatrix} \qquad \tilde{\boldsymbol{\omega}}' = \begin{bmatrix} 0 & -\omega_{\zeta_i} & \omega_{\eta_i} \\ \omega_{\zeta_i} & 0 & -\omega_{\xi_i} \\ -\omega_{\eta_i} & \omega_{\xi_i} & 0 \end{bmatrix} \tag{9}$$

Vektor, $\boldsymbol{r^P}$, derivert med hensyn på tid gir translasjonshastigheten til punktet, P, i forhold til globalt koordinatsystem.

$$\dot{\boldsymbol{r}}^{\boldsymbol{P}} = \dot{\boldsymbol{r}} + \dot{\boldsymbol{A}}\boldsymbol{s}^{\boldsymbol{P}'} + \boldsymbol{A}\dot{\boldsymbol{s}}^{\boldsymbol{P}'} = \dot{\boldsymbol{r}} + \boldsymbol{A}\tilde{\boldsymbol{\omega}}'\boldsymbol{s}^{\boldsymbol{P}'} = \dot{\boldsymbol{r}} - \boldsymbol{A}\widetilde{\boldsymbol{s}^{\boldsymbol{P}'}}\boldsymbol{\omega}' \tag{10}$$

Vektor $\boldsymbol{s}^{\boldsymbol{P}'}$ er konstant, derfor faller leddet bort ved derivasjon. Hastighetsvektoren, $\dot{\boldsymbol{r}}^{\boldsymbol{P}}$, derivert med hensyn på tid gir akselerasjonen til punktet P.

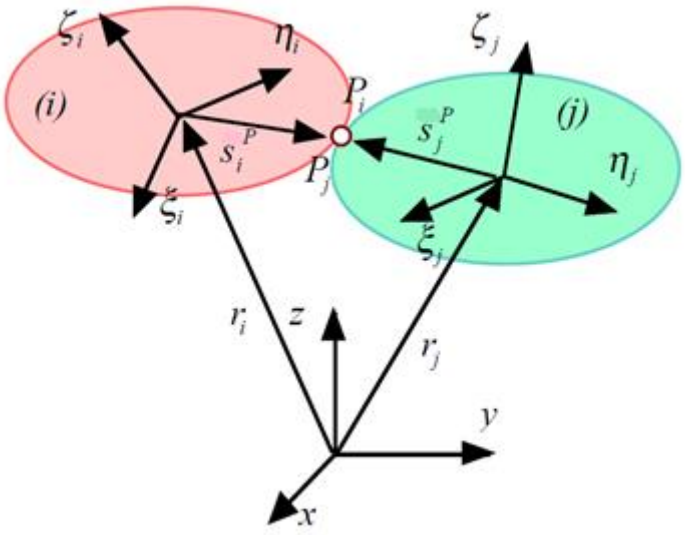
$$\ddot{\boldsymbol{r}}^{\boldsymbol{P}} = \ddot{\boldsymbol{r}} + \boldsymbol{A}\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\boldsymbol{s}^{\boldsymbol{P}\prime} + \boldsymbol{A}\dot{\tilde{\boldsymbol{\omega}}}'\boldsymbol{s}^{\boldsymbol{P}\prime} = \ddot{\boldsymbol{r}} + \boldsymbol{A}\tilde{\boldsymbol{\omega}}'\tilde{\boldsymbol{\omega}}'\boldsymbol{s}^{\boldsymbol{P}\prime} - \boldsymbol{A}\widetilde{\boldsymbol{s}^{\boldsymbol{P}'}}\dot{\boldsymbol{\omega}}' \tag{11}$$

3.2 Ledd og drivere for PDPH kranen

Kranens mekaniske ledd og drivere beskrives matematisk i den kinematiske modellen av kranen. Et fritt legeme i 3D har 6 frihetsgrader (DOF). Disse representerer legemets translasjon og rotasjon. Et mekanisk system beveger seg i henhold til sine ledd og drivere. Dette betyr at legemene er bundet til hverandre. Et legemes dimensjon er beskrevet ved hjelp av vektorkoordinat. Bindingene mellom hvert legeme er basert på sammenhengen mellom legemers vektorkoordinat. En komplett kinematisk beskrivelse av et system skal ha like mange bindinger som antall DOF (Nikravesh 2008). En kinematisk binding betegnes med symbolet Φ. De matematiske ligningene for hver binding er avhengig av hva type ledd eller driver man beskriver. Kranens matematiske ledd og drivere er beskrevet under.

3.2.1 Rotasjonsledd

For å beskrive kranens rotasjonsledd matematisk, benyttes en sfærisk binding sammen med restriksjoner for rotasjon om bestemte akser.



Figur 3.6 Vektorbeskrivelse sfærisk binding (Nikravesh 2013)

Figur 3.6 viser definerte vektorer i en sfærisk binding. Vektorsummen fra det globale koordinatsystemet, via det lokale koordinatsystemet til bindingen for hvert av de to legemene må være like stor. Ligningen binder 3 frihetsgrader.

$$\boldsymbol{\Phi} = \boldsymbol{r_i} + \boldsymbol{s_i}^{\boldsymbol{P}} - \boldsymbol{r_j} - \boldsymbol{s_j}^{\boldsymbol{P}} = \boldsymbol{r_i} + \boldsymbol{A}\boldsymbol{s_i}^{\boldsymbol{P'}} - \boldsymbol{r_j} - \boldsymbol{A}\boldsymbol{s_j}^{\boldsymbol{P'}} = 0 \tag{12}$$

Denne bindingen binder sammen de to legemene i punktet P. Legemene kan fremdeles rotere uavhengig av hverandre, men er bundet til å følge hverandres translasjon i punktet.

Legemers rotasjon i forhold til hverandre kan stoppes ved å låse koordinat-akser mot hverandre. Man etablerer enhetsvektorene $\boldsymbol{u}_\xi, \boldsymbol{u}_\eta$ og $\boldsymbol{u}_\zeta$, som følger retningen til legemets lokale koordinatakse.

$$\boldsymbol{u}_\xi = \begin{bmatrix}1\\0\\0\end{bmatrix}, \boldsymbol{u}_\eta = \begin{bmatrix}0\\1\\0\end{bmatrix}, \boldsymbol{u}_\zeta = \begin{bmatrix}0\\0\\1\end{bmatrix} \tag{13}$$

Man utnytter at prikkproduktet mellom to parallelle vektorer er null. Hver ligning binder en rotasjonsfrihet (Nikravesh 2013).

$$\Phi = (\boldsymbol{A_i}\boldsymbol{u}_\xi)^T(\boldsymbol{A_j}\boldsymbol{u}_\eta) = 0 \tag{14}$$

$$\Phi = (\boldsymbol{A_i}\boldsymbol{u}_\xi)^T(\boldsymbol{A_j}\boldsymbol{u}_\zeta) = 0 \tag{15}$$

$$\Phi = (\boldsymbol{A_i}\boldsymbol{u}_\eta)^T(\boldsymbol{A_j}\boldsymbol{u}_\zeta) = 0 \tag{16}$$

3.2.2 Rotasjonsdriver

Rotasjonsdriveren styrer legemers rotasjon. Bindingsligningen for en rotasjonsdriver er en funksjon av legemers relative vinkel samt driverfunksjonen d(t). Det vil si at et legemes vinkelposisjon er avhengig av dens tilknyttede legemes vinkelposisjon. Driveren d(t) styrer krantårnets rotasjon om z-aksen (Agder).

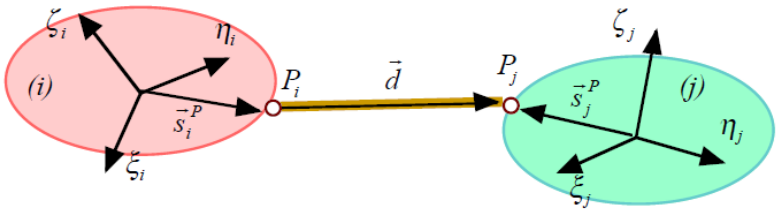
$$\Phi = \varphi_{i,z} - \varphi_{j,z} - d(t) = 0 \tag{17}$$

Rotasjonsdriveren deriveres med hensyn på tid for å beskrive hastighet og akselerasjon.

$$\dot{\Phi} = \ddot{\varphi}_{i,z} - \ddot{\varphi}_{j,z} - \dot{d}(t) = 0$$

$$\ddot{\Phi} = \ddot{\varphi}_{i,z} - \ddot{\varphi}_{j,z} - \ddot{d}(t) = 0$$

3.2.3 Longitudinal driver



Figur 3.7 Vektorbeskrivelse longitudinal driver (Nikravesh 2013)

Kranens kranarmer styres ved hjelp av hydrauliske sylindre. Disse longitudinale driverne gir translasjon mellom legemer. Figur 3.7 viser definerte vektorer for en longitudinal driver. Bindingsligningen for en longitudinal driver er som følger:

$$\Phi = \boldsymbol{d}^T \boldsymbol{d} - l(t)^2 = 0 \tag{18}$$

Hvor **d** er vektorlengden mellom festene for driverne. $l(t)$ gir endring i vektorlengde som funksjon av tid (Nikravesh, 2013).

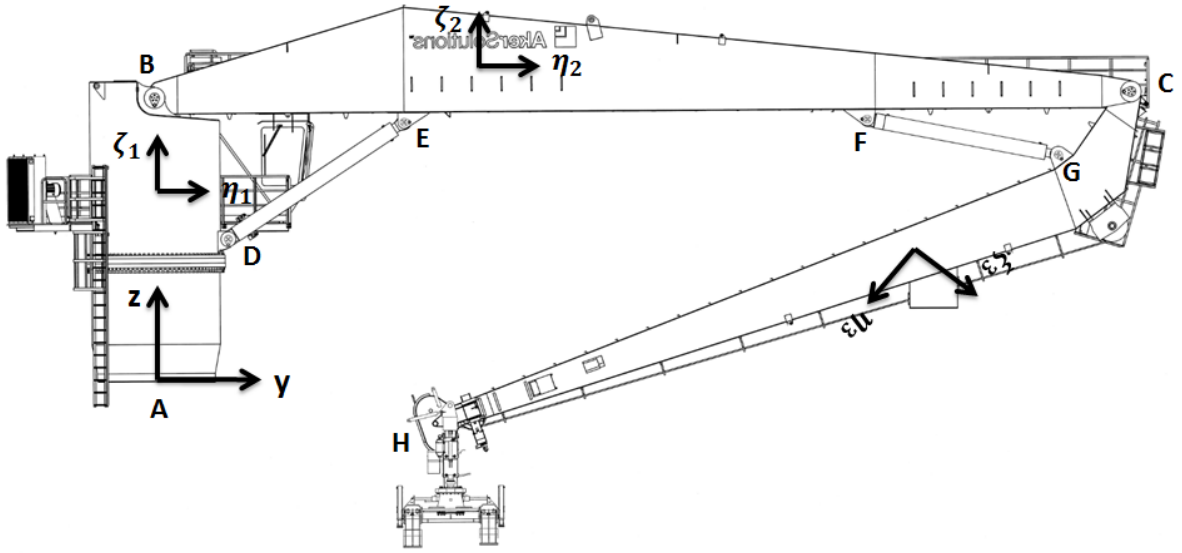
Hastigheten til driveren:

$$\dot{\Phi} = 2d^T\dot{d} - 2l(t)\dot{l(t)} = 0$$

Akselerasjonen til driveren:

$$\ddot{\Phi} = 2\dot{d^T}\dot{d} + 2d^T\ddot{d} - 2\dot{l(t)}\dot{l(t)} - 2l(t)\ddot{l(t)} = 0$$

4 Posisjonsanalyse



Figur 4.1 PDPH kran med markert globalt koordinatsystem og lokale koordinatsystem.

Ved å bruke bindingsligningene som definert over kan man sette opp likningene for hele kranen. Kranen består av tre legemer, Rotating part, Inner jib og Outher jib. Figur 4.1 viser globalt vektorsystem (x,y,z) og lokale vektorsystem (ξ_i, η_i, ζ_i) . Legemene er bundet til bakken, og til hverandre, ved hjelp av rotasjonsbindinger i punkt A, B og C. I binding A styres kranen ved hjelp av en rotasjonsdriver. Kranen styres også ved hjelp av to hydrauliske sylindere (longitudinal driver) mellom punkt D-E og F-G

Kinematiske bindinger:

Rotasjonsledd i punkt A mellom fundament og krantårn. Låser koordinatakse ζ_1 for rotasjon mot global x-koordinat, samt rotasjon mellom koordinatakse ζ_1 og global y-koordinat. 5 frihetsgrader.

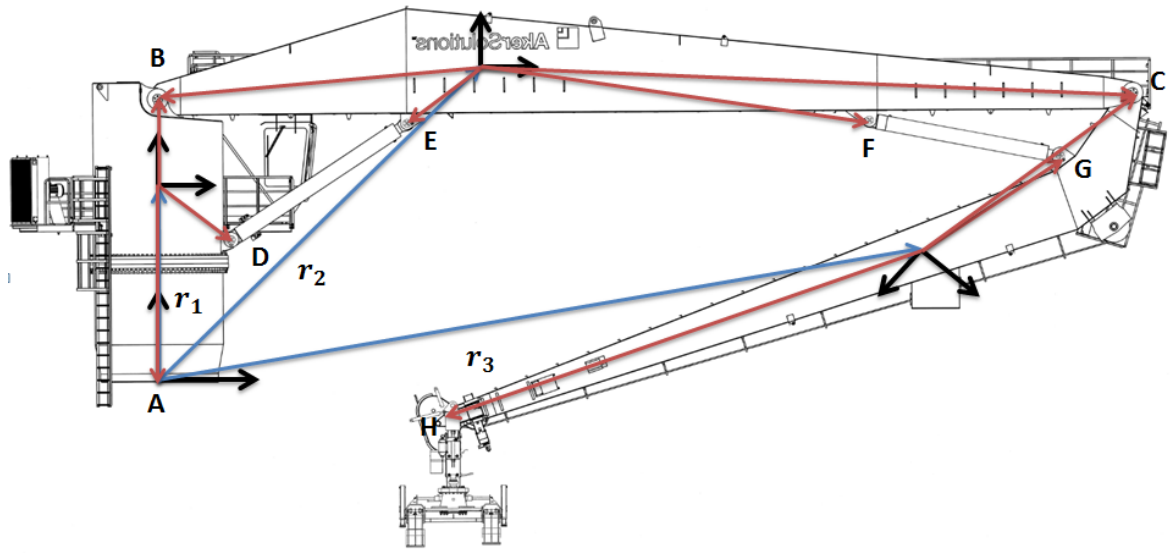
Rotasjonsledd i punkt B mellom krantårn og kranarm 1. Låser koordinatakse ξ_1 for rotasjon mot koordinatakse η_2, samt rotasjon mellom koordinatakse $\xi_1$og koordinatakse ζ_2. 5 frihetsgrader.

Rotasjonsledd i punkt C mellom kranarm 1 og kranarm2. Låser koordinatakse ξ_2 for rotasjon mot koordinatakse η_3, samt rotasjon mellom koordinatakse ξ_2 og koordinatakse ζ_3. 5 frihetsgrader.

Rotasjonsdriver i punkt A. 1 frihetsgrad.

Longitudinal driver mellom punkt D og E. 1 frihetsgrad.

Longitudinal driver mellom punkt D og E. 1 frihetsgrad.



Figur 4.2 Definerte vektorkoordinat for PDPH kranen

Figur 4.2 viser de definerte vektorene for systemet. De blå vektorene, $\boldsymbol{r_1}$, $\boldsymbol{r_2}$ og $\boldsymbol{r_3}$ er posisjons vektorer til hvert av legemene i det globale koordinatsystemet. De røde vektorene definerer posisjonen til bindingene. Vektorlengder bestemmes ut fra tekniske tegninger. For å ha god kontroll over vektorlengdene settes de opp i en samlet matrise i Matlab. Dette gjør at en enkelt kan endre verdier senere.

$\boldsymbol{s_1}^{\boldsymbol{A\prime}}$ = Lengde fra lokalt koordinatsystem 1 til binding A.

$\boldsymbol{s_1}^{\boldsymbol{B\prime}}$ = Lengde fra lokalt koordinatsystem 1 til binding B.

$\boldsymbol{s_1}^{\boldsymbol{D\prime}}$ = Lengde fra lokalt koordinatsystem 1 til binding D.

$\boldsymbol{s_2}^{\boldsymbol{B\prime}}$ = Lengde fra lokalt koordinatsystem 2 til binding B

$\boldsymbol{s_2}^{\boldsymbol{C\prime}}$ = Lengde fra lokalt koordinatsystem 2 til binding C.

$\boldsymbol{s_2}^{\boldsymbol{E\prime}}$ = Lengde fra lokalt koordinatsystem 2 til binding E.

$\boldsymbol{s_2}^{\boldsymbol{F\prime}}$ = Lengde fra lokalt koordinatsystem 2 til binding F.

$\boldsymbol{s_3}^{\boldsymbol{C\prime}}$ = Lengde fra lokalt koordinatsystem 3 til binding C.

$\boldsymbol{s_3}^{\boldsymbol{G\prime}}$ = Lengde fra lokalt koordinatsystem 3 til binding G.

Ut fra kartleggingen av kranen kan man sette opp Bindingsligningene ($\boldsymbol{\Phi}$).

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{r_1} + \boldsymbol{A_1 s_1}^{\boldsymbol{A\prime}} \\ \boldsymbol{u_x}^{\boldsymbol{T}}(\boldsymbol{A_1 u_{\zeta_1}}) \\ \boldsymbol{u_y}^{\boldsymbol{T}}(\boldsymbol{A_1 u_{\zeta_1}}) \\ \boldsymbol{r_1} + \boldsymbol{A_1 s_1}^{\boldsymbol{B\prime}} - \boldsymbol{r_2} - \boldsymbol{A_2 s_2}^{\boldsymbol{B\prime}} \\ (\boldsymbol{A_1 u_{\xi_1}})^{\boldsymbol{T}}(\boldsymbol{A_2 u_{\zeta_2}}) \\ (\boldsymbol{A_1 u_{\xi_1}})^{\boldsymbol{T}}(\boldsymbol{A_2 u_{\eta_2}}) \\ \boldsymbol{r_2} + \boldsymbol{A_2 s_2}^{\boldsymbol{C\prime}} - \boldsymbol{r_3} - \boldsymbol{A_3 s_3}^{\boldsymbol{C\prime}} \\ (\boldsymbol{A_2 u_{\xi_2}})^{\boldsymbol{T}}(\boldsymbol{A_3 u_{\zeta_3}}) \\ (\boldsymbol{A_2 u_{\xi_2}})^{\boldsymbol{T}}(\boldsymbol{A_3 u_{\eta_3}}) \\ \varphi_{1,z} - d(t) \\ \left(\boldsymbol{d_1}^{\boldsymbol{T}}\boldsymbol{d_1}\right) - l_1(t)^2 \\ \left(\boldsymbol{d_2}^{\boldsymbol{T}}\boldsymbol{d_2}\right) - l_2(t)^2 \end{bmatrix} = 0 \tag{19}$$

$\boldsymbol{d_1}$ er vektor mellom punkt D og E for sylinder 1.

$$\boldsymbol{d_1} = \boldsymbol{r_1} + \boldsymbol{A_1 s_1}^{\boldsymbol{D\prime}} - \boldsymbol{r_2} - \boldsymbol{A_2 s_2}^{\boldsymbol{E\prime}} \tag{20}$$

$\boldsymbol{d_2}$ er vektor mellom punkt F og G for sylinder 2.

$$\boldsymbol{d_2} = \boldsymbol{r_2} + \boldsymbol{A_2 s_2}^{\boldsymbol{F\prime}} - \boldsymbol{r_3} - \boldsymbol{A_3 s_3}^{\boldsymbol{G\prime}} \tag{21}$$

4.1 Newton Raphson

Posisjonsligningene er ulineære. Dette gjør at Newton Raphson metode brukes for å løse disse ligningene. Metoden baserer seg på å finne røttene til ligningen numerisk. Utgangspunktet for å finne en løsning av ligningen er kvalifisert tipping av start posisjonen (**q**) for kranen. Ved hjelp av startverdiene vil Newton Raphson metode konvergere mot de eksakte løsningene. Etter hvert som posisjonen endrer seg over tid finner Newton Raphson nye verdier for posisjon ved hjelp av gammel verdi. Det vil si at for hvert tidssteg i simuleringen vil verdiene for posisjon konvergere mot ny korrekt verdi. Ny verdi ($\boldsymbol{q}^{k+1}$), er lik gammel verdi ($\boldsymbol{q}^{k}$), minus korreksjonsfaktor ($\Delta\boldsymbol{q}$). Fordi $\Delta\boldsymbol{q}$

konvergerer mot 0 settes det en nedre toleranse verdi for godkjenning av Δ i datasimulering (Ben-Israel 1966).

$$\boldsymbol{q}^{k+1} = \boldsymbol{q}^{k} - \Delta\boldsymbol{q} \tag{22}$$

Hvor:

$$\Delta\boldsymbol{q} = \boldsymbol{\Phi_q}^{-\mathbf{1}} \tag{23}$$

5 Hastighetsanalyse

For å finne hastighet og akselerasjon for det mekaniske systemet gjør en nytte av Jacobimatrisen. I motsetning til posisjonsanalysen er løsningen av ligninger for hastighet og akselerasjon lineær, og kan løses direkte ved hjelp av Jacobimatrisen. Jacobimatrisen $\boldsymbol{\Phi_q}$, er definert som den partiellderiverte av bindingsligningen med hensyn på posisjonsvektor $\boldsymbol{q}$.

$$\boldsymbol{\Phi_q} = \left[\frac{\partial \boldsymbol{\Phi}}{\partial \boldsymbol{q}}\right] \tag{24}$$

Bindingsligning for hastighet er som følger:

$$\dot{\boldsymbol{\Phi}}(\boldsymbol{q}, t) = \boldsymbol{\Phi_q}\dot{\boldsymbol{q}} + \boldsymbol{\Phi}_t = 0 \tag{25}$$

Hvor:

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\boldsymbol{r}}_{\boldsymbol{i}} \\ \boldsymbol{\omega_i}' \end{bmatrix} \tag{26}$$

Vektoren $\dot{\boldsymbol{q}}$ består av translasjonshastighet og rotasjonshastighet for hvert legeme i systemet. Den tidsderiverte matriseligningen for kranen er:

$$\dot{\boldsymbol{\Phi}} = \begin{bmatrix} \dot{\boldsymbol{r}}_{\boldsymbol{1}} - \boldsymbol{A_1}\widetilde{\boldsymbol{s_1}^{A'}}\boldsymbol{\omega_1}' \\ -\boldsymbol{u_x}^{\boldsymbol{T}}(\boldsymbol{A_1}\widetilde{\boldsymbol{u}_{\zeta_1}}\boldsymbol{\omega_1}') \\ -\boldsymbol{u_y}^{\boldsymbol{T}}(\boldsymbol{A_1}\widetilde{\boldsymbol{u}_{\zeta_1}}\boldsymbol{\omega_1}') \\ \dot{\boldsymbol{r}}_{\boldsymbol{1}} - \boldsymbol{A_1}\widetilde{\boldsymbol{s_1}^{B'}}\boldsymbol{\omega_1'} - \dot{\boldsymbol{r}}_{\boldsymbol{2}} + \boldsymbol{A_2}\widetilde{\boldsymbol{s_2}^{B'}}\boldsymbol{\omega_2}' \\ -\left(\boldsymbol{A_1}\widetilde{\boldsymbol{u}_{\xi_1}}\boldsymbol{\omega_1'}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_2}\boldsymbol{u}_{\zeta_2}\right) - \left(\boldsymbol{A_1}\boldsymbol{u}_{\xi_1}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_2}\widetilde{\boldsymbol{u}_{\zeta_2}}\boldsymbol{\omega_2}'\right) \\ -\left(\boldsymbol{A_1}\widetilde{\boldsymbol{u}_{\xi_1}}\boldsymbol{\omega_1'}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_2}\boldsymbol{u}_{\eta_2}\right) - \left(\boldsymbol{A_1}\boldsymbol{u}_{\xi_1}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_2}\widetilde{\boldsymbol{u}_{\eta_2}}\boldsymbol{\omega_2}'\right) \\ \dot{\boldsymbol{r}}_{\boldsymbol{2}} - \boldsymbol{A_2}\widetilde{\boldsymbol{s_2}^{C'}}\boldsymbol{\omega_2'} - \dot{\boldsymbol{r}}_{\boldsymbol{3}} + \boldsymbol{A_3}\widetilde{\boldsymbol{s_3}^{C'}}\boldsymbol{\omega_3}' \\ -\left(\boldsymbol{A_2}\widetilde{\boldsymbol{u}_{\xi_2}}\boldsymbol{\omega_2'}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_3}\boldsymbol{u}_{\zeta_3}\right) - \left(\boldsymbol{A_2}\boldsymbol{u}_{\xi_2}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_3}\widetilde{\boldsymbol{u}_{\zeta_3}}\boldsymbol{\omega_3}'\right) \\ -\left(\boldsymbol{A_2}\widetilde{\boldsymbol{u}_{\xi_2}}\boldsymbol{\omega_2'}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_3}\boldsymbol{u}_{\eta_3}\right) - \left(\boldsymbol{A_2}\boldsymbol{u}_{\xi_2}\right)^{\boldsymbol{T}}\left(\boldsymbol{A_3}\widetilde{\boldsymbol{u}_{\eta_3}}\boldsymbol{\omega_3}'\right) \\ \dot{\varphi}_{\eta_1} - 4 \\ \left(2\boldsymbol{d_1}^{\boldsymbol{T}}\dot{\boldsymbol{d}}_{\boldsymbol{1}}\right) - 2l_1(t)\dot{l_1}(t) \\ \left(2\boldsymbol{d_2}^{\boldsymbol{T}}\dot{\boldsymbol{d}}_{\boldsymbol{2}}\right) - 2l_2(t)\dot{l_2}(t) \end{bmatrix} = 0 \tag{27}$$

Hvor:

$$\dot{\boldsymbol{d}}_{\boldsymbol{1}} = \dot{\boldsymbol{r}}_{\boldsymbol{1}} + \boldsymbol{A_1}\widetilde{\boldsymbol{s_1}^{D'}}\boldsymbol{\omega_1'} - \dot{\boldsymbol{r}}_{\boldsymbol{2}} + \boldsymbol{A_2}\widetilde{\boldsymbol{s_2}^{E'}}\boldsymbol{\omega_2'} \tag{28}$$

og

$$\dot{\boldsymbol{d}}_{\boldsymbol{2}} = \dot{\boldsymbol{r}}_{\boldsymbol{1}} + \boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{s}_{\boldsymbol{2}}^{\boldsymbol{F'}}}\boldsymbol{\omega}_{\boldsymbol{2}}' - \dot{\boldsymbol{r}}_{\boldsymbol{3}} + \boldsymbol{A}_{\boldsymbol{3}}\widetilde{\boldsymbol{s}_{\boldsymbol{3}}^{\boldsymbol{G'}}}\boldsymbol{\omega}_{\boldsymbol{3}}' \tag{29}$$

Jacobimatrisen, $\boldsymbol{\Phi}_{\boldsymbol{q}}$ er:

$$\boldsymbol{\Phi}_{\boldsymbol{q}} = \begin{bmatrix}
\boldsymbol{I}_{3x3} & -\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{A'}}} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \\
\boldsymbol{0}_{1x3} & -\boldsymbol{u_x}^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{u}_{\zeta_1}}) & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} \\
\boldsymbol{0}_{1x3} & -\boldsymbol{u_y}^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{u}_{\zeta_1}}) & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} \\
\boldsymbol{I}_{3x3} & -\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{B'}}} & -\boldsymbol{I}_{3x3} & \boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{s}_{\boldsymbol{2}}^{\boldsymbol{B'}}} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \\
\boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{u}_{\xi_1}})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{2}}\boldsymbol{u}_{\eta_2}) & \boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{1}}\boldsymbol{u}_{\xi_1})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{u}_{\eta_2}}) & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} \\
\boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{u}_{\xi_1}})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{2}}\boldsymbol{u}_{\zeta_2}) & \boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{1}}\boldsymbol{u}_{\xi_1})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{u}_{\zeta_2}}) & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} \\
\boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{I}_{3x3} & -\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{s}_{\boldsymbol{2}}^{\boldsymbol{C'}}} & -\boldsymbol{I}_{3x3} & \boldsymbol{A}_{\boldsymbol{3}}\widetilde{\boldsymbol{s}_{\boldsymbol{3}}^{\boldsymbol{C'}}} \\
\boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{u}_{\xi_2}})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{3}}\boldsymbol{u}_{\zeta_3}) & \boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{2}}\boldsymbol{u}_{\xi_2})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{3}}\widetilde{\boldsymbol{u}_{\zeta_3}}) \\
\boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{u}_{\xi_2}})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{3}}\boldsymbol{u}_{\eta_3}) & \boldsymbol{0}_{1x3} & -(\boldsymbol{A}_{\boldsymbol{2}}\boldsymbol{u}_{\xi_2})^{\boldsymbol{T}}(\boldsymbol{A}_{\boldsymbol{3}}\widetilde{\boldsymbol{u}_{\eta_3}}) \\
\boldsymbol{0}_{1x3} & 0 \quad 0 \quad 1 & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} \\
2\boldsymbol{d_1}^{\boldsymbol{T}} & -2\boldsymbol{d_1}^{\boldsymbol{T}}\boldsymbol{A}_{\boldsymbol{1}}\widetilde{\boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{D'}}} & -2\boldsymbol{d_1}^{\boldsymbol{T}} & 2\boldsymbol{d_1}^{\boldsymbol{T}}\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{s}_{\boldsymbol{2}}^{\boldsymbol{E'}}} & \boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} \\
\boldsymbol{0}_{1x3} & \boldsymbol{0}_{1x3} & 2\boldsymbol{d_2}^{\boldsymbol{T}} & -2\boldsymbol{d_2}^{\boldsymbol{T}}\boldsymbol{A}_{\boldsymbol{2}}\widetilde{\boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{F'}}} & -2\boldsymbol{d_2}^{\boldsymbol{T}} & 2\boldsymbol{d_2}^{\boldsymbol{T}}\boldsymbol{A}_{\boldsymbol{3}}\widetilde{\boldsymbol{s}_{\boldsymbol{3}}^{\boldsymbol{G'}}}
\end{bmatrix} \tag{30}$$

Translasjon og rotasjons uavhengige ledd $\boldsymbol{\Phi}_t$ settes utenfor matrise multiplikasjonen og adderes som vektor.

$$\boldsymbol{\Phi}_t = \begin{bmatrix}
0_{1x3} \\ 0_{1x3} \\ 0_{1x3} \\ 0_{1x3} \\ 0_{1x3} \\ -4 \\ -2l_1(t)\dot{l}_1(t) \\ -2l_2(t)\dot{l}_2(t)
\end{bmatrix} \tag{31}$$

Man finner vektor $\dot{\boldsymbol{q}}$ ved hjelp av invers Jacobimatrise (Nikravesh, 2008).

$$\dot{\boldsymbol{q}} = \boldsymbol{\Phi_q}^{-1}(-\boldsymbol{\Phi}_t) \tag{32}$$

6 Akselerasjons analyse

På akselerasjonsnivå vil Jacobimatrisen være identisk lik den man har for hastighet. Bindingsligningen for akselerasjon har derimot fått flere akselerasjonsuavhengige ledd $\boldsymbol{\Phi}_t$. Bindingsligning for akselerasjon er:

$$\ddot{\boldsymbol{\Phi}}(\boldsymbol{q},t) = \boldsymbol{\Phi}_{\boldsymbol{q}}\ddot{\boldsymbol{q}} + \boldsymbol{\gamma} = 0 \tag{33}$$

Hvor

$$\ddot{\boldsymbol{q}} = \begin{bmatrix} \ddot{\boldsymbol{r}}_{\boldsymbol{i}} \\ \dot{\boldsymbol{\omega}}_{\boldsymbol{i}}{}' \end{bmatrix} \tag{34}$$

Bindingsligningens akselerasjonsuavhengige ledd $\boldsymbol{\gamma}$ vises i matrisen nedenfor. Vær oppmerksom på at det finnes akselerasjonsuavhengige ledd i de dobbelderiverte longitudinale bindingene. Dette gir flere ledd i $\boldsymbol{\gamma}$.

$$\ddot{\boldsymbol{d}}_{\boldsymbol{1}} = \ddot{\boldsymbol{r}}_{\boldsymbol{1}} + \boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{s_1}^{D'} + \boldsymbol{A_1}\dot{\widetilde{\boldsymbol{\omega}}}_{\boldsymbol{1}}\boldsymbol{s_1}^{D'} - \ddot{\boldsymbol{r}}_{\boldsymbol{2}} - \boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{s_2}^{E'} - \boldsymbol{A_2}\dot{\widetilde{\boldsymbol{\omega}}}_{\boldsymbol{2}}\boldsymbol{s_2}^{E'} \tag{35}$$

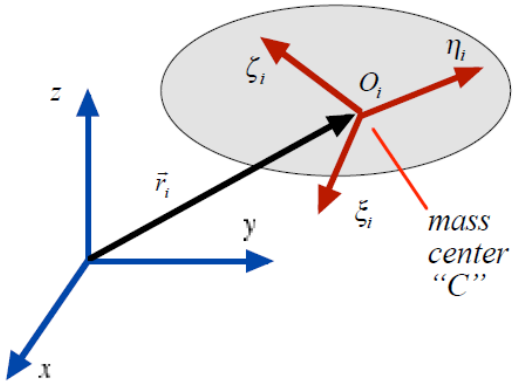
Og:

$$\ddot{\boldsymbol{d}}_{\boldsymbol{2}} = \ddot{\boldsymbol{r}}_{\boldsymbol{2}} + \boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{s_2}^{F'} + \boldsymbol{A_2}\dot{\widetilde{\boldsymbol{\omega}}}_{\boldsymbol{2}}\boldsymbol{s_2}^{F'} - \ddot{\boldsymbol{r}}_{\boldsymbol{3}} - \boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{s_3}^{G'} - \boldsymbol{A_3}\dot{\widetilde{\boldsymbol{\omega}}}_{\boldsymbol{3}}\boldsymbol{s_3}^{G'} \tag{36}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{s_1}^{A'} \\ \boldsymbol{u_x}^T(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\zeta_1}) \\ \boldsymbol{u_y}^T(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\zeta_1}) \\ \boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{s_1}^{B'} - \boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{s_2}^{B'} \\ \left(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\boldsymbol{u}_{\zeta_2}\right) + \left(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\zeta_2}\right) + \left(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\zeta_2}\right) + \left(\boldsymbol{A_1}\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\zeta_2}\right) \\ \left(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\boldsymbol{u}_{\eta_2}\right) + \left(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\eta_2}\right) + \left(\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\eta_2}\right) + \left(\boldsymbol{A_1}\boldsymbol{u}_{\xi_1}\right)^T\left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\eta_2}\right) \\ \boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{s_2}^{C'} - \boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{s_3}^{C'} \\ \left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\boldsymbol{u}_{\zeta_3}\right) + \left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{u}_{\zeta_3}\right) + \left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{u}_{\zeta_3}\right) + \left(\boldsymbol{A_2}\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{u}_{\zeta_3}\right) \\ \left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\boldsymbol{u}_{\eta_3}\right) + \left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{u}_{\eta_3}\right) + \left(\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{u}_{\eta_3}\right) + \left(\boldsymbol{A_2}\boldsymbol{u}_{\xi_2}\right)^T\left(\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{u}_{\eta_3}\right) \\ \ddot{\varphi}_{\eta_1} \\ \left(2\dot{\boldsymbol{d}}_{\boldsymbol{1}}{}^T\dot{\boldsymbol{d}}_{\boldsymbol{1}}\right) - 2\dot{l}_1(t)\dot{l}_1(t) - 2l_1(t)\ddot{l}_1(t) + 2\boldsymbol{d_1}{}^T\boldsymbol{A_1}\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{1}}}'\boldsymbol{s_1}^{D'} - 2\boldsymbol{d_1}{}^T\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{s_2}^{E'} \\ \left(2\dot{\boldsymbol{d}}_{\boldsymbol{2}}{}^T\dot{\boldsymbol{d}}_{\boldsymbol{2}}\right) - 2\dot{l}_2(t)\dot{l}_2(t) - 2l_2(t)\ddot{l}_2(t) + 2\boldsymbol{d_2}{}^T\boldsymbol{A_2}\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{2}}}'\boldsymbol{s_2}^{F'} - 2\boldsymbol{d_2}{}^T\boldsymbol{A_3}\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\widetilde{\boldsymbol{\omega}_{\boldsymbol{3}}}'\boldsymbol{s_3}^{G'} \end{bmatrix} \tag{37}$$

7 Reaksjonskrefter

Kunnskap om kranens reaksjonskrefter er nødvendig for dimensjonering av kranens konstruksjon. Reaksjonskrefter beregnes ved hjelp av Jakobimatrisen og ved bruk av Lagranges multiplikator. Metoden beskrives siden.



Figur 7.1 Legemets lokale koordinatsystem ligger i massesenteret

Bevegelsesligningene for et fritt legeme beskrives ved hjelp av Newtons 2.lov og Eulers rotasjonsligning. Bevegelsesligningen for translasjonen til et legemes massesenter beskrives som:

$$\sum \boldsymbol{f} = m\ddot{\boldsymbol{r}} \tag{38}$$

Hvor:

m = Legemets masse

$\sum \boldsymbol{f}$ = Kraftvektor som virker på legemet.

$\ddot{\boldsymbol{r}}$ = Massesenterets akselerasjon.

Ligningen kan settes opp som en matrise:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} \ddot{r}_x \\ \ddot{r}_y \\ \ddot{r}_z \end{bmatrix} \tag{39}$$

Rotasjonsligningen til legemet er:

$$\boldsymbol{n}' = \boldsymbol{J}'\dot{\boldsymbol{\omega}}' + \widetilde{\boldsymbol{\omega}'}\boldsymbol{J}'\boldsymbol{\omega}' \tag{40}$$

Hvor:

$\boldsymbol{J}'$ = Treghetsmomentet til legemet
$\boldsymbol{n}'$ = Moment

Translasjons ligningen (Newtons 2.lov) er beskrevet ut fra globalt akse system, mens rotasjons ligningen (Euler) er beskrevet i legemets lokale system. Bevegelsesligningene gir ligningsmatrisen:

$$\boldsymbol{M}_i\dot{\boldsymbol{v}}_i = \boldsymbol{g}_i \tag{41}$$

Hvor:

$$\boldsymbol{M}_i = \begin{bmatrix} m_i\boldsymbol{I} & 0 \\ 0 & \boldsymbol{J}_i' \end{bmatrix}, \qquad \dot{\boldsymbol{v}}_i = \begin{bmatrix} \ddot{\boldsymbol{r}}_i \\ \dot{\boldsymbol{\omega}}_i' \end{bmatrix}, \qquad \boldsymbol{g}_i = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{n}_i' - \widetilde{\boldsymbol{\omega}}_i'\boldsymbol{J}_i'\boldsymbol{\omega}_i' \end{bmatrix}$$

Et system av legemer vil ha en eller flere kinematiske bindinger. Disse bindingene tilfører eksterne krefter på systemet.

$$\boldsymbol{M}_i\dot{\boldsymbol{v}}_i = \boldsymbol{g}_i + \boldsymbol{g}_i^{\boldsymbol{C}} \tag{42}$$

Hvor:

$\boldsymbol{g}_i^{\boldsymbol{C}}$ = reaksjonskrefter i bindingene

Reaksjonskreftene kan bestemmes ved hjelp av Lagrange multiplikator.

$$\boldsymbol{g}_i^{\boldsymbol{C}} = \boldsymbol{\Phi}_{\boldsymbol{q}}^{\boldsymbol{T}}\boldsymbol{\lambda} \tag{43}$$

Hvor:

$\boldsymbol{\Phi_q}^{\boldsymbol{T}}$ = Jacobian transponert

$\boldsymbol{\lambda}$ = Lagrange multiplikator

Lagranges multiplikator er en vektor som inneholder like mange elementer som antall kinematiske bindinger i systemet.

$$\boldsymbol{\lambda} = \left(\boldsymbol{\Phi_q}^{\boldsymbol{T}}\right)^{-\mathbf{1}}(\boldsymbol{M}_i\ddot{\boldsymbol{q}} - \boldsymbol{g}_i) \tag{44}$$

Ved hjelp av Lagrange multiplikator og Jacobian invers beregnes krefter i bindinger og drivere (Nikravesh, 2013).

8 Statisk analyse

For å kontrollere reaksjonskrefter beregnet gjennom bruk av Lagrange multiplikator utføres en statisk analyse av systemet. Det betyr at man studerer systemet i en tilstand som ikke endrer seg over tid. Et stasjonært system må oppfylle likevekts betingelsene.

1. Vektorsummen av de ytre kreftene som virker på legemet må være lik null.

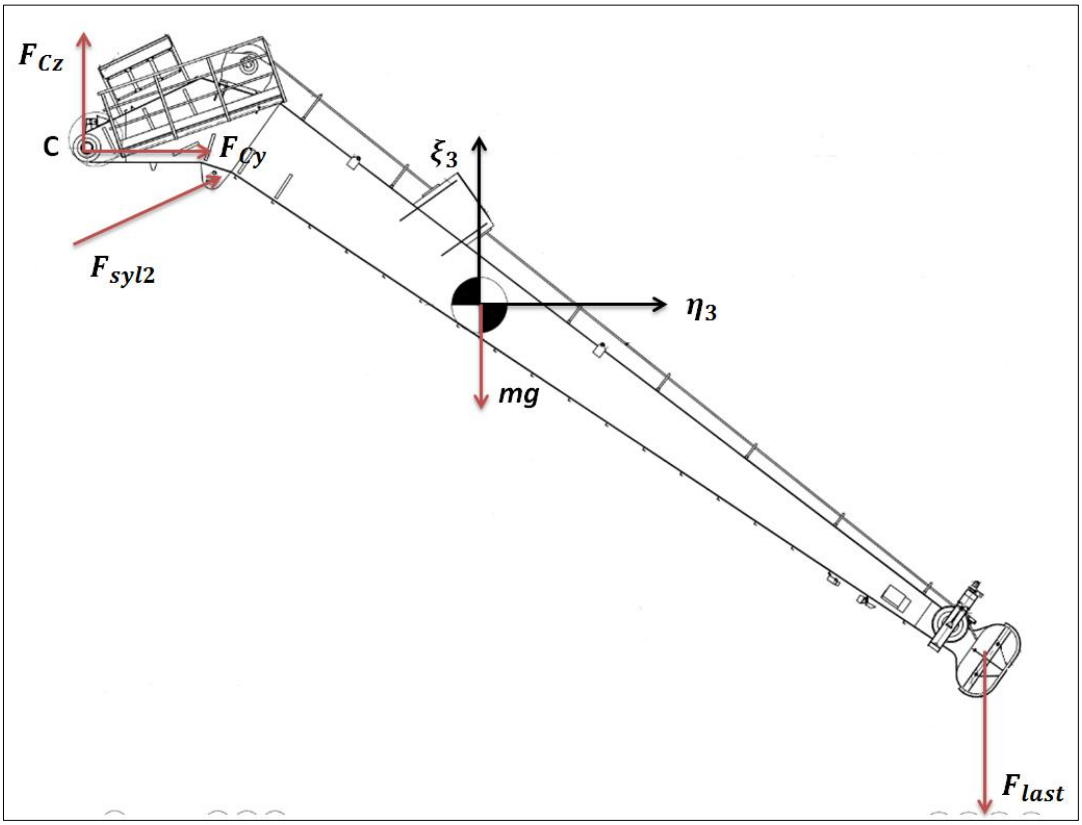
   $$\sum F = 0$$

2. Summen av kraftmoment om et vilkårlig valgt momentpunkt må være lik null.

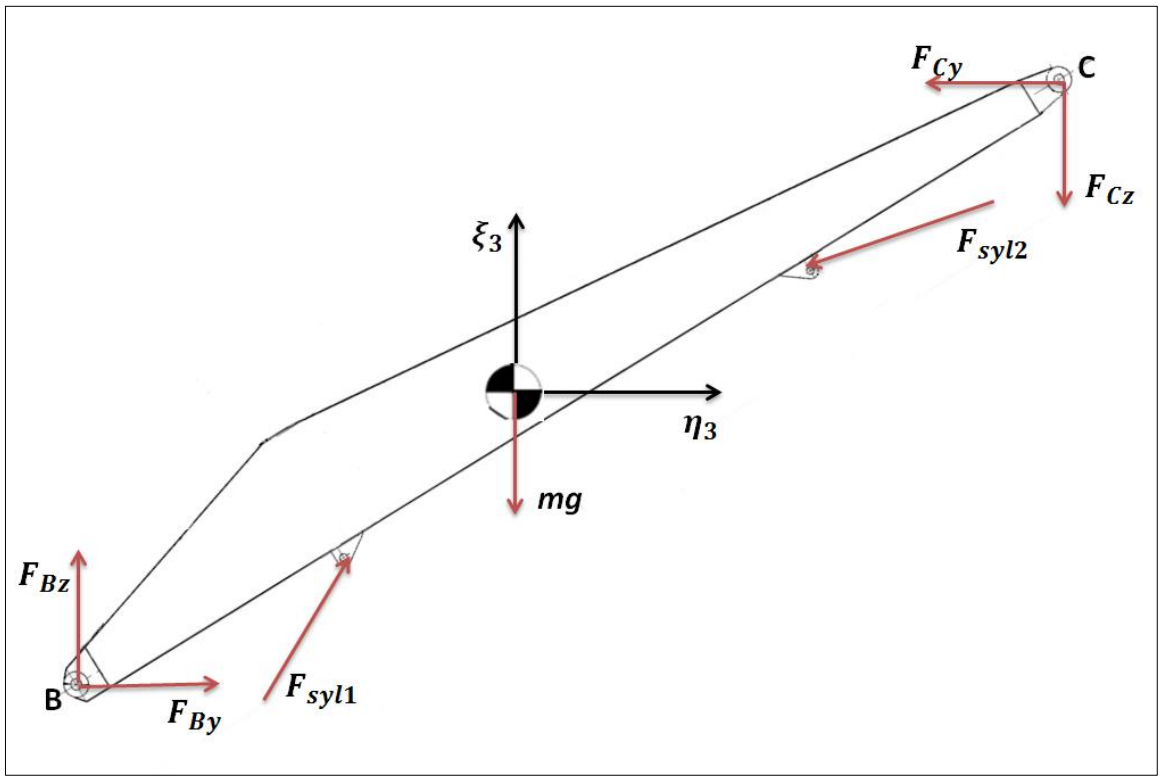
   $$\sum M = 0$$

(Davidsen, 2010)

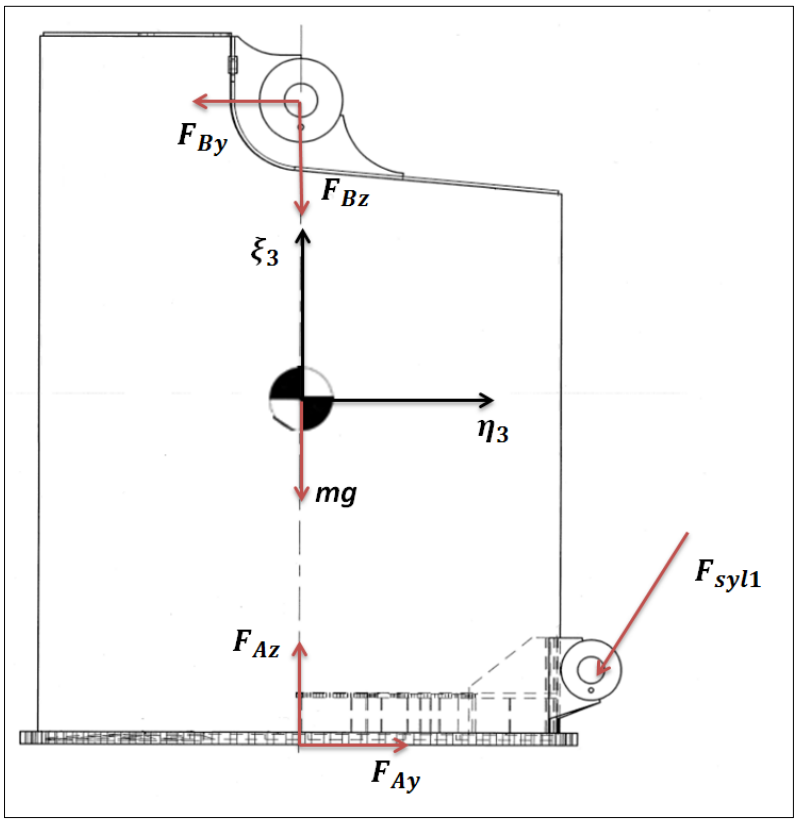
Med utgangspunkt i tekniske tegninger for kranen lages en oversikt over hvilke reaksjonskrefter som virker på kranen. Legemets massesenter og reaksjonskrefter markeres.



Figur 8.1 Reaksjonskrefter kranarm 2



Figur 8.2 Reaksjonskrefter kranarm 1



Figur 8.3 Reaksjonskrefter krantårn

I den statiske analysen ønsker man å studere kreftene som oppstår i ledd og drivere. Kreftene er like store, men motsatt rettet i koblinger mellom legemer. Figurene viser angrepspunkt for kreftene (røde piler). Kreftene som oppstår vil være avhengig av kranens posisjon og den påførte lasten, F_{last}.

Definerte vektorlengder:

$\boldsymbol{r}_{A-B} = -A_1 s'_{1A} + A_1 s'_{1B}$ Vektor fra punkt A til B

$\boldsymbol{r}_{A-D} = -A_1 s'_{1A} + A_1 s'_{1D}$ Vektor fra punkt A til D

$\boldsymbol{r}_{A-I} = -A_1 s'_{1A}$ Vektor fra punkt A til krantårnets massesenter

$\boldsymbol{r}_{B-C} = -A_2 s'_{2B} + A_2 s'_{2C}$ Vektor fra punkt B til C

$\boldsymbol{r}_{B-F} = -A_2 s'_{2B} + A_2 s'_{2F}$ Vektor fra punkt B til F

$\boldsymbol{r}_{B-J} = -A_2 s'_{2B}$ Vektor fra punkt B til kranarm 1 sitt massesenter

$\boldsymbol{r}_{B-E} = -A_2 s'_{2B} + A_2 s'_{2E}$ Vektor fra punkt B til E

$\boldsymbol{r}_{C-H} = -A_3 s'_{3C} + A_3 s'_{3H}$ Vektor fra punkt C til H

$\boldsymbol{r}_{C-K} = -A_3 s'_{3C}$ Vektor fra punkt C til kranarm 2 sitt massesenter

$\boldsymbol{r}_{C-G} = -A_3 s'_{3C} + A_3 s'_{3G}$ Vektor fra punkt C til G

Definerte krefter:

$\boldsymbol{F}_{syl1}$ Kraftvektor sylinder 1

$\boldsymbol{F}_{syl2}$ Kraftvektor sylinder 2

$\boldsymbol{F}_{Ay}$ Kraftvektor i binding A, y-retning

$\boldsymbol{F}_{Az}$ Kraftvektor i binding A, z-retning

$\boldsymbol{F}_{By}$ Kraftvektor i binding B, y-retning

$\boldsymbol{F}_{Bz}$ Kraftvektor i binding B, z-retning

$\boldsymbol{F}_{Cy}$ Kraftvektor i binding C, y-retning

$\boldsymbol{F}_{Cz}$ Kraftvektor i binding C, z-retning

Kreftene er motsatt rettet på legemene.

Dekomponert kraftvektor for $\boldsymbol{F}_{syl1}$ og $\boldsymbol{F}_{syl2}$ finner man ved hjelp av vektorenes tilhørende enhetsvektorer.

$\boldsymbol{u}_1 = \frac{1}{|\boldsymbol{d_1}|}\boldsymbol{d_1}$ Enhetsvektor sylinder 1

$\boldsymbol{u}_2 = \frac{1}{|\boldsymbol{d_2}|}\boldsymbol{d_2}$ Enhetsvektor sylinder 2

Ut fra skisser, og ved hjelp av likevekts ligningene finner man kreftene som virker på systemet.
Kraftmoment om ledd C for kranarm 2:

$$\sum M_C = \boldsymbol{r}_{C-H} \times \begin{bmatrix} 0 \\ 0 \\ -F_{last} \end{bmatrix} + \boldsymbol{r}_{C-K} \times \begin{bmatrix} 0 \\ 0 \\ -m_3 g \end{bmatrix} + F_{syl,2}(\boldsymbol{r}_{C-G} \times \boldsymbol{u}_2) = 0 \quad (45)$$

Summen av krefter kranarm 2:

$$\sum F = \begin{bmatrix} F_{Cx} \\ F_{Cy} \\ F_{Cz} \end{bmatrix} + F_{cyl2}\boldsymbol{u}_2 + \begin{bmatrix} 0 \\ 0 \\ -m_3 g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -F_{last} \end{bmatrix} = 0 \tag{46}$$

Kraftmoment om ledd B for kranarm1:

$$\sum M_B = \boldsymbol{r}_{B-C} \times \begin{bmatrix} -F_{Cx} \\ -F_{Cy} \\ -F_{Cz} \end{bmatrix} - F_{syl,2}(\boldsymbol{r}_{B-F} \times \boldsymbol{u}_2) + \boldsymbol{r}_{B-J} \times \begin{bmatrix} 0 \\ 0 \\ -m_2 g \end{bmatrix} + F_{syl,1}(\boldsymbol{r}_{B-E} \times \boldsymbol{u}_1) = 0 \tag{47}$$

Summen av krefter kranarm 1:

$$\sum F = \begin{bmatrix} F_{Bx} \\ F_{By} \\ F_{Bz} \end{bmatrix} + F_{cyl1}\boldsymbol{u}_1 + \begin{bmatrix} 0 \\ 0 \\ -m_2 g \end{bmatrix} - F_{syl,2}\boldsymbol{u}_2 + \begin{bmatrix} -F_{Cx} \\ -F_{Cy} \\ -F_{Cz} \end{bmatrix} = 0 \tag{48}$$

Kraftmoment om ledd A for krantårn:

$$\sum M_A = \boldsymbol{r}_{A-B} \times \begin{bmatrix} -F_{Bx} \\ -F_{By} \\ -F_{Bz} \end{bmatrix} - F_{syl,1}(\boldsymbol{r}_{A-D} \times \boldsymbol{u}_1) + \boldsymbol{r}_{A-I} \times \begin{bmatrix} 0 \\ 0 \\ -m_1 g \end{bmatrix} = 0 \tag{49}$$

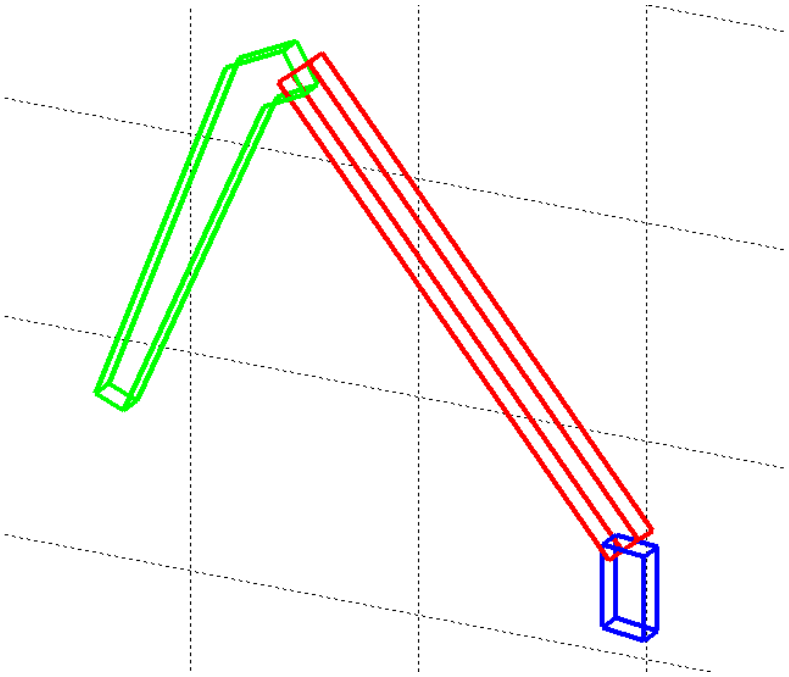
Summen av krefter krantårn:

$$\sum F = \begin{bmatrix} F_{Ax} \\ F_{Ay} \\ F_{Az} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_1 g \end{bmatrix} - F_{cyl1}\boldsymbol{u}_1 + \begin{bmatrix} -F_{Bx} \\ -F_{By} \\ -F_{Bz} \end{bmatrix} = 0 \tag{50}$$

Ved å utføre en statisk analyse av systemet kan man validere reaksjonskreftene man får gjennom bruk av Lagranges multiplikator. Dette gjøres ved å låse driverne slik at den dynamiske analysen i Matlab blir stasjonær. Resultater av valideringen kommer i kapittel 9.4.

9 Validering av kinematisk modell i Matlab

Det er laget en enkel visualisering av kranen for enklere å kontrollere og forstå kranens oppbygning og kinetikk. Denne visualiseringen hjelper med å kontrollere endringer som gjøres. Alle endringer i krandesign kan gjøres i Matlab funksjonens konstanter. Her har man mulighet til å endre kranens dimensjoner, treghetsmoment, driverdynamikk og simuleringstid. Alle endringer i dimensjoner genereres automatisk til visualiseringen. Figur 9.1 viser denne kranvisualiseringen.

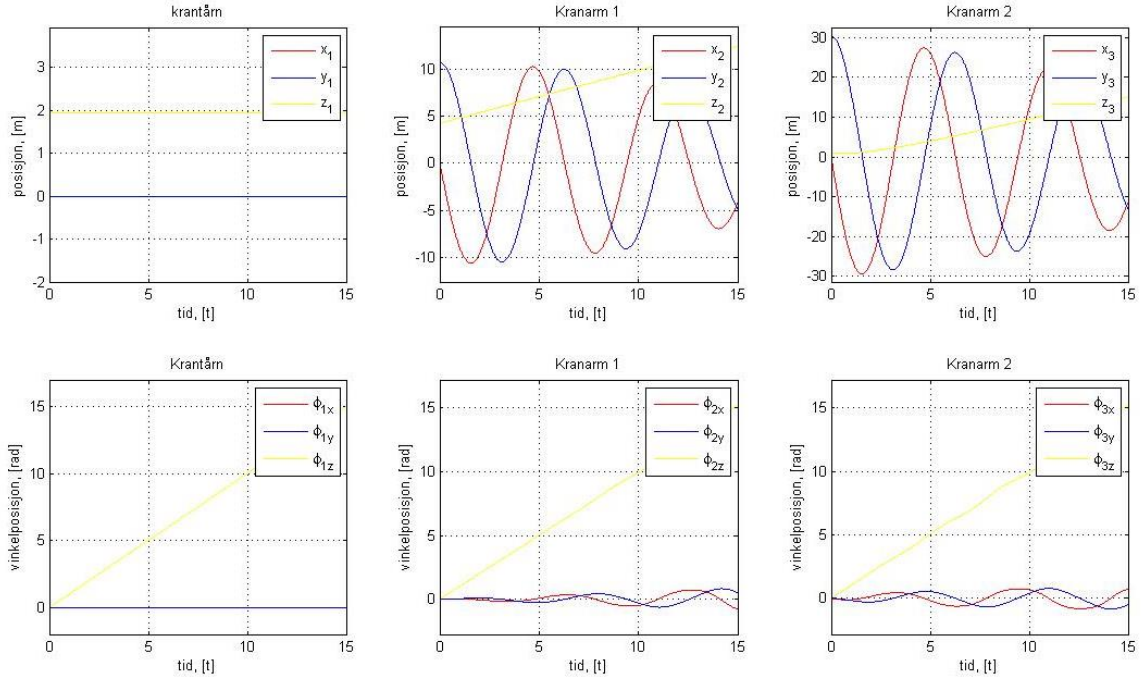


Figur 9.1 Visualisering av kran (matlab)

Visualiseringen gir en god indikasjon på om dimensjoner og bevegelsesmønster for legemene er korrekte, men gir ingen informasjon om hvilke krefter som virker. Validering av krefter er gjort i kapittel 9.4. Alle analysene av krankinematikken er utført ved å la rotasjonsdriver rotere med $d(t) = 1\ rad/s$. Hydrauliske sylindre er bestemt til å kjøre med, $l_{syl1}(t) = 0{,}2m/s$ og $l_{syl2}(t) = -0{,}2m/s$. Analysene er utført med en simuleringstid på 15 sekunder. Kranen starter med vinkel orientering lik null for kranlegemer.

9.1 Posisjon

Figurene under viser posisjonering og orientering av lokale koordinatsystem med hensyn til globalt koordinatsystem (x,y,z). Posisjon har benevning meter, og vinkelposisjon har benevning radianer.

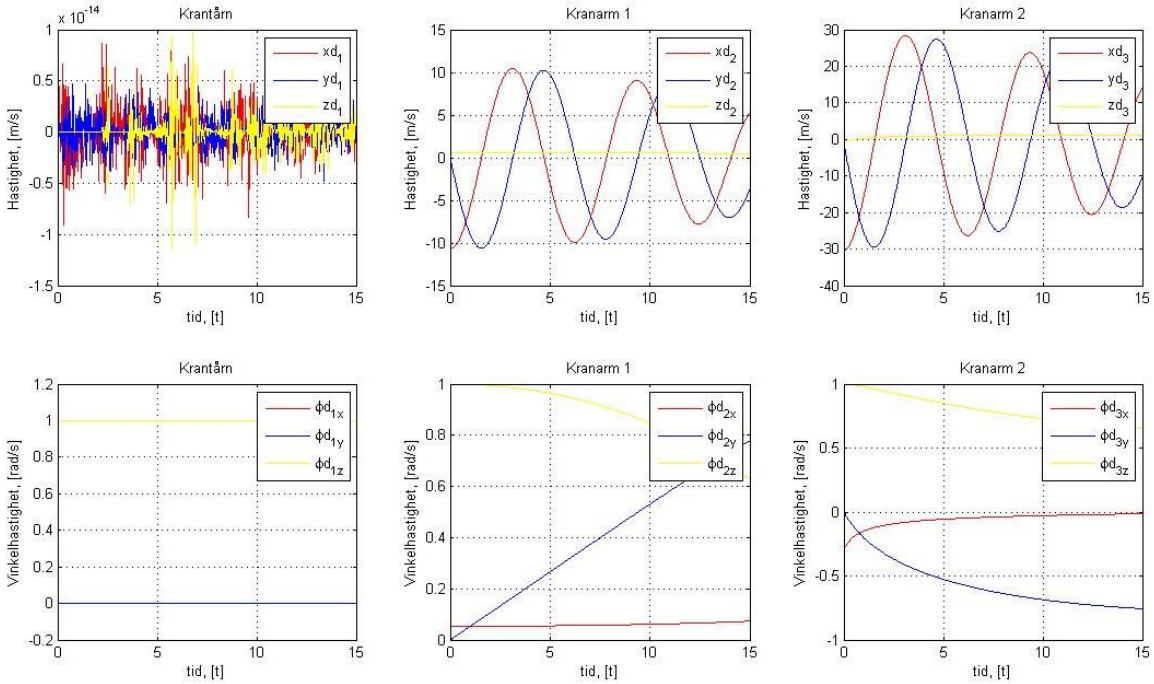


Figur 9.2 Posisjonsanalyse

Plotet øverst til venstre viser at posisjon x_1, y_1 og z_1 ikke endrer posisjon. Høyden z_1 er definert som halve høyden av krantårnet. Figuren nederst til venstre indikerer at krantårnet (King) endrer sin vinkelposisjon om z-aksen med tiden, t (gul graf). Orientering om global x og y-akse er stasjonær lik null. Plot 2 og 3 øverst fra venstre viser at kranarm 1 og 2 endrer utstrekning om global z-akse. Dette kommer frem av de sinuslignende kurvene for posisjon x_2, y_2 og x_3, y_3. Studerer man plotene kan en se at amplitudene endrer seg over tid. Dette har sammenheng i at avstanden fra global z-akse minker på grunn av at kranarmene trekker seg sammen. Plot 5 og 6 nederst viser at kranarm 1 og 2 roterer med samme hastighet om z-aksen. Det registreres også endring i vinkelposisjon om x og y-aksen på de økende sinuskurvene. Dette er forårsaket av driverne for hydrauliske sylindre. Kranarm 1 og 2 er satt til å starte ved null rotasjon.

9.2 Hastighet

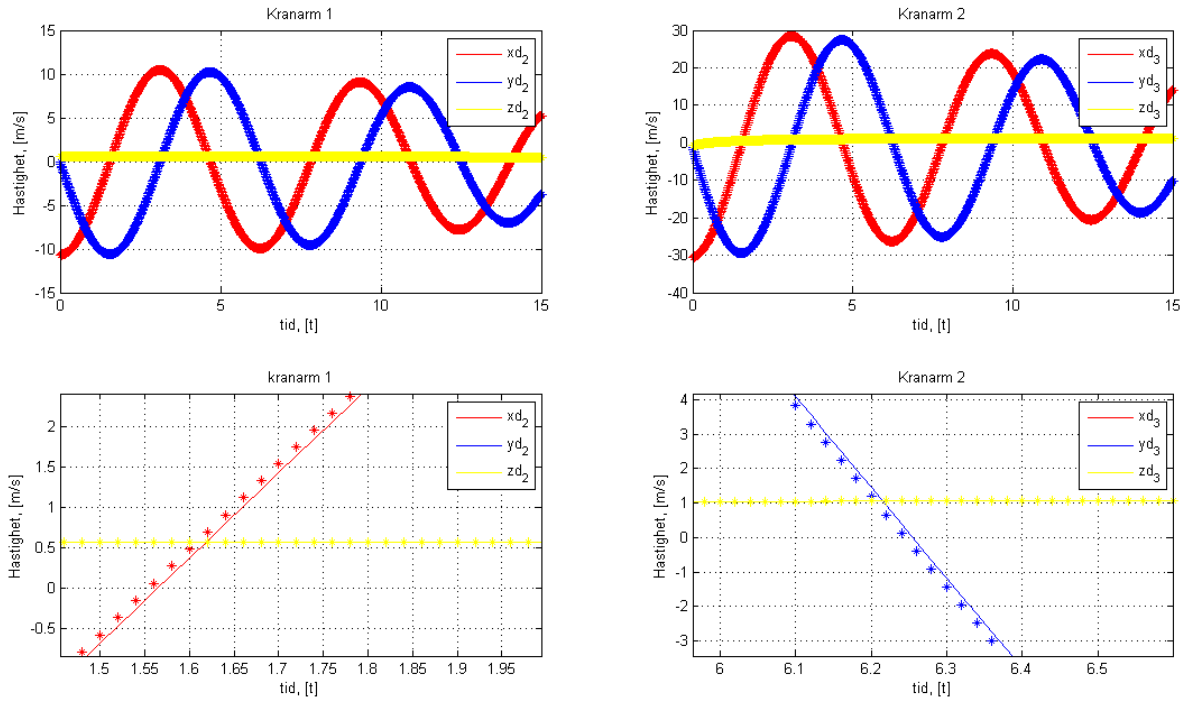
Figurene under viser vinkelhastighet og hastighet for lokale koordinatsystem med hensyn til globalt koordinatsystem (x,y,z). Vinkelhastighet har benevning rad/s, og hastighet har benevning m/s.



Figur 9.3 Hastighetsanalyse

Plot 1 øverst til venstre viser at hastighet xd_1, yd_1 og zd_1 er lik null. Utslagene i grafen er i størrelsesorden 10^{-14} som i praksis har liten betydning. Subplot 4 viser at krantårnet roterer med 1 rad/s om global z-akse. Det er ingen rotasjon om global x og y-akse. Plot 2 og 3 øverst fra venstre viser at rotasjonshastigheten om global z-akse for kranarm 1 og 2 minker i amplitude. Dette er igjen fordi kranarmene trekker seg sammen.

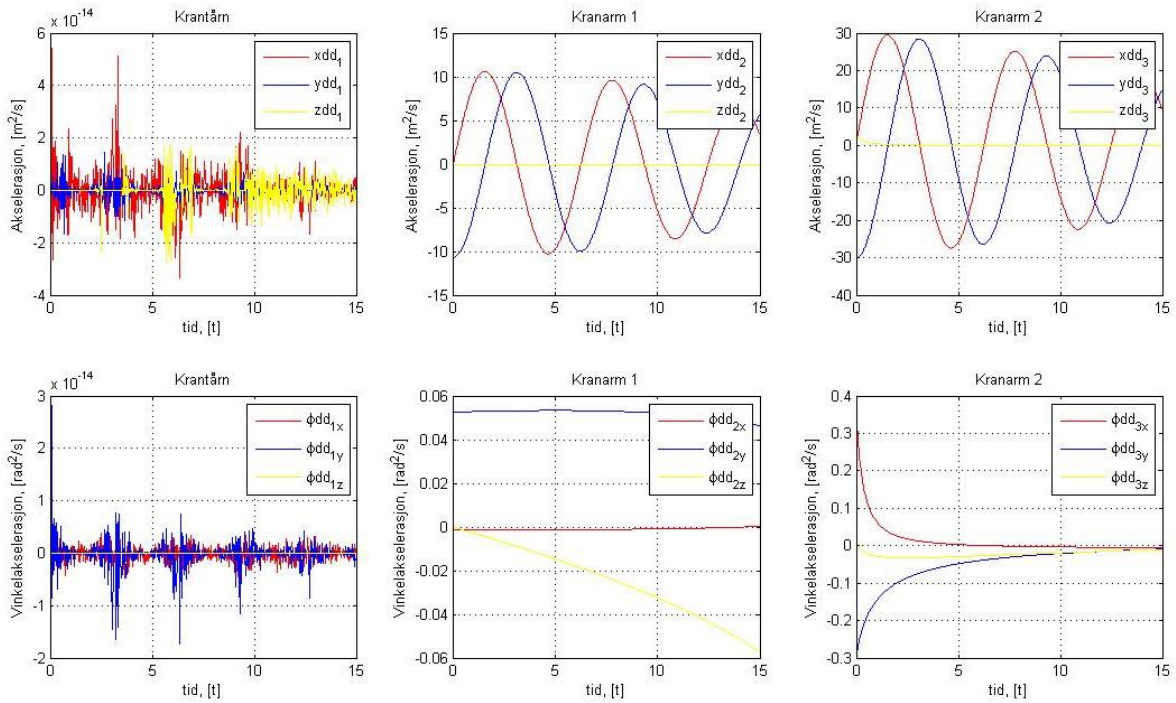
Det er foretatt en numerisk differensiering i Matlab for å kontrollere eventuelle feil ved hastighet og akselerasjons analyse. I Figur 9.4 er resultatet av den numeriske differensieringen plottet sammen med hastighetsanalysen basert på Jacobian. Plott 1 og 2 øverst viser hele hendelsesforløpet, mens plot 3 og 4 nederst er forstørret opp for mer detaljert visning. Resultatene viser at kurvene følger hverandre tett. Ut fra disse resultatene er det sannsynlig at hastighetsanalysen er korrekt.



Figur 9.4 Sammenligning mellom numerisk og Jacobi basert hastighetsanalyse.

9.3 Akselerasjon

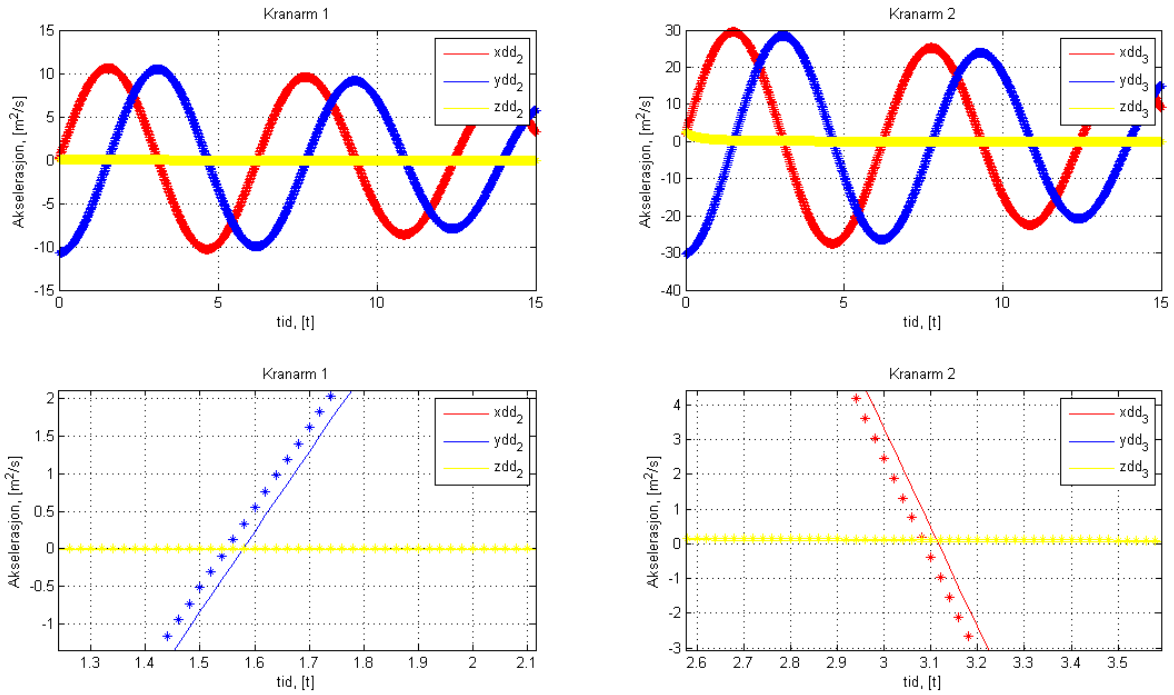
Figurene under viser vinkelakselerasjon og akselerasjon for lokale koordinatsystem med hensyn til globalt koordinatsystem (x,y,z). Vinkelakselerasjon har benevning rad/s^2 og akselerasjon har benevning m/s^2.



Figur 9.5 Akselerasjonsanalyse

Plot 1 og 4 øverst og nederst til venstre viser at akselerasjon for krantårn er i størrelses orden 10^{-14}. Dette er korrekt da krantårnet er stasjonert med konstant hastighet for rotasjon om z-aksen.

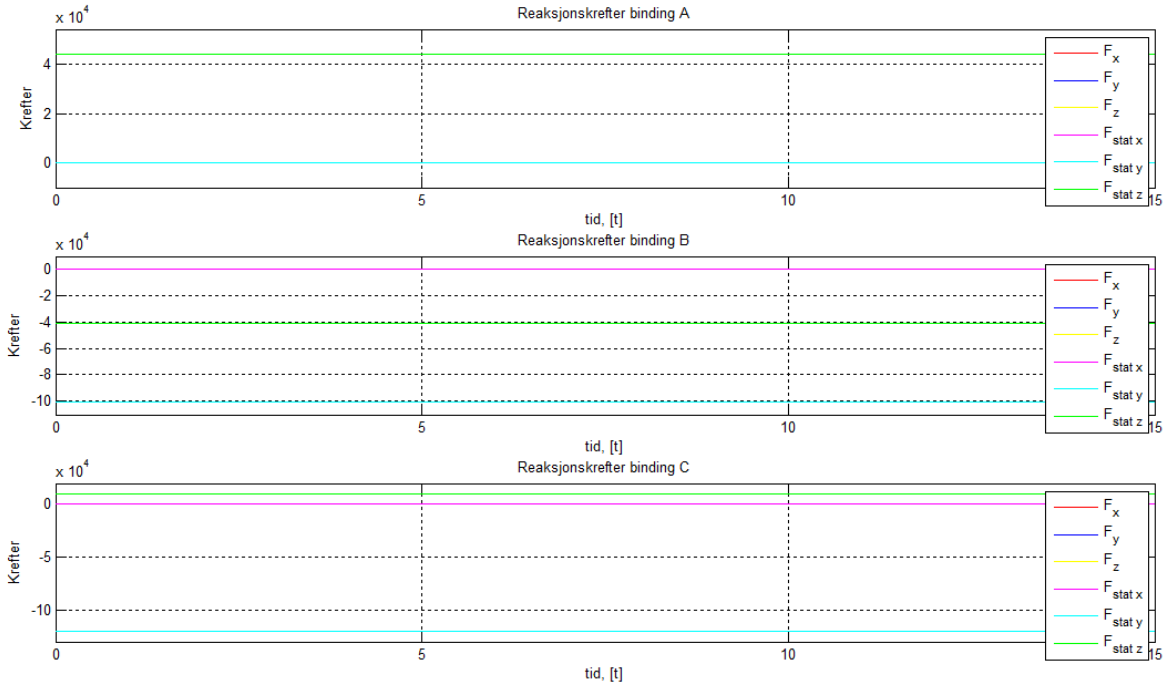
På samme måte som ved hastighetsanalysen er det gjort en numerisk kontroll av akselerasjon for kranarmer. Plotene under viser den tidsderiverte av hastighet, sammen med akselerasjon basert på Jacobian invers. Kurvene følger hverandre tett, og man kan anta at analysen er korrekt.



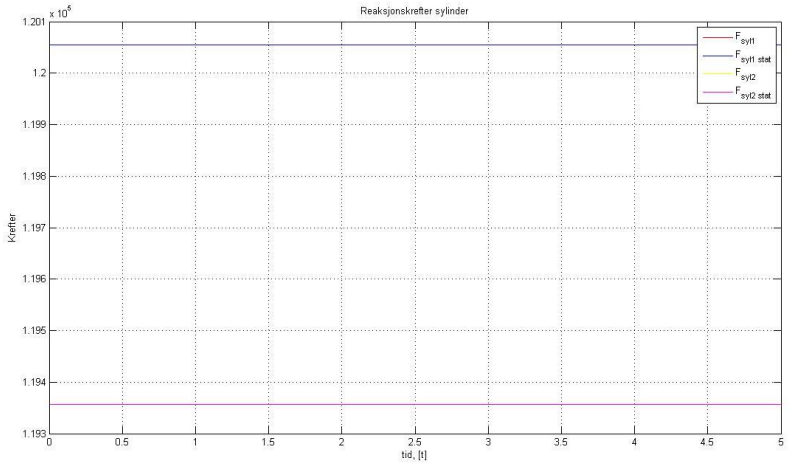
Figur 9.6 Sammenligning mellom numerisk og jacobian basert akselerasjonsanalyse.

9.4 Reaksjonskrefter

For å kontrollere at reaksjonskrefter beregnet gjennom Lagranges multiplikator er korrekt sammenligner man resultatene med statiske beregninger. Man endrer driverne fra dynamisk til statisk tilstand. $l_{syl1}(t) = 0\ m/s,\ l_{syl2}(t) = 0\ m/s$ og $d(t) = 0\ rad/s$. Resultatet av dette viser at statiske beregninger er eksakt lik reaksjonskrefter basert på Lagrains multiplikator og Jacobian invers.

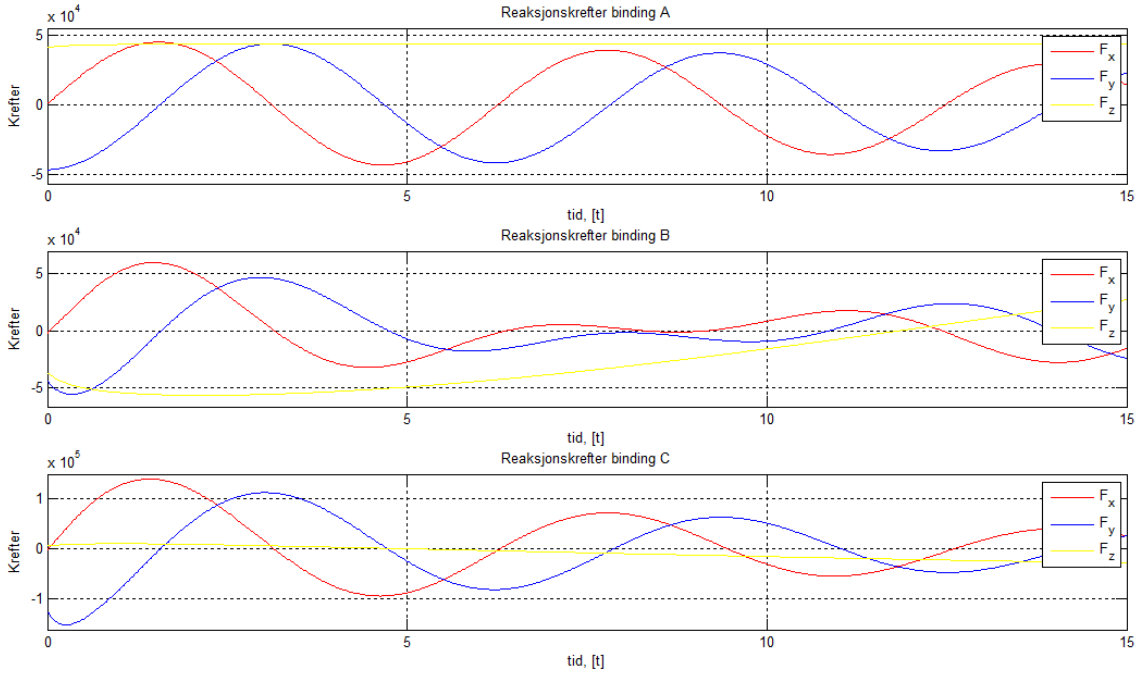


Figur 9.7 Sammenlignig av reaksjonskrefter



Figur 9.8 Sammenligning reaksjonskrefter for sylindre

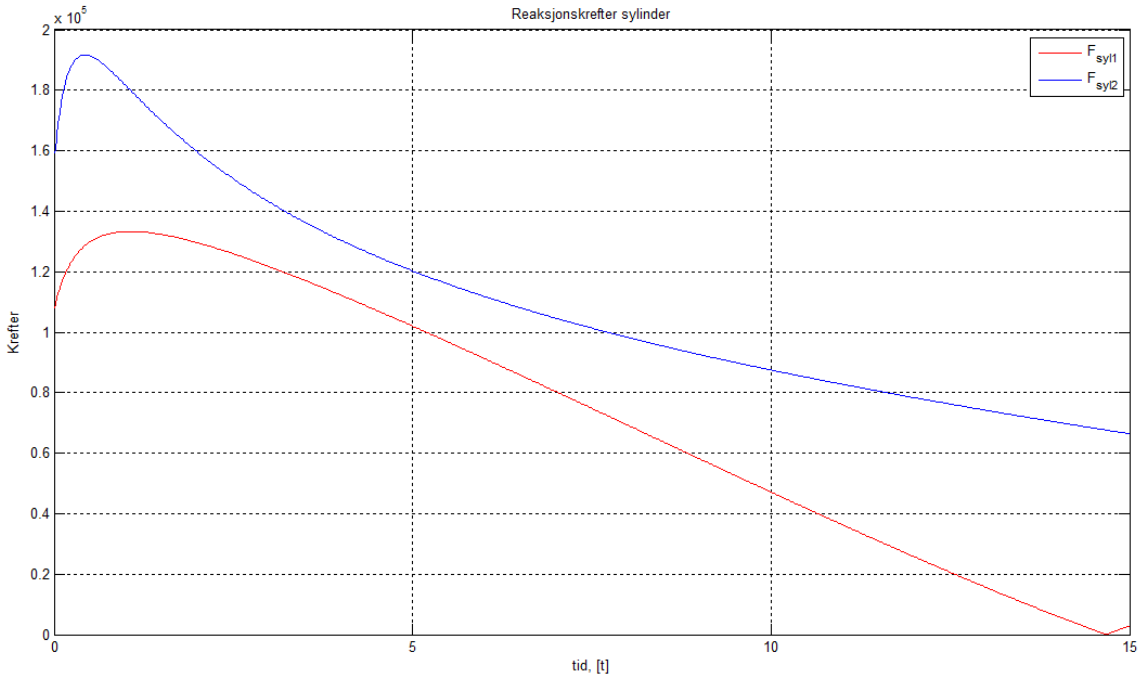
Resultatene viser at statiske beregninger av reaksjonskrefter er like for de to teknikkene. Det kan likevel være av interesse å studere de dynamiske reaksjonskreftene. Nedenfor vises resultater fra en simulering med samme pådrag for sylindre og rotasjonsdriver som for posisjonsanalysen. $d(t) = 1\,rad/s,\ l_{syl1}(t) = 0{,}2m/s$ og $l_{syl2}(t) = -0{,}2m/s$. Benevning for kreftene er Newton.



Figur 9.9 Reaksjonskrefter bindinger

Plot 1 øverst beskriver reaksjonskrefter for binding mellom krantårn og fundament. Ut fra plotet kan man se at kraft fra underlaget i z-retning er konstant og lik kraften fra statisk beregning. Dette er kraften som holder kranen stasjonært til bakkenivå, og kan derfor antas å være korrekt. Kreftene i x og y-retning har form som en sinus på grunn av kranens rotasjon om z-aksen. Amplituden minker som funksjon av tid på grunn av at kranarmene trekker seg sammen og derfor gir mindre moment om z-aksen.

Figur 9.10 nedenfor viser krefter påført sylindre.



Figur 9.10 Krefter hydrauliske sylindre

Kreftene som er påført de hydrauliske sylindrene senkes etter som at kranarmer endrer posisjon. På bakgrunn av dette kan en anta at reaksjonskrefter og sylinderkrefter også er korrekte i dynamisk tilstand.

10 Optimering av design

Ut fra den kinematiske modellen av kranen kan man studere kranens optimale design. Målet er å lage en rutine som automatisk finner optimal design av kranen. Hva en optimal design er, er avhengig av hva en ønsker at kranen skal gjøre. En optimal design er derfor som regel en vekting mellom flere designønsker. Det må skilles mellom hva som er absolutte designspesifikasjoner og hva som er ønskelig å optimere. En optimering må overholde designspesifikasjonene. I denne rapporten optimeres kranen stegvis. En automatisk modell for å bestemme og optimere design vil ha stor betydning for hvor godt designet blir, samt spare mye utviklingsarbeid.

10.1 Fmincon

Til optimeringen brukes optimeringsrutinen, fmincon, levert av Matlab. Denne rutinen er designet for å håndtere grensebetingede ulineære optimeringsproblem. Fmincon forsøker å finne et grensebetinget minimum av en skalar funksjon med flere variabler. Funksjonen bestemmes av:

$$\min_{x} f(x) \quad \begin{cases} \boldsymbol{c}(\boldsymbol{x}) \leq 0 \\ \boldsymbol{ceq}(\boldsymbol{x}) = 0 \\ \boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{b} \\ \boldsymbol{Aeq} \cdot \boldsymbol{x} = \boldsymbol{beq} \\ \boldsymbol{lb} \leq \boldsymbol{x} \leq \boldsymbol{ub} \end{cases} \tag{51}$$

Hvor:

$\boldsymbol{c}(\boldsymbol{x})$ og $\boldsymbol{ceq}(\boldsymbol{x})$ er en funksjon som returnerer en vektor med bestemte grenseverdier. Funksjonen kan være ulineær.

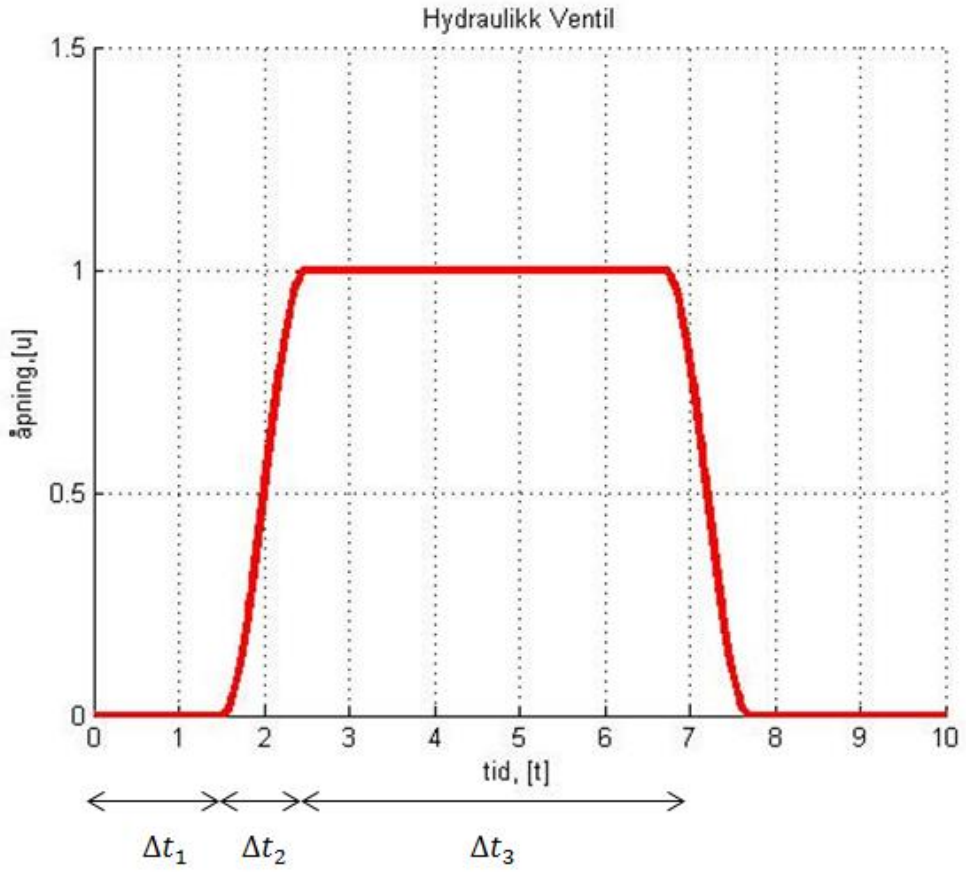
$\boldsymbol{A}$ og $\boldsymbol{Aeq}$ er matriser som definerer grenser for lineært avhengige og uavhengige designvariabler bestemt av vektor $\boldsymbol{b}$ og $\boldsymbol{beq}$.

$\boldsymbol{lb}$ og $\boldsymbol{ub}$ er vektorer som bestemmer øvre og nedre grenseverdi for designvariabler, $\boldsymbol{x}$.

Objektsfunksjonen, $f(x)$, returnerer en minimert funksjonsverdi basert på designvariabler, $\boldsymbol{x}$, og grenseverdier. Designvariablene for rutinen starter ut fra estimerte startverdier. Startverdier oppgis i vektor, $\boldsymbol{x_0}$.

10.2 Driverfunksjon

I det første optimeringsforsøket, kapittel 10.3, optimeres kranens driverfunksjoner, $l_{syl1}(t)$, $l_{syl2}(t)$ og $d(t)$. Kranens drivere styres ved hjelp av hydrauliske retningsventiler. Ventilene kan åpnes og lukkes som en funksjon av tid (t). Funksjonen går fra null, som er stengt ventil, til en, som er maksimalt åpen ventil. Figur 10.1 nedenfor viser funksjonen for åpning og lukking av ventil.



Figur 10.1 Ventil funksjon (variablene $\Delta t_1 = 1,5$, $\Delta t_2 = 1$ og $\Delta t_3 = 4$ er bestemt for å gi en god illustrasjon)

Å ha mulighet til å styre driverfunksjonene til kranen gjør at det er mulig å optimere kjøremønsteret til kranen med hensyn til ønsket design. Variablene Δt_1, Δt_2 og Δt_3 innføres. Tidsintervallene er markert i Figur 10.1 Ventil funksjon (variablene $\Delta t1$=$1,5$, $\Delta t_2 = 1$ og $\Delta t_3 = 4$ er bestemt for å gi en god illustrasjon). Her er Δt_1 innslagstid for åpning av ventil, Δt_2 åpne og lukketid for ventilen og Δt_3 intervallet for åpen ventil. Det betyr at optimeringen kan justere innslagstid, åpne og lukketid, og intervall for åpen ventil. Ved å endre åpne og lukkekarakteristikken for ventiler har man mulighet for å oppnå ønsket design med hensyn på kjøremønster. Ventilfunksjonen er definert som vist under:

Åpning av ventil:

$$u(t) = 0{,}5 \cdot \sin\left(\frac{\pi}{\Delta t_2} \cdot (t - \Delta t_1) - \frac{\pi}{2}\right) + 0{,}5 \tag{52}$$

Åpen ventil:

$$u(t) = 1 \tag{53}$$

Lukking av ventil:

$$u(t) = 0{,}5 \cdot \sin\left(\frac{\pi}{\Delta t_2} \cdot (t - \Delta t_3) + \frac{\pi}{2}\right) + 0{,}5 \tag{54}$$

Oljestrømmen gjennom ventil er proporsjonal med ventilåpningen. Dette fører til at oljestrøm som funksjon av tid er:

Åpning av ventil:

$$Q(t) = 0{,}5 \cdot u \cdot Q_{max} \cdot (\sin\left(\frac{\pi}{\Delta t_2} \cdot (t - \Delta t_1) - \frac{\pi}{2}\right) + 0{,}5) \tag{55}$$

Intervall:

$$Q(t) = u \cdot Q_{max} \tag{56}$$

Lukking av ventil:

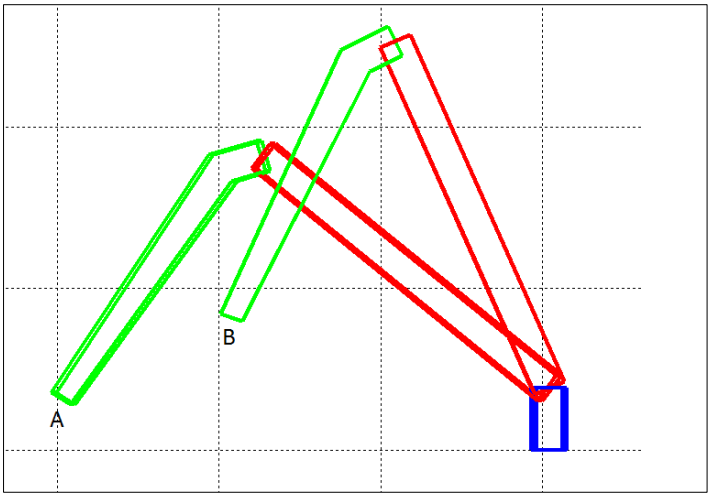
$$Q(t) = 0{,}5 \cdot u \cdot Q_{max} \cdot (\sin\left(\frac{\pi}{\Delta t_2} \cdot (t - \Delta t_1) - \frac{\pi}{2}\right) + 0{,}5) \tag{57}$$

Hvor:

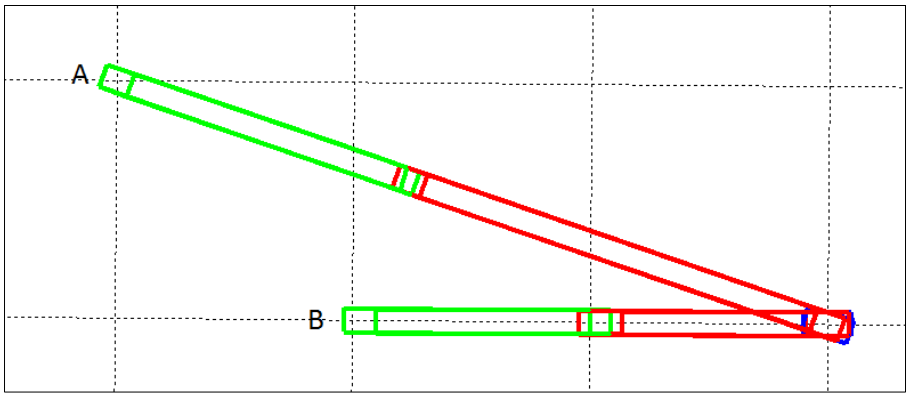
u = Pådrag for ventil (0 <= u <= 1).

Q_{max} = Maksimal oljestrøm gjennom ventil.

Ved å oppgi ønsket startpunkt, A, og sluttpunkt, B, for krantupp kan man hente ut opplysninger om nødvendig kjørelengde for driverne. Figur 10.2 og Figur 10.3 viser hva som menes med start og sluttposisjon for krantupp.



Figur 10.2 Startposisjon, A, og sluttposisjon, B, sett fra siden



Figur 10.3 Startposisjon, A og sluttposisjon, B, sett ovenfra

De ønskede vektorkoordinater for start og sluttposisjon oppgis i Matlab. For å kunne bestemme krantuppens ønskede plassering endres bindingsligningene (Φ) fra å ha driverfunksjon for posisjon, til å ha fast plassering. Dette betyr at bindingsligningen for å bestemme krantuppens posisjon er:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{r_1 + A_1 s_1}^{\boldsymbol{A\prime}} \\ \boldsymbol{u_x}^{\boldsymbol{T}}(\boldsymbol{A_1 u_{\zeta_1}}) \\ \boldsymbol{u_y}^{\boldsymbol{T}}(\boldsymbol{A_1 u_{\zeta_1}}) \\ \boldsymbol{r_1 + A_1 s_1}^{\boldsymbol{B\prime}} \boldsymbol{- r_2 - A_2 s_2}^{\boldsymbol{B\prime}} \\ (\boldsymbol{A_1 u_{\xi_1}})^{\boldsymbol{T}}(\boldsymbol{A_2 u_{\zeta_2}}) \\ (\boldsymbol{A_1 u_{\xi_1}})^{\boldsymbol{T}}(\boldsymbol{A_2 u_{\eta_2}}) \\ \boldsymbol{r_2 + A_2 s_2}^{\boldsymbol{C\prime}} \boldsymbol{- r_3 - A_3 s_3}^{\boldsymbol{C\prime}} \\ (\boldsymbol{A_2 u_{\xi_2}})^{\boldsymbol{T}}(\boldsymbol{A_3 u_{\zeta_3}}) \\ (\boldsymbol{A_2 u_{\xi_2}})^{\boldsymbol{T}}(\boldsymbol{A_3 u_{\eta_3}}) \\ \boldsymbol{r_3 + A_3 s_3}^{\boldsymbol{H\prime}} \boldsymbol{- r_H} \end{bmatrix} = 0 \qquad (58)$$

Nå bestemmes krantuppens vektorposisjon ved å bestemme vektor $\boldsymbol{r_H}$. Endring i bindingsmatrise gjør også at Jacobian matrisen endres. Kranposisjonering, $\boldsymbol{q}$, bestemmes ved hjelp av Newton Raphson. Ut fra resultatet hentes opplysninger om kjørelengde og rotasjon for kranen. Det er nå mulig å beregne nødvendig oljevolum for bevegelsen. Nødvendig oljevolum for sylindre beregnes ut fra kjørelengde og stempelareal.

Oljevolum sylinder:

$$V = \Delta l \cdot A_1 \tag{59}$$

og:

$$V = \Delta l \cdot A_2 \tag{60}$$

Hvor:

A_1 = Areal stempelside$[m^2]$.

A_2 = Areal sylinderside $[m^2]$.

Δl = posisjons forandring $[m]$.

Til å rotere kranen er det montert fire hydrauliske motorer som gjennom planetgir og tannutveksling dreier krantårnet om fundamentet. Nødvendig oljevolum for rotasjonen beregnes ut fra omdreining av krantårn multiplisert med girutvekslingen og fortrengningsvolum for de fire hydraulikkmotorene.

$$V = 4 \cdot O \cdot D \cdot i \tag{61}$$

Hvor:

O = omdreininger $[rot]$.

D = fortrengningsvolum, 90 $[cm^3/rot]$.

i = girutveksling,

Girutvekslingen for planetgirene er oppgitt til, $i_{planet} = 202$. Utgangsakslingen fra planetgiret er oppgitt å ha $n = 13$ tenner, og tannkransen til krantårnet er oppgitt å ha, $n = 126$. Dette gir en total girutveksling fra krantårn til hydraulikk motor lik:

$$i_{tot} = 202 \cdot \frac{126}{13} = 1957{,}85 \tag{62}$$

Når nødvendig oljevolum for bevegelsen er kalkulert kan man beregne ventilfunksjonen for driverne. Hastigheten til driverne finner man direkte ut fra åpne og lukkefunksjonen for ventil:

Hastighet sylindre:

$$v(t) = \frac{Q(t)}{A_1} \tag{63}$$

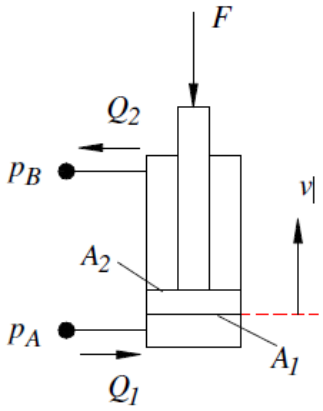
og:

$$v(t) = \frac{Q(t)}{A_2} \tag{64}$$

Hvor:

$Q(t)$ = Oljestrøm $[m^3/s]$

$v(t)$ = hastighet sylinderstempel $[m/s]$.



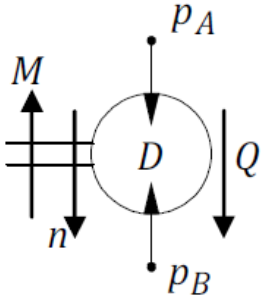
Rotasjonshastighet hydraulisk motor:

$$\omega(t) = \frac{Q(t)}{4 \cdot D} \tag{65}$$

Hvor:

$\omega(t)$ = omdreininger $[rot/s]$.

D = fortrengningsvolum $[m^3/rot]$.



Ved å integrere funksjonene med hensyn på tid finner man posisjon. Arealene, A_1 og A_2 samt fortrengningsvolumet, D, er konstant og endrer seg ikke over tid.

Posisjon hydrauliske sylindre:

$$r(t) = \frac{\int Q(t)}{A} + r_{init} \tag{66}$$

Vinkelposisjon hydraulisk motor:

$$\theta(t) = \frac{\int Q(t)}{4 \cdot D} + \theta_{init} \tag{67}$$

Akselerasjonen til driverne finner man ved å derivere funksjonene med hensyn på tid.

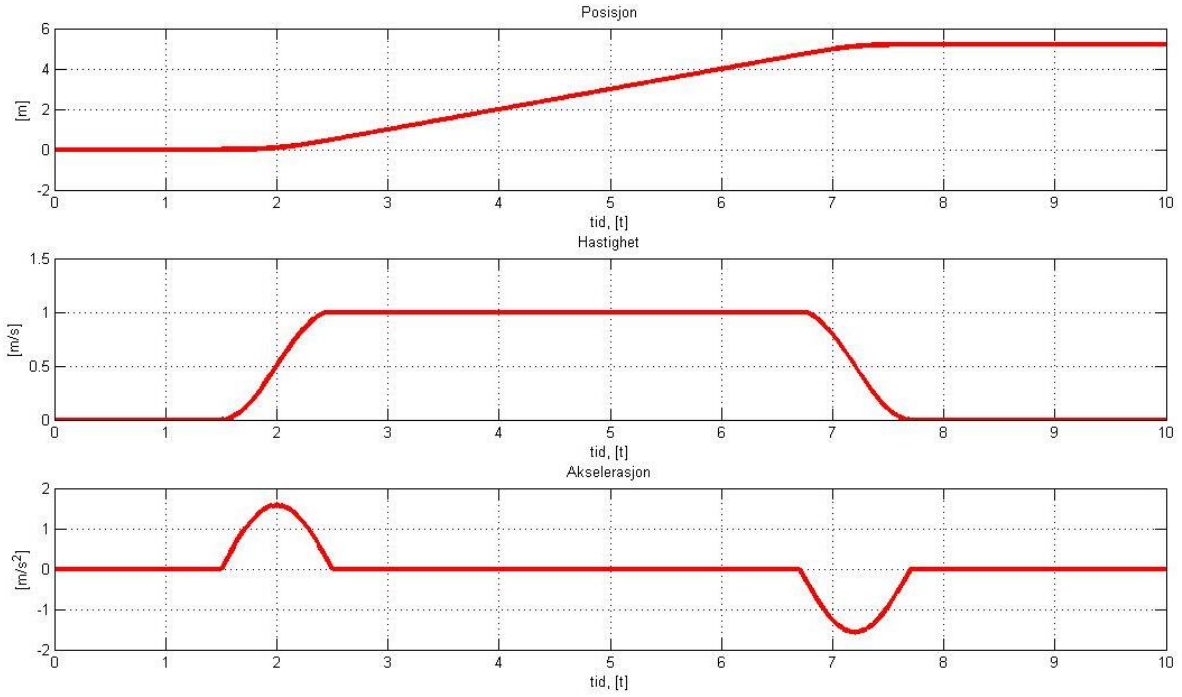
Akselerasjon hydrauliske sylindre

$$a(t) = \frac{Q(t)}{A} dt \tag{68}$$

Vinkelakselerasjon hydraulisk motor:

$$\alpha(t) = \frac{Q(t)}{4 \cdot D} dt \tag{69}$$

Driverfunksjonene bestemmes ut fra nødvendig oljevolum og variablene Δt_1, Δt_2 og Δt_3. Figur 10.4 viser verifisering av funksjonene for posisjon, hastighet og akselerasjon.



Figur 10.4 Posisjon, hastighet og akselerasjonskurve. A=1, D=1, u=1, $\Delta t_1 = 1,5$, $\Delta t_2 = 1$ og $\Delta t_3 = 4$ og Q_max=1.

10.3 Optimering av maksimal oljestrøm

Til optimering brukes optimeringsrutinen fmincon i Matlabs toolbox. Designvariabler for optimering defineres og øvre og nedre grenser for variablene etableres. Oljeforbruket minimeres gjennom objektfunksjonen i fmincon som minimerer funksjonsverdien. Designvariabler er definert i vektor $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} \Delta t_{1,1} \\ \Delta t_{2,1} \\ \Delta t_{3,1} \\ \Delta t_{1,2} \\ \Delta t_{2,2} \\ \Delta t_{3,2} \\ \Delta t_{1,3} \\ \Delta t_{2,3} \\ \Delta t_{3,3} \end{bmatrix} \tag{70}$$

Hvor:

$\Delta t_{1,1}$= Innslagstid for åpning av ventil for syl.1.

$\Delta t_{2,1}$ = Åpne og lukketid for ventil for syl.1.

$\Delta t_{3,1}$ = Intervall for åpen ventil for syl.1.

$\Delta t_{1,2}$ = Innslagstid for åpning av ventil for syl.2

$\Delta t_{2,2}$ = Åpne/lukketid for ventil for syl.2.

$\Delta t_{3,2}$ = Intervall for åpen ventil for syl.2.

$\Delta t_{1,3}$ = Innslagstid for åpning av ventil for syl.3.

$\Delta t_{2,3}$ = Åpne/lukketid for ventil for syl.3.

$\Delta t_{3,3}$ = Intervall for åpen ventil for syl.3.

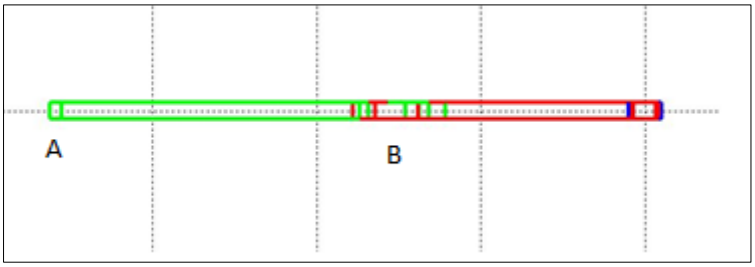
Med bakgrunn i dette kan optimeringsrutinen endre innslagstid, åpne og lukketid, samt intervall for ventil funksjoner. Endring i variablene $\Delta t_{1,i}$, $\Delta t_{2,i}$ og $\Delta t_{3,i}$ gir utslag i ventilpådraget, u, som bestemmer hydraulikkstrømmen. For at optimeringsrutinen skal holde variabler innenfor bestemte grenser er det nødvendig å definere disse grensene. Øvre og nedre grenser for designvariabler defineres:

$$lb = \begin{bmatrix} 0 \\ 0.2 \\ 0 \\ 0 \\ 0.2 \\ 0 \\ 0 \\ 0.2 \\ 0 \end{bmatrix}, \qquad ub = \begin{bmatrix} \Delta t_0 - 2 \cdot \Delta t_{2,1} \\ 3 \\ \Delta t_0 - 2 \cdot \Delta t_{2,1} \\ \Delta t_0 - 2 \cdot \Delta t_{2,2} \\ 3 \\ \Delta t_0 - 2 \cdot \Delta t_{2,2} \\ \Delta t_0 - 2 \cdot \Delta t_{2,3} \\ 3 \\ \Delta t_0 - 2 \cdot \Delta t_{2,3} \end{bmatrix} \tag{71}$$

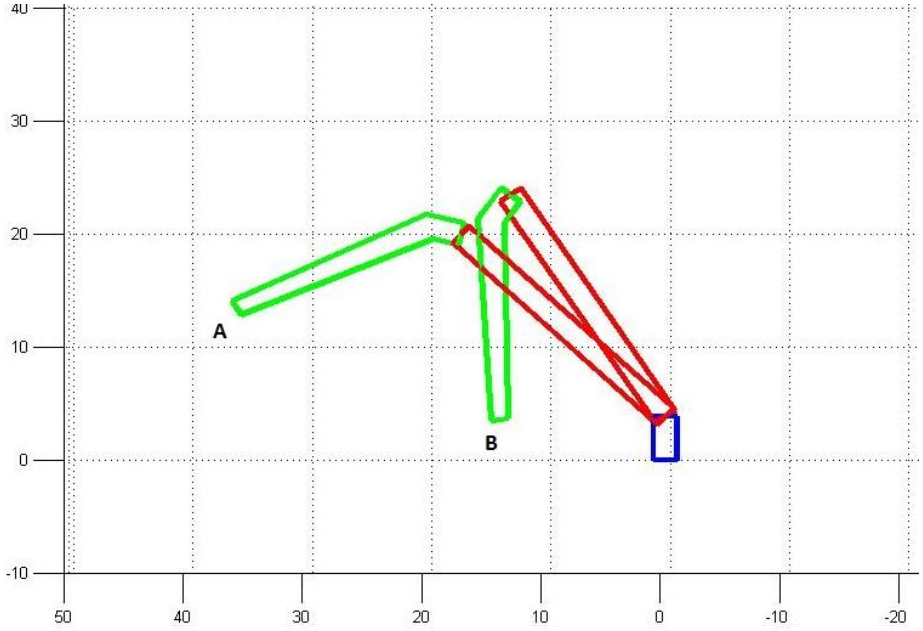
Hvor:

$\Delta t_0 =$ Kjøretid for kranen.

Ved å optimalisere maksimal oljestrøm ved fast kjøretid kan man verifisere om optimeringsrutinen fungerer som ønsket. Figur 10.7 viser resultatet av kjøring fra A til B for minimering av oljestrøm. Start posisjon A er satt til koordinat [0 35 15], sluttposisjon B er satt til koordinat [0 15 5]. Start og sluttposisjon vises i Figur 10.6.

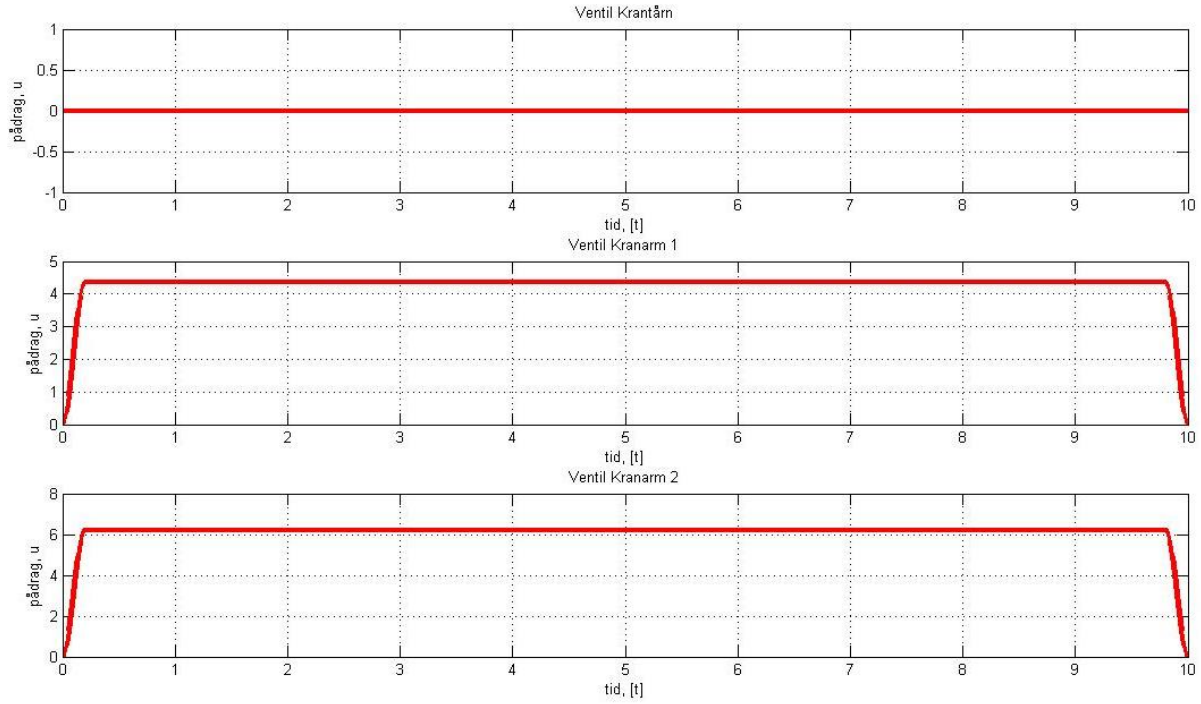


Figur 10.5 Startposisjon A og sluttposisjon B sett ovenfra



Figur 10.6 Startposisjon A og sluttposisjon B

Kranen kjører fra A til B på $\Delta t_0 = 10\ [s]$. Optimerings resultat vises i Figur 10.7 nedenfor.



Figur 10.7 Optimalt ventilpådrag for minimering av volumstrøm.

Figur 10.7 viser at ventilene er åpne hele den tilgjengelige tiden. Designvariablene, $\Delta t_{1,2}$ og $\Delta t_{1,3}$, som angir innslagstiden for ventilene er begge lik null. Designvariablene, $\Delta t_{2,2}$ og $\Delta t_{2,3}$, som bestemmer åpne og lukkeintervallet er på 0,2 sekunder, som er minimal grenseverdi. Optimeringen av ventilfunksjonen gir minimal oljestrøm gjennom ventiler for den bestemte kjøringen. Ventil for krantårn åpner aldri. Dette er fordi ønskede koordinat posisjoner for start og sluttposisjon ikke gir rotasjon av krantårn. Ventilpådraget er høyere enn den tillatte maksimalgrensen for ventilene. Dette er fordi det ikke er satt noen grense for pådrag i optimeringen for oljestrøm. Det er heller ikke meningen på dette stadiet. Figur 10.7 viser at optimeringen følger de grenser som er satt for variablene og finner gode verdier for formålet.

10.4 Optimering av kranens kjøretid

Det vil være mer designmessig interesse i å optimere tiden kranen trenger for å kjøre en strekning fra A til B. Kjøretiden, Δt_0, endres til å være en designvariabel i optimeringsrutinen. Kjøretiden, Δt_0, er tiden kranen trenger for å kjøre den bestemte strekningen. Designvariabler for optimeringen er definert i vektor $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} \Delta t_0 \\ \Delta t_{1,1} \\ \Delta\, t_{2,1} \\ \Delta t_{3,1} \\ \Delta t_{1,2} \\ \Delta t_{2,2} \\ \Delta t_{3,2} \\ \Delta t_{1,3} \\ \Delta t_{2,3} \\ \Delta t_{3,3} \end{bmatrix} \tag{72}$$

Innføringen av designvariabel for kjøretid, Δt_0, gjør at grenser for designvariablene, $\Delta t_{1,i}$, $\Delta t_{2,i}$ og $\Delta t_{3,i}$ er lineært avhengig av variabel Δt_0. Disse grensene må derfor defineres gjennom matrise på formen:

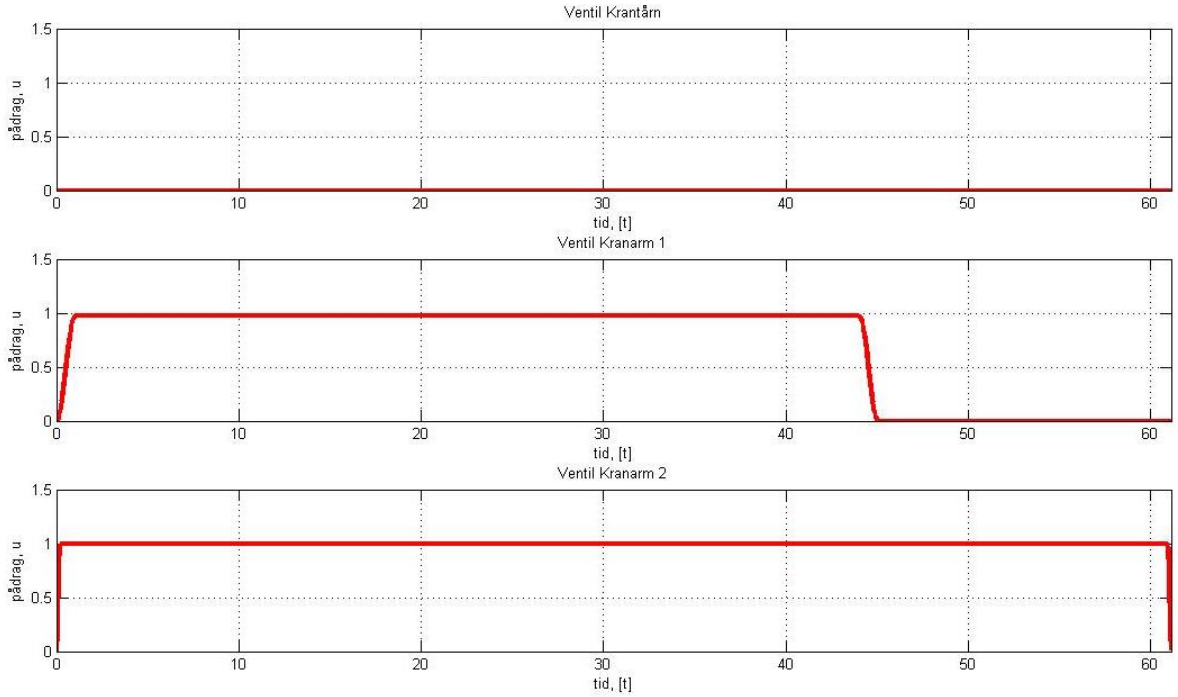
$$\boldsymbol{Ax} \leq \boldsymbol{b} \tag{73}$$

Dette gir følgende matrise for grenser:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta t_0 \\ \Delta t_{1,1} \\ \Delta t_{2,1} \\ \Delta t_{3,1} \\ \Delta t_{1,2} \\ \Delta t_{2,2} \\ \Delta t_{3,2} \\ \Delta t_{1,3} \\ \Delta t_{2,3} \\ \Delta t_{3,3} \end{bmatrix} \leq \begin{bmatrix} -5 \\ 0 \\ 0 \\ -0.2 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.2 \\ 3 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.2 \\ 3 \\ 0 \\ 0 \end{bmatrix} \tag{74}$$

I denne optimeringen prøver optimeringsrutinen å finne raskest kjøring fra A til B, med hensyn til grenser for designvariabler og designkrav. Det betyr at optimeringsrutinen prøver å minimere designvariabel Δt_0, ved hjelp av designvariablene den har tilgjengelig og innenfor de designkrav som er satt. I tillegg til øvre og nedre grenser av designvariabler er det satt krav til at ventiler ikke kan slippe gjennom mer volumstrøm enn 900 [l/min]. Dette design kravet er hentet ut fra datablad over ventilspesifikasjoner. Kranen kjører fra startposisjon A med koordinat [0 35 15] til posisjon B

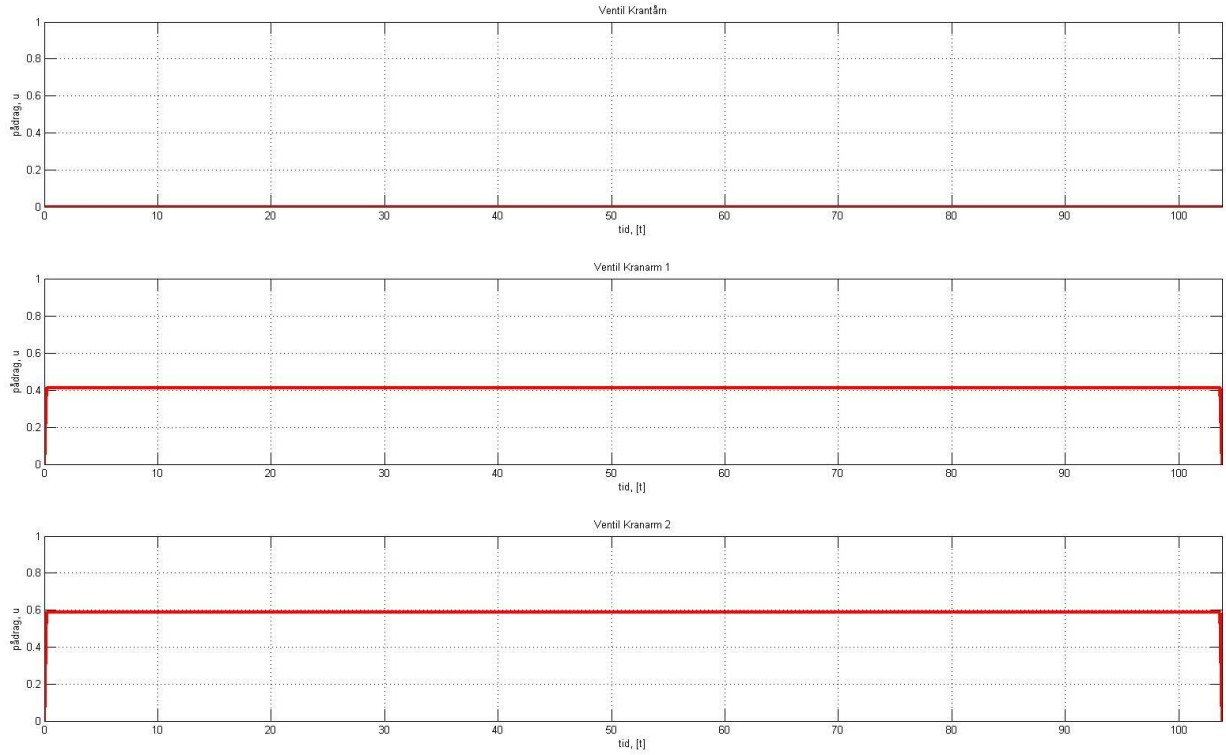
koordinat [0 15 5]. Dette er samme start og sluttposisjon som før og kan ses i Figur 10.5 og Figur 10.6. Figur 10.8 viser resultatet av optimeringen.



Figur 10.8 Ventilpådrag

Figur 10.8 viser at kranen bruker 61,114 sekunder på å utføre bevegelsen fra A til B. Optimeringen er bundet til ikke å bruke mer volumstrøm enn det ventilene kan slippe gjennom. Dette kravet opprettholdes i at ventilpådrag stopper ved verdi 1, som er maksimal åpning av ventil. Krantårnet står i ro, og gir ikke noe ventilpådrag. Ventil for kranarm 2 bruker hele sin tilgjengelige tid. Åpne og lukketiden for ventilen, $\Delta t_{2,3}$, er 0,2 sekunder, som betyr at optimeringen finner raskest kjøring fra A til B. Siden optimeringsrutinen bare minimerer maksimaltid innenfor bestemte restriksjoner er det godtatt at ventil for kranarm 1 oppfører seg som den gjør.

PDPH kranen er avhengig av oljetrykk fra oljeriggens ringline-system. Her er også andre verktøy tilkoblet. Alle verktøy er avhengig av å ha oljetrykk tilgjengelig. Derfor er det satt begrensning på kranens maksimale oljeforbruk, $Q_{max} = 900\ [\frac{l}{s}]$. Dette betyr at en ventil med maksimalt pådrag kan bruke opp hele oljeforsyningen. Dette betyr at man må kjøre kranen slik at ventilene til sammen ikke bruker over maksimal grense. Disse nye grensene legges inn i optimeringsrutinen. Figur 10.9 viser resultatet av simuleringen.



Figur 10.9 Ventilpådrag, optimering av kjøretid med begrensning på maksimal ventilgjennomstrømning og oljeforbruk. $Q_{max,ventil} = 900$ [$\frac{l}{min}$], $Q_{max} = 900$ [$\frac{l}{min}$].

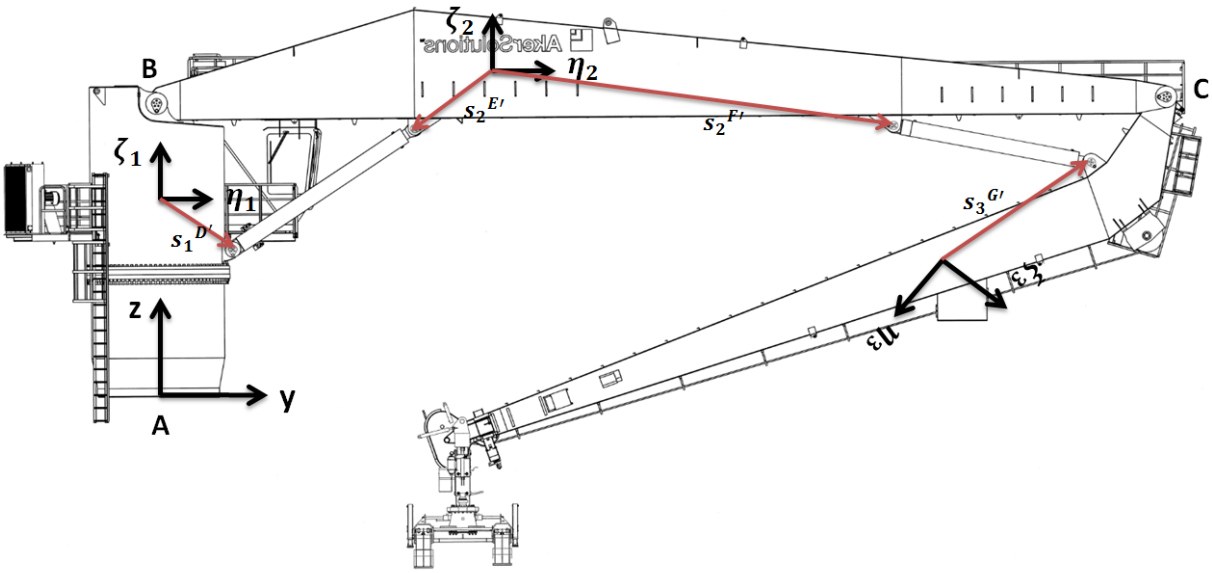
Kranen bruker 103,817 sekunder på å kjøre fra A til B. De to ventiler åpner nå slik at maksimalt oljeforbruk ikke overgås. Ventil for kranarm 1, u_2, åpner 41,2 % og ventil for kranarm 2, u_3, åpner 58,8 %. Dette gir et maksimalt oljeforbruk på 900 [l/min] som er maksimalgrensen. Man kan nå anta at designvariabler og grenser er gode for modellen. Dette betyr variablene gir gode driverfunksjoner til kranen. Simuleringer er gjort med et stort stempelareal for sylindre. Dette er gjort for å oppnå maksimal volumstrøm slik at man kunne kontrollere begrensningene som var satt på volumstrøm. Senere simuleringer er gjort med reelt sylinder areal, og gir derfor raskere kjøring fra A til B. Informasjon om hydrauliske sylindre er hentet i tekniske tegninger.

10.5 Optimering av plassering for hydrauliske sylindre

Tidligere er det etablert en modell for optimering av kjøretid fra posisjon A til posisjon B. Det neste optimeringsforsøket optimerer sylinderplasseringen med hensyn på krefter. Objektfunksjonen til optimeringen er:

$$f(x) = F_{syl1} + F_{syl2} \tag{75}$$

For å forenkle modellen optimeres det bare med hensyn til konstruksjon, optimering av kjøretid tas ikke med. Endring av konstruksjon for kranen gjør at vektorposisjoner i kinematikkmodellen endres ved hjelp av designvariabler. Vektorkoordinat $s_1{}^{D\prime}$, $s_2{}^{E\prime}$, $s_2{}^{F\prime}$ og $s_3{}^{G\prime}$ for sylinderplassering innføres som designvariabler. Figur 10.10 viser sylinderposisjon.



Figur 10.10 Vektorkoordinat $s_1{}^{D\prime}$, $s_2{}^{E\prime}$, $s_2{}^{F\prime}$ og $s_3{}^{G\prime}$ for optimering sylinderplassering.

Dette gjør at optimeringsrutinen kan finne en ny sylinderposisjon som gir lavere funksjonsverdi for krefter. Designvariabler for optimeringen er definert i vektor $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} \tag{76}$$

Øvre grense for designvariabler er bestemt til 1.2 og nedre grense til 0.8. Designvariablene multipliseres med vektorposisjoners y og z koordinat. Endring i designvariabler gir endring i posisjon for sylindre.

$$s_1{}^{D\prime}{}_{y,new} = s_1{}^{D\prime}{}_{y,init} \cdot x_1 \tag{77}$$

$$s_1{}^{D\prime}{}_{z,new} = s_1{}^{D\prime}{}_{z,init} \cdot x_2 \quad (78)$$

$$s_2{}^{E\prime}{}_{y,new} = s_2{}^{E\prime}{}_{y,init} \cdot x_3 \quad (79)$$

$$s_2{}^{E\prime}{}_{z,new} = s_2{}^{E\prime}{}_{z,init} \cdot x_4 \quad (80)$$

$$s_2{}^{F\prime}{}_{y,new} = s_2{}^{F\prime}{}_{y,init} \cdot x_5 \quad (81)$$

$$s_2{}^{F\prime}{}_{z,new} = s_2{}^{F\prime}{}_{z,init} \cdot x_6 \quad (82)$$

$$s_3{}^{G\prime}{}_{y,new} = s_3{}^{G\prime}{}_{y,init} \cdot x_7 \quad (83)$$

$$s_3{}^{G\prime}{}_{z,new} = s_3{}^{G\prime}{}_{z,init} \cdot x_8 \quad (84)$$

For å oppnå realistiske resultat for optimering av sylinderplassering er det nødvendig å innføre design krav. Maksimal og minimal lengde for sylindre begrenses. Sylinderlengder er hentet i tekniske tegninger, og gir følgende begrensninger for posisjon.

$$5075\ [mm] \leq syl_1 \leq 9050\ [mm] \quad (85)$$

$$4670\ [mm] \leq syl_2 \leq 8240\ [mm] \quad (86)$$

Kranen optimeres med hensyn på sylinderkraft. Det maksimale oljetrykket en kan belaste ringlinesystemet med er 210 [bar]. Dette gir restriksjoner for maksimal sylinderkraft. Sylinderkrefter beregnes ut fra formelen:

$$F = \frac{p_A A_1 - p_B A_2}{10} \quad (87)$$

Hvor:

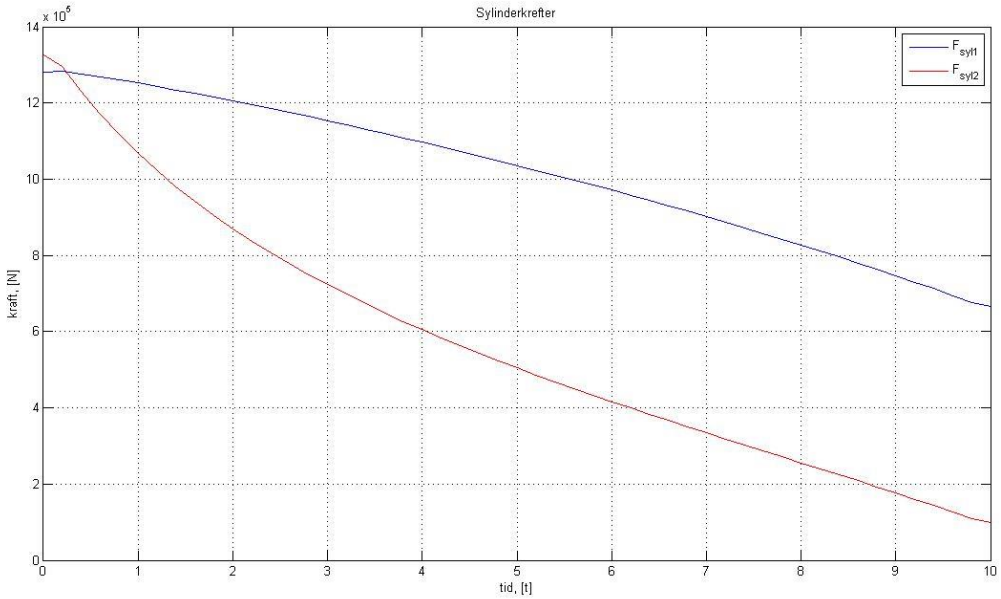
p_A= trykk på stempelsiden av hydraulikksylinderen.

p_B= trykk på sylindersiden av hydraulikksylinderen.

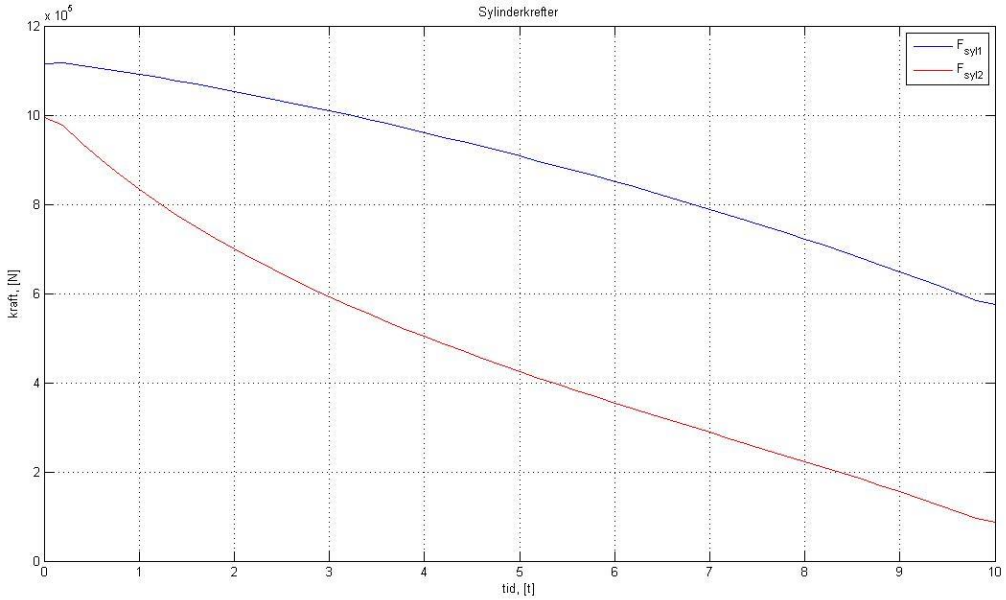
A_1 = Areal stempelside.

A_2 = Areal sylinderside.

Returtrykk for sylindere er bestemt til 15 bar. Dette gir en maksimalkraft for sylinder 1 lik 1630 [kN]. Sylinder 2 opereres i begge kjøreretninger og har maksimalt strekk og trykk for sylinder lik 1425 [kN] og −719 [kN]. Kjøretiden er konstant lik 20 sekunder og kranen løfter en last på 10 tonn. Figur 10.11 viser sylinderkrefter ved fast sylinderposisjon og Figur 10.12 viser sylinderkrefter ved optimert sylinderposisjon.



Figur 10.11 Sylinderkrefter F_{syl1} og F_{syl2} ved fast sylinderposisjon



Figur 10.12 Sylinderkrefter F_{syl1} og F_{syl2} ved optimering av sylinderposisjon

Som man kan se på de to figurene senkes sylinderkreftene ved optimering av posisjon. Maksimal sylinderkraft for sylinder 1 senkes med omlag 200 [kN]. Maksimal sylinderkraft for sylinder 2 senkes med omlag 300 [kN]. Dette betyr at optimeringen finner gode resultat for sylinderplassering. Ved å studere resultatet av designvariabler for optimeringen kommer konstruksjonsendringen klarere frem.

$$\boldsymbol{x} = \begin{bmatrix} 1.20 \\ 1.20 \\ 0.80 \\ 1.20 \\ 0.98 \\ 1.20 \\ 0.92 \\ 0.80 \end{bmatrix} \tag{88}$$

Vektorkoordinatene til sylinderfestene er flyttet slik at sylindrene får en bedre arbeidsvinkel. Nedenfor vises vektorkoordinat for sylinderplassering ved flytting fra opprinnelig posisjon til optimert posisjon:

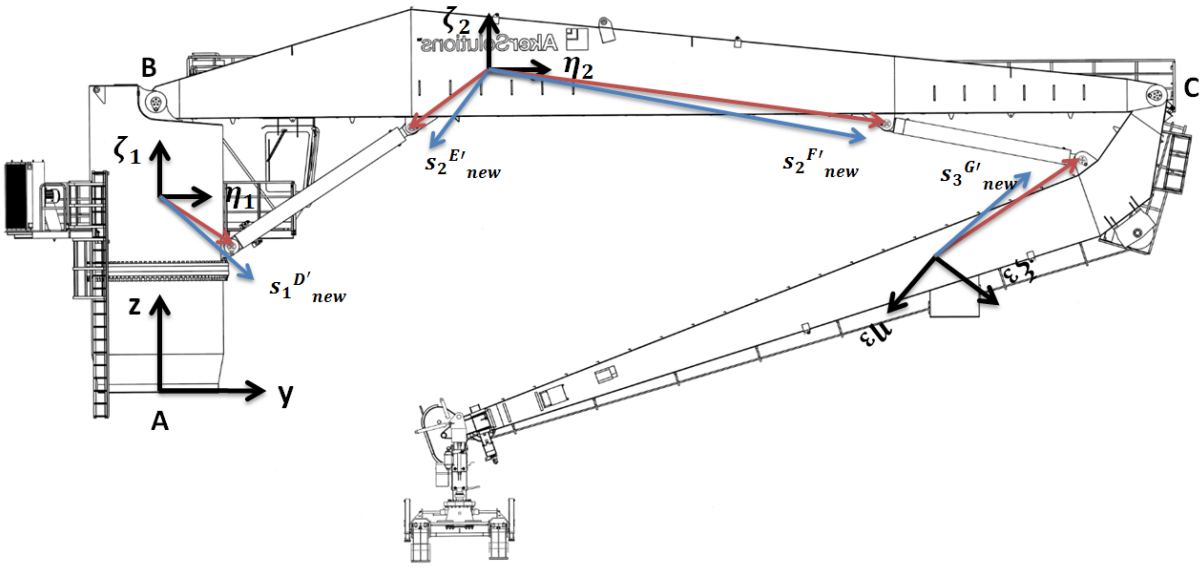
$\boldsymbol{s_1}^{\boldsymbol{D'}}{}_{\boldsymbol{init}}$ = [0 1.75 -1.47], $\boldsymbol{s_1}^{\boldsymbol{D'}}{}_{\boldsymbol{new}}$ = [0 2.1 -1.76]

$\boldsymbol{s_2}^{\boldsymbol{E'}}{}_{\boldsymbol{init}}$ = [0 -4.66 -1], $\boldsymbol{s_2}^{\boldsymbol{E'}}{}_{\boldsymbol{new}}$ = [0 -3.73 -1.2]

$\boldsymbol{s_2}^{\boldsymbol{F'}}{}_{\boldsymbol{init}}$ = [0 6,48 -1], $\boldsymbol{s_2}^{\boldsymbol{F'}}{}_{\boldsymbol{new}}$ = [0 6,35 -1.2]

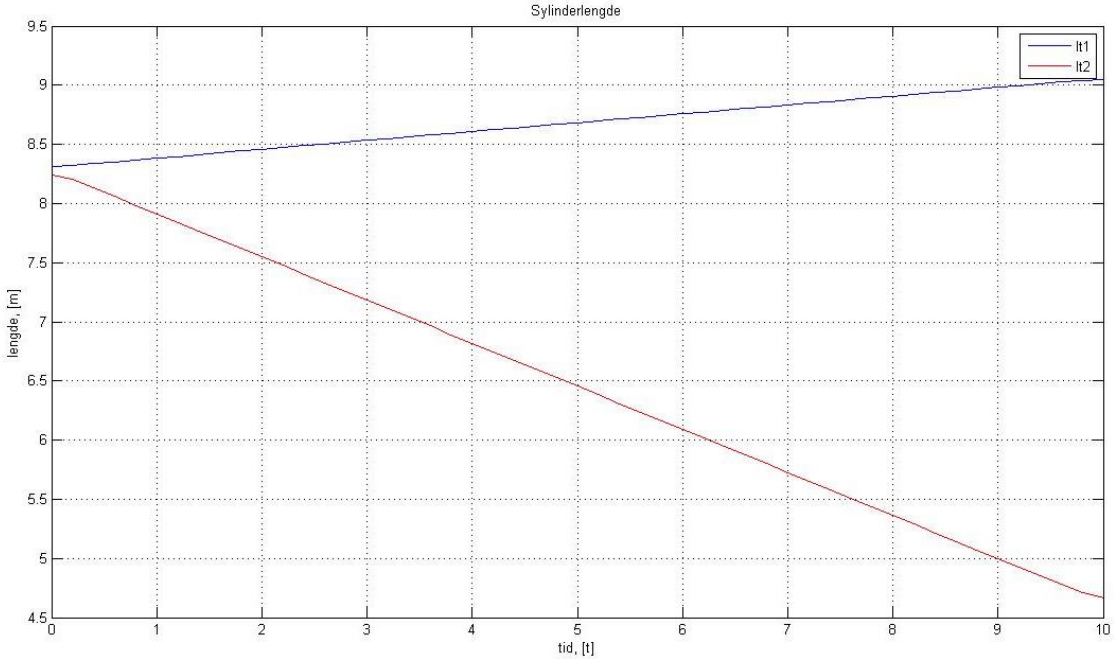
$\boldsymbol{s_3}^{\boldsymbol{G'}}{}_{\boldsymbol{init}}$ = [0 -4,80 2,24], $\boldsymbol{s_3}^{\boldsymbol{G'}}{}_{\boldsymbol{new}}$ = [0 -4,41 1,79]

Endringene er skissert med blå piler i Figur 10.13.



Figur 10.13 Optimert sylinderposisjon markert med blå piler

Figur 10.14 viser at optimeringen holder seg innenfor sine designgrenser for sylinderlengder

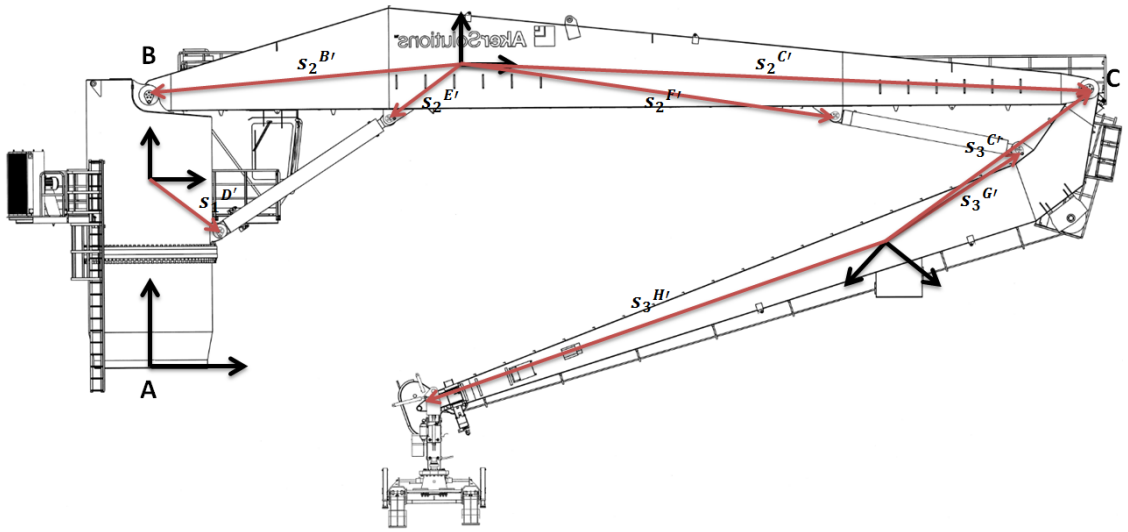


Figur 10.14 Sylinder lengder ved kjøring frå A til B

Endringer som gjøres i sylinderplassering har innvirkning for hele krankonstruksjonen. Reaksjonskrefter og bøyemoment for kranarmer endres, det gjør også kranens rekkevidde og kjørehastighet. En god konstruksjon må overholde alle disse designkravene.

10.6 Optimering av kranens konstruksjon

Teknikken som ble brukt for å endre sylinderplassering for kranen kan også brukes til å endre størrelser for kranlegemer. Dette betyr at det er mulig å endre kranens konstruksjon. Endring av konstruksjon for kranen gjør at vektorkoordinat $\boldsymbol{s_1}^{D\prime}$, $\boldsymbol{s_2}^{E\prime}$, $\boldsymbol{s_2}^{F\prime}$, $\boldsymbol{s_3}^{G\prime}$, $\boldsymbol{s_2}^{B\prime}$, $\boldsymbol{s_2}^{C\prime}$, $\boldsymbol{s_3}^{C\prime}$ og $\boldsymbol{s_3}^{H\prime}$ endres ved hjelp av designvariabler. Figur 10.10 viser vektorkoordinat som kan endres i optimeringen.



Figur 10.15 Vektorkoordinat $\boldsymbol{s_1}^{D\prime}$, $\boldsymbol{s_2}^{E\prime}$, $\boldsymbol{s_2}^{F\prime}$, $\boldsymbol{s_3}^{G\prime}$, $\boldsymbol{s_2}^{B\prime}$, $\boldsymbol{s_2}^{C\prime}$, $\boldsymbol{s_3}^{C\prime}$ og $\boldsymbol{s_3}^{H\prime}$ for optimering av krankonstruksjon

Designvariabler for optimeringen er definert i vektor $\boldsymbol{x}$:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix} \tag{89}$$

Designvariabel x_9 gir endring i kranarm 1, og designvariabel x_{10} endrer kranarm 2. Ved å multiplisere alle vektorkoordinater for et kranlegeme med sin tilhørende designvariabel endres legemet proporsjonalt.

$$\boldsymbol{s_2}^{\boldsymbol{B\prime}}{}_{new} = \boldsymbol{s_2}^{\boldsymbol{B\prime}}{}_{init} \cdot x_9 \tag{90}$$

$$\boldsymbol{s_2}^{\boldsymbol{C\prime}}{}_{new} = \boldsymbol{s_2}^{\boldsymbol{C\prime}}{}_{init} \cdot x_9 \tag{91}$$

$$\boldsymbol{s_2}^{\boldsymbol{E\prime}}{}_{new} = \boldsymbol{s_2}^{\boldsymbol{E\prime}}{}_{init} \cdot x_9 \tag{92}$$

$$\boldsymbol{s_2}^{\boldsymbol{F\prime}}{}_{new} = \boldsymbol{s_2}^{\boldsymbol{F\prime}}{}_{init} \cdot x_9 \tag{93}$$

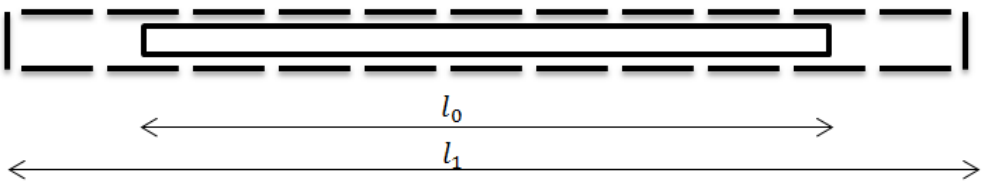
$$\boldsymbol{s_3}^{\boldsymbol{C\prime}}{}_{new} = \boldsymbol{s_3}^{\boldsymbol{C\prime}}{}_{init} \cdot x_{10} \tag{94}$$

$$\boldsymbol{s_3}^{\boldsymbol{G\prime}}{}_{new} = \boldsymbol{s_3}^{\boldsymbol{G\prime}}{}_{init} \cdot x_{10} \tag{95}$$

$$\boldsymbol{s_3}^{\boldsymbol{H\prime}}{}_{new} = \boldsymbol{s_3}^{\boldsymbol{H\prime}}{}_{init} \cdot x_{10} \tag{96}$$

Designvariabler, $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ og x_8 for sylinderplassering er implementert som i kapittel 10.5. Matlab programmet er nå i stand til å endre krankonstruksjon for å tilfredsstille designkrav.

Ved å endre størrelse og utforming av kranarmer endres også kranarmenes masse og treghetsmoment. For at optimering av krankonstruksjon skal være reel må denne endringen tas med. Reaksjonskrefter for kranen beregnes ved hjelp av Jacobian matrisen. For at resultatene skal være riktige må de lokale koordinatsystemene ligge i legemets massesenter. Dette betyr at ved å endre kranarmers oppbygning endrer man også massesenterets plassering i kranarmer. Designvariabler x_9 og x_{10} gir proporsjonal endring i kranarm 1 og kranarm 2. Det gjør at en kan anta at massesenteret posisjonerer seg korrekt i kranarmer. Endringen i krankonstruksjon på grunn av optimering av sylinderposisjon neglisjeres. Det antas at treghetsmomentet for kranarmer endres eksponentielt. Bakgrunnen for denne antagelsen er vist gjennom utledning av treghetsmoment for en bjelke som forstørres proporsjonalt. Figur 10.16 skisserer en bjelke som forstørres. Stiplet linje viser forstørrelsen.



Figur 10.16

Endring av treghetsmoment for en bjelke:

$$I_{ny} = \left(\frac{l_1}{l_0}\right)^2 \cdot I_{init} \tag{97}$$

Hvor:

I =treghetsmoment [kg · m^2].

l_1= lengde opprinnelig bjelke [kg].

l_1 =lengde til forstørret bjelke [kg].

Treghetsmoment for rektangulær bjelke er definert som:

$$I_{init} = \frac{1}{12} \cdot m_0 \cdot {l_0}^2 \tag{98}$$

$$I_{ny} = \frac{1}{12} \cdot m_1 \cdot {l_1}^2 \tag{99}$$

Hvor:

m=massen til bjelken [kg].

Massen til legemet beregnes ved hjelp av densiteten til bjelkematerialet:

$$m_0 = l_0 \cdot \rho \tag{100}$$

$$m_1 = l_1 \cdot \rho \tag{101}$$

Hvor:

ρ=Densitet [kg/m]

Forholdet mellom nytt og opprinnelig treghetsmoment:

$$\frac{I_{ny}}{I_{init}} = \frac{\frac{1}{12} \cdot l_1 \cdot \rho \cdot {l_1}^2}{\frac{1}{12} \cdot l_0 \cdot \rho \cdot {l_0}^2} = \left(\frac{l_1}{l_0}\right)^3 \tag{102}$$

Dette viser at det er rimelig å anta at treghetsmomentet til kranarmene endres eksponentielt som funksjon av endring i designvariablene, ${x_9}^3$ for kranarm 1 og ${x_{10}}^3$ for kranarm 2. Også massen til kranlegemene endrer seg ved endring i konstruksjon.

$$\frac{m_1}{m_0} = \frac{l_1 \cdot \rho}{l_0 \cdot \rho} = \frac{l_1}{l_0} \tag{103}$$

$$m_1 = \left(\frac{l_1}{l_0}\right) \cdot m_0 \tag{104}$$

Massen til kranlegemer endrer seg lineært med sine designvariabler, x_9 for kranarm 1 og x_{10} for kranarm 2.

En analyse gjøres for å validere om implementering av optimering for kranlegemer fungerer. Objektsfunksjonen for optimeringen settes til å maksimere kranarmer ved kjøring fra A til B.

$$f(x) = -x_9 - x_{10} \tag{105}$$

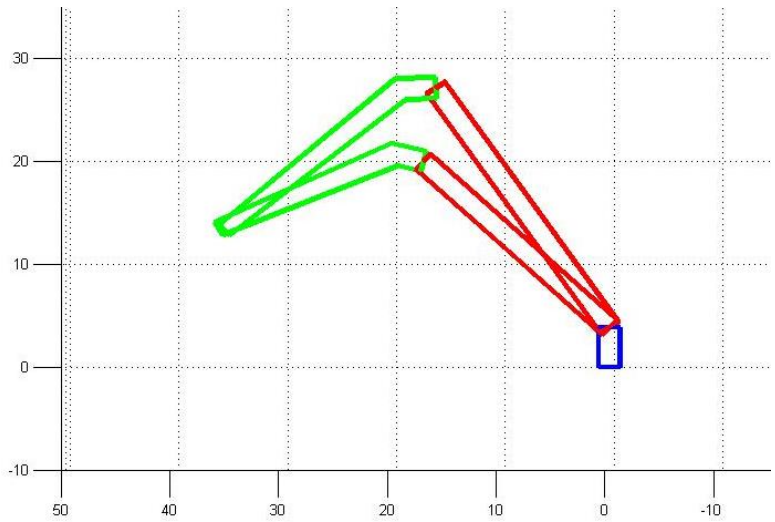
Øvre og nedre grenser for designvariabler bestemmes:

$$lb = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \end{bmatrix}, \qquad ub = \begin{bmatrix} 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \end{bmatrix} \tag{106}$$

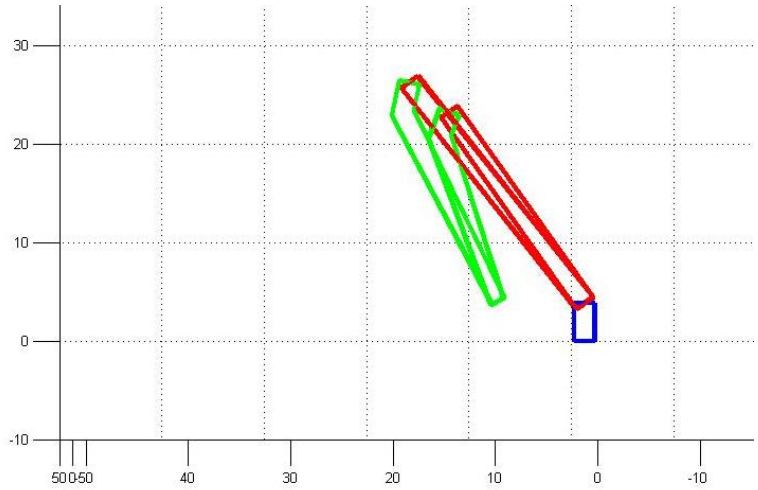
Resultatet av optimeringen vises i Figur 10.17 og Figur 10.18. Designvariabler for optimeringen er:

$$\boldsymbol{x} = \begin{bmatrix} 1.01 \\ 0.96 \\ 1.12 \\ 1.01 \\ 1.04 \\ 1.00 \\ 1.03 \\ 0.94 \\ 1.20 \\ 1.20 \end{bmatrix} \tag{107}$$

Resultatet viser at designvariabler for endring av kranlegemer, x_9 og x_{10} når sine bestemte maksimalgrenser. Kranarm 1 og 2 forstørres med faktor lik 1.2.



Figur 10.17 maksimering av kranarmer, startposisjon A



Figur 10.18 maksimering av kranarmer, sluttposisjon B

En ny optimeringstest utføres. Øvre grense heves for designvariabler x_9 og x_{10} for å kontrollere at alle grenser for modellen overholdes.

$$lb = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \end{bmatrix}, \qquad ub = \begin{bmatrix} 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 1.2 \\ 2.0 \\ 2.0 \end{bmatrix} \tag{108}$$

Resultatet viser at designkrav for sylinder-lengder holder forstørringen av kranlegemer tilbake. Resultatet av optimeringen vises i designvariablene for optimeringen.

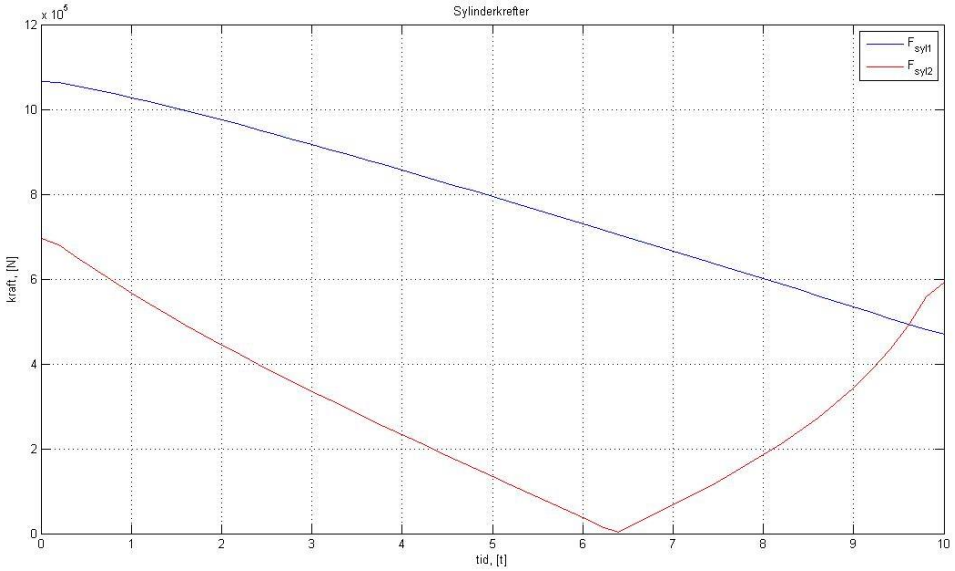
$$\boldsymbol{x} = \begin{bmatrix} 1.20 \\ 0.80 \\ 1.20 \\ 1.20 \\ 1.07 \\ 1.02 \\ 1.20 \\ 0.91 \\ 1.36 \\ 1.49 \end{bmatrix} \tag{109}$$

Dette betyr at modellen endrer sin konstruksjon innenfor grenser som er satt. En komplett optimeringsmodell bør inneholde mulighet for å velge andre typer sylindre, som kan gi kranen større rekkevidde og lastekapasitet.

Det er nå dokumentert at optimering av kranarmer med hensyn til designkrav og grenser fungerer. Det er videre ønskelig å studere om hele kranmodellen endrer seg ved ny optimering av maksimale sylinderkrefter. Objektsfunksjonen settes til å optimere sylinderkrefter.

$$f(x) = F_{syl1} + F_{syl2} \tag{110}$$

Resultatet av analysen vises i Figur 10.19.



Figur 10.19 Sylinderkrefter F_{syl1} og F_{syl2} ved optimering av sylinderposisjon og kranarmer med endring i treghetsmoment.

Figur 10.19 viser at optimeringsrutinen finner lavere funksjonsverdi, $f(x)$, for maksimale sylinderkrefter. Kranarmene endres proporsjonalt med designvariablene $x_9 = 1.1928$ for kranarm 1 og designvariabel $x_{10} = 0.7747$ for kranarm 2. Designvariabler for optimeringen er:

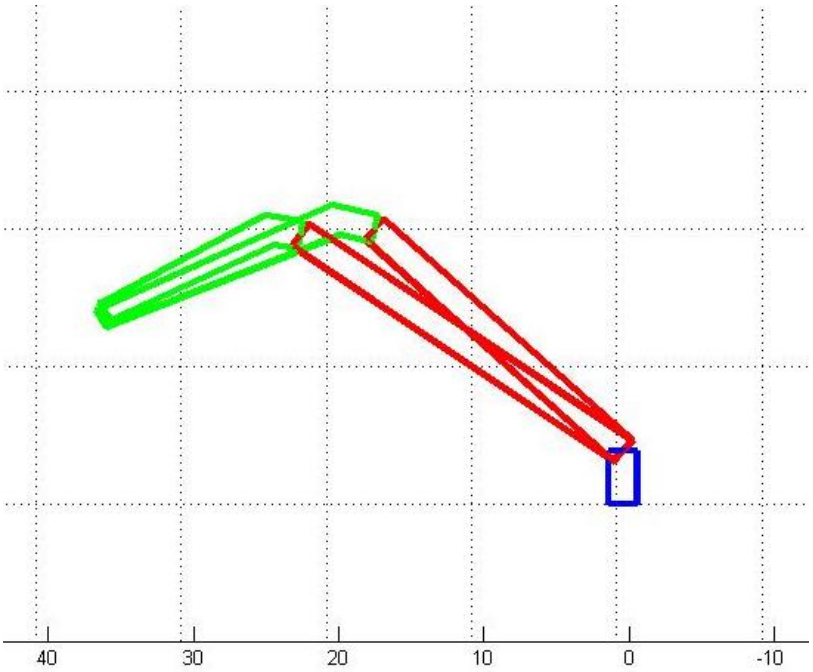
$$\boldsymbol{x} = \begin{bmatrix} 1.20 \\ 1.20 \\ 0.88 \\ 1.20 \\ 1.11 \\ 1.20 \\ 0.86 \\ 0.80 \\ 1.19 \\ 0.77 \end{bmatrix} \tag{111}$$

Den marginalt lavere sylinderkraften, F_{syl1}, er fordi endringen i kranarm 1 gjør optimeringen i stand til å finne enda bedre arbeidsvinkel for sylinderen. Dette er fordi sylinderplasseringen er en funksjon av koordinatposisjon $\boldsymbol{s_2}^{\boldsymbol{E\prime}}$.

$$s_2{}^{E\prime}{}_{y,new} = s_2{}^{E\prime}{}_{y,init} \cdot x_3 \tag{112}$$

$$s_2{}^{E\prime}{}_{z,new} = s_2{}^{E\prime}{}_{z,init} \cdot x_4 \tag{113}$$

Figur 10.20 viser endringen av konstruksjon for optimeringen.



Figur 10.20 Optimering av kranarmer, startposisjon A

Det å optimere kranlegemer med hensyn til å minimere funksjonsverdien for maksimale sylinderkrefter gir ikke et optimalt krandesign, men det viser at optimeringen av den komplette kranmodellen fungerer bra.

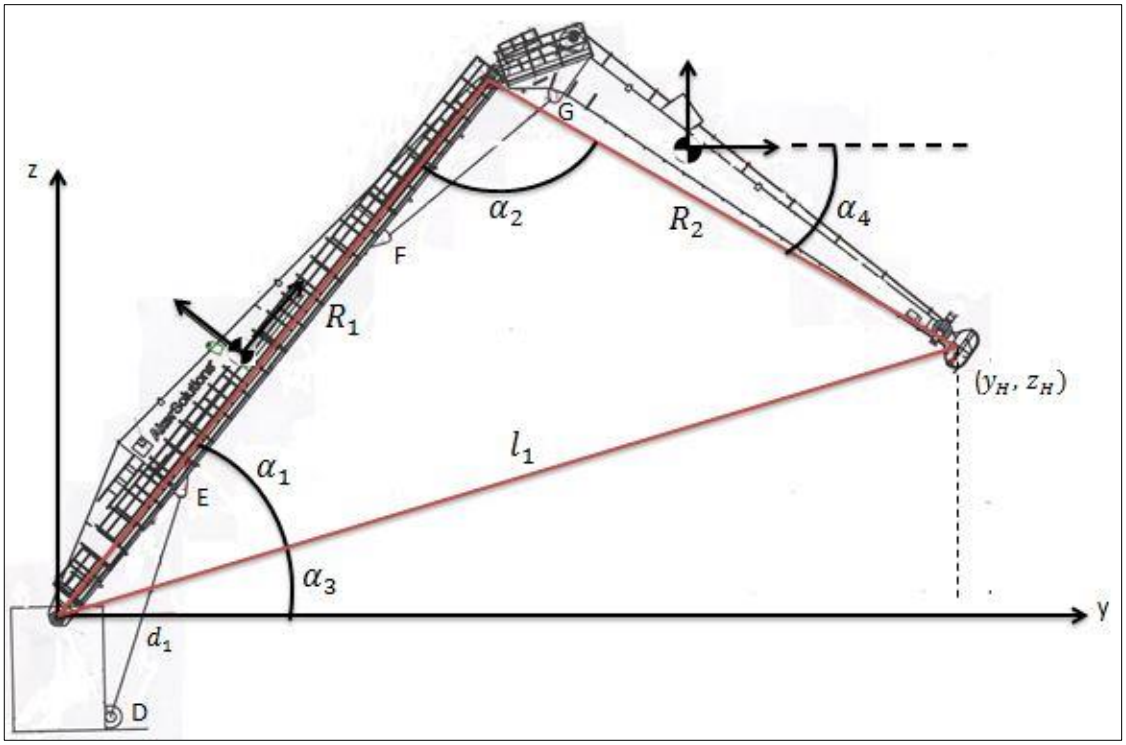
Matlab programmet finner løsning for optimal design av krankonstruksjon basert på de designkrav og grenser for designvariabler som er bestemt. Kranarmer kan endre størrelse og sylindre endre sin plassering. Dette gjør at grunnlaget for en mer komplett optimering av kranen er på plass. En endelig optimering av kranens konstruksjon vil være basert på andre designkrav enn det kranen er testet for i denne rapporten. Testene som er utført er gjort for å kontrollere implementeringer i programvare. PDPH kranen er et produkt som skal selges til en kunde. Dette gjør at det endelige krandesignet i stor grad vil være et produkt av pris. Objektsfunksjon, $f(x)$ for krankonstruksjon bør derfor være en kostfunksjon over kranens pris. Kranen optimeres da for billigst mulig design, innenfor bestemte designkrav og grenser.

11 Problemer med Newton Raphson metode

Det er oppdaget problemer ved bruk av Newton Raphson metode for å bestemme startposisjon. Ved dårlige initialverdier for startposisjon finner Newton Raphson løsninger som rent kinematisk er mulig, men som ikke er mulig for krankonstruksjonen å gjennomføre. Figur 11.1 illustrerer problemet.

Figur 11.1 Problemer med løsninger for Newton Raphson metode

Problemet gjør at initialverdier for posisjon må beregnes ved hjelp av håndberegninger. For å lage en rutine som bestemmer nødvendig oljevolum for drivere, uten bruk av Newton Raphson metode, beregnes også sylinderlengder og rotasjon ved hjelp av geometri betraktninger. Dette vil gi en raskere rutine. Figur 11.2 viser geometribetrakninger.



Figur 11.2 Kran geometri

Pytagoras setning brukes for å finne lengden og vinkel fra B, til bestemt startposisjon for krantupp, H.

$$l_1 = \sqrt{{y_H}^2 + {z_H}^2} \tag{114}$$

$$\alpha_3 = tan^{-1}\left(\frac{z_H}{y_H}\right) \tag{115}$$

Cosinussetningen brukes til å finne vinkler i den definerte trekanten.

$$\alpha_1 = cos^{-1}\left(\frac{{R_2}^2 - {R_1}^2 - {l_3}^2}{-2 \cdot R_1 \cdot l_3}\right) \tag{116}$$

$$\alpha_2 = cos^{-1}\left(\frac{{l_3}^2 - {R_1}^2 - {R_2}^2}{-2 \cdot R_1 \cdot R_2}\right) \tag{117}$$

Vinkelorientering for lokalt koordinatsystem mot vektor $\boldsymbol{s_2}^{\boldsymbol{CH'}}$.

$$\alpha_4 = tan^{-1}\left(\frac{{s_2}^{CH\prime}{}_z}{{s_2}^{CH\prime}{}_y}\right) \tag{118}$$

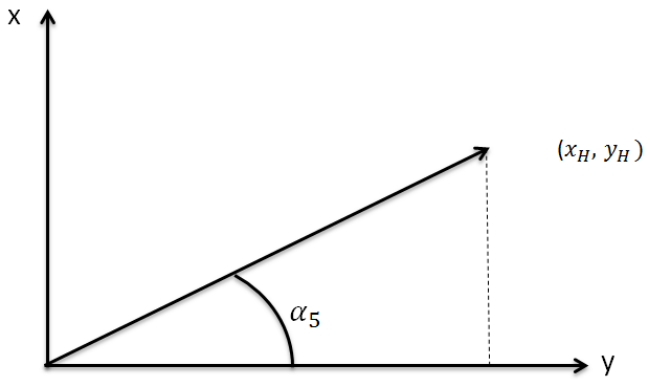
Vinkeldreining for lokalt koordinatsystem 2 om x aksen er:

$$\varphi_{2x} = \alpha_1 + \alpha_3 \tag{119}$$

Vinkeldreining for lokalt koordinatsystem 3 om x aksen er:

$$\phi_{3x} = \alpha_1 + \alpha_3 - \pi + \alpha_2 + \alpha_4 \tag{120}$$

Vinkeldreining om z aksen for koordinatsystem vises i Figur 11.3.



Figur 11.3 Vinkeldreining. (kran illustrert ovenfra)

Pytagoras setning brukes til å finne vinkelen:

$$\alpha_5 = tan^{-1}\left(\frac{x_H}{y_H}\right) \tag{121}$$

Dette gir vinkel rotasjon for lokale koordinat lik:

$$\varphi_{iz} = \alpha_5 \tag{122}$$

Ved hjelp av disse vinklene transformeres lokale vektorer til globale vektorer. Transformerer vektorkoordinat:

$$\boldsymbol{s'_{AB}} = -\boldsymbol{s_1}^{A\prime} + \boldsymbol{s_1}^{B\prime} \tag{123}$$

$$\boldsymbol{s_{AB} = A_1 s_{AB}{}'} \tag{124}$$

$$\boldsymbol{s'_{AD} = -s_1{}^{A\prime} + s_1{}^{D\prime}} \tag{125}$$

$$\boldsymbol{s_{AD} = A_1 s_{AD}{}'} \tag{126}$$

$$\boldsymbol{s'_{BE} = -s_2{}^{B\prime} + s_2{}^{E\prime}} \tag{127}$$

$$\boldsymbol{s_{BE} = A_2 s_{BE}{}'} \tag{128}$$

$$\boldsymbol{s'_{BF} = -s_2{}^{B\prime} + s_2{}^{F\prime}} \tag{129}$$

$$\boldsymbol{s_{BF} = A_2 s_{BF}{}'} \tag{130}$$

Ved hjelp av disse vektorene er det mulig å finne posisjoner for lokale koordinatsystem og posisjoner for sylinderfester.

Posisjon for lokale koordinatsystem beregnes som vist under:

$$\boldsymbol{r_2 = s_{AB} + A_2(-s_2{}^{B\prime})} \tag{131}$$

$$\boldsymbol{r_3 = r_2 + A_2(s_2{}^{C\prime}) + A_3(-s_3{}^{C\prime})} \tag{132}$$

Posisjon sylinderfeste D:

$$\boldsymbol{r_D = s_{AD}} \tag{133}$$

Posisjon sylinderfeste E:

$$\boldsymbol{r_E = s_{AB} + s_{BE}} \tag{134}$$

Posisjon sylinderfeste F:

$$\boldsymbol{r_F = s_{AB} + s_{BF}} \tag{135}$$

Posisjon sylinderfeste G:

$$\boldsymbol{r_G = r_3 + A_3 s_3^{G'}} \tag{136}$$

Ut fra kjente posisjonskoordinater for sylinderfester beregnes sylinderlengder:

$$d_1 = \sqrt{(r_{Ex} - r_{Dx})^2 + (r_{Ey} - r_{Dy})^2 + (r_{Ez} - r_{Dz})^2} \tag{137}$$

$$d_2 = \sqrt{(r_{Gx} - r_{Fx})^2 + (r_{Gy} - r_{Fy})^2 + (r_{Gz} - r_{Fz})^2} \tag{138}$$

Disse kalkulerte startverdiene hjelper kranen til å finne rett utgangsposisjon. Denne teknikken gjør også at det ikke er nødvendig å beregne oljevolum ved bruk av Newton Raphson slik som forklart i kapittel 10.2 Nødvendig oljevolum for kjøring fra startposisjon A til sluttposisjon B bestemmes ved å begegne endring i sylinderlengder og vinkelposisjon.

12 Konklusjon

Det er utviklet et program for konseptgenerering av PDPH kranen. Programmet utfører en dynamisk analyse av kranen og optimerer kranens konstruksjon og kjøretid. Posisjonsanalysen er basert på ulineære bindingsligninger for krankinematikken og beregnes ved hjelp av Newton Raphson metode. Hastighet og akselerasjonsanalyse for kranen beregnes ved hjelp av Jacobian matrise for bindingsligninger. Programmet utfører en dynamisk analyse av reaksjonskrefter. Disse er beregnet ved hjelp av Jacobian invers og Lagranges multiplikator. Kranmodellen er testet gjennom simuleringer, og resultater er validert.

Den validerte kranmodellen er optimalisert med hensyn på konstruksjon og kjørehastighet etter spesifiserte designkrav. Driverfunksjoner for kranen er utviklet for å kjøre kranen en ønsket distanse. Drivere optimaliseres for raskest mulig kjøring fra startposisjon til sluttposisjon. Kranens konstruksjon kan endres for å møte ønskede designspesifikasjoner. Størrelse på kranarmer og sylinderposisjonering optimaliseres for å tilfredsstille designkrav. Endring i krankonstruksjon tar hensyn til endring i treghetsmoment for kranlegemer.

For å finne et reelt optimalt design med hensyn til konstruksjon må det utvikles en kostfunksjon for optimering. Kranens endelige design vil da være basert på en kostnadsanalyse. Denne oppgaven kan danne et grunnlag for en slik videreutvikling.

13 Kilder

Agder, U. i. Kinematic Constraints III. Fronter, Morten Kjeld Ebbesen.

Ben-Israel, A. (1966). "A Newton Raphson Method for the Solution of System Equations." from http://benisrael.net/NEWTON-GI-2.pdf.

Davidsen, B. (2010). "Statikk." from http://ansatte.uit.no/bjorn.davidsen/FysikkNotater/Statikk.pdf.

Nikravesh, P. E. (2008). Planar multibody dynamics, Taylor & Francis Group.

Nikravesh, P. E. (2013). "Reading Assignments." from http://www.u.arizona.edu/~pen/ame553/lessons.html.

Solution, A. (2013). "History." from http://www.akersolutions.com/en/Utility-menu/About-us1/History/.

Vedlegg

Vedlegg 1: Kinematisk modell, Matlab

```
close all
clear all
clc

%Henter konstanter=========================================================
k=konstanter;

h = k(1,13);       %Høyde krantårn (z-retning)
b1 = k(2,13);        %bredde (y-retning)
D1 = k(3,13);        %dybde (x-retning)


s1A = k(1:3,1);      %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);      %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);      %lengde fra lokal koordinat 1 til joint D
s2B = k(1:3,4);      %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5);      %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6);      %lengde fra lokal koordinat 2 til joint E
s2F = k(1:3,7);      %lengde fra lokal koordinat 2 til joint F
s3C = k(1:3,8);      %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9);      %lengde fra lokal koordinat 3 til joint G
s3H = k(1:3,10);     %lengde fra lokal koordinat 3 til joint H

m1 = k(1,17);        %masse krantårn
m2 = k(2,17);        %masse kranarm1
m3 = k(3,17);        %masse kranarm2

M1 = k(1:3,27:29);
M2 = k(1:3,30:32);
M3 = k(1:3,33:35);

J1 = k(1:3,18:20);
J2 = k(1:3,21:23);
J3 = k(1:3,24:26);

v_1 = k(2,15);       %hastighet sylinder 1
v_2 = k(3,15);       %hastighet sylinder 2

a_1 = k(2,16);       %akselerasjon sylinder 1
a_2 = k(3,16);       %akselerasjon sylinder 2

g = k(1:3,36);         %gravitasjonskraft

%===========================================================================
u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';

%initial verdier posisjon   ================================================

q=[-s1A',0,0,0,(-s1A'+s1B'-s2B'),0,0,0,(-s1A'+s1B'-s2B'+s2C'-s3C'),0,0,0]';
```

```
% Initial lengde Sylinder 1 og 2  ======================================
r1=q(1:3,1);    %x1,y1,z1
r2=q(7:9,1);    %x2,y2,z2
r3=q(13:15,1);  %x3,y3,z3

v1 = q(4:6,1);      %v1 (inneholdene phi1, theta1 og tau1)
v2 = q(10:12,1);    %v2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);    %v3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1);         %transformasjonsmatrise for v1
A2 = A(v2);         %transformasjonsmatrise for v2
A3 = A(v3);         %transformasjonsmatrise for v3

d1 = r1 + A1*s1D - A2*s2E - r2;   %lengde longitudinal driver 1
d2 = r2 + A2*s2F - A3*s3G - r3;   %lengde longitudinal driver 2

l_1_init = k(1,43);
l_2_init = k(2,43);


%Simulering ============================================================

t_min= k(1,42);                 %start tid
dt= k(2,42);                    %tidssteg
t_max= k(3,42);                 %slutt tid
t=[t_min:dt:t_max]';
N=size(t,1);                    %antall iterasjoner

Y(1:N,1:18)=0;
Yd(1:N,1:18)=0;
Ydd(1:N,1:18)=0;

YA(1:N,1:6)=0;
YB(1:N,1:6)=0;
YC(1:N,1:6)=0;

Ysyl1(1:N,1:2)=0;
Ysyl2(1:N,1:2)=0;

Y_syl1(1:N,1)=0;
Y_syl2(1:N,1)=0;

Y_d1(1:N)=0;
Y_d2(1:N)=0;

%============================================================

for i=1:N

    r1=q(1:3,1);        %x1,y1,z1
    r2=q(7:9,1);        %x2,y2,z2
    r3=q(13:15,1);      %x3,y3,z3

    v1 = q(4:6,1);      %v1 (inneholdene phi1, theta1 og tau1)
    v2 = q(10:12,1);    %v2 (inneholdene phi2, theta2 og tau2)
    v3 = q(16:18,1);    %v3 (inneholdene phi3, theta3 og tau3)
```

```
    A1 = A(v1);          %transformasjonsmatrise for v1
    A2 = A(v2);          %transformasjonsmatrise for v2
    A3 = A(v3);          %transformasjonsmatrise for v3

%POSISJON=============================================================

%q=NewtonR('jac','phi',q,t(i,1));
q = fzerom_par('phi',q,t(i,1));

%HASTIGHET=============================================================
lt1 = Cyl1_length(t(i,1),l_1_init,v_1,a_1);           %Sylinder 1 driver
lt1d = Cyl1_length_d(t(i,1),l_1_init,v_1,a_1);        %Sylinder 1 driver
lt1dd = Cyl1_length_dd(t(i,1),l_1_init,v_1,a_1);      %Sylinder 1 driver

lt2 = Cyl2_length(t(i,1),l_2_init,v_2,a_2);           %Sylinder 2 driver
lt2d = Cyl2_length_d(t(i,1),l_2_init,v_2,a_2);        %Sylinder 2 driver
lt2dd = Cyl2_length_dd(t(i,1),l_2_init,v_2,a_2);      %Sylinder 2 driver


qd = jac(q)\[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2*lt1*lt1d(1,1)
2*lt2*lt2d(1,1)]';

r1d=qd(1:3,1);        %x1,y1,z1
r2d=qd(7:9,1);        %x2,y2,z2
r3d=qd(13:15,1);      %x3,y3,z3

v1d = qd(4:6,1);      %v1
v2d = qd(10:12,1);    %v2
v3d = qd(16:18,1);    %v3

d1 = r1 + A1*s1D - A2*s2E - r2;     %vektorkoord longitudinal driver 1
d2 = r2 + A2*s2F - A3*s3G - r3;     %vektorkoord longitudinal driver 2

d1d = r1d + A1*skew(v1d)*s1D - A2*skew(v2d)*s2E - r2d;   % derivert
d2d = r2d + A2*skew(v2d)*s2F - A3*skew(v3d)*s3G - r3d;   % derivert

%AKSELERASJON==========================================================

rh = [-A1*skew(v1d)^2*s1A;
     -u_x'*A1*skew(v1d)^2*u_z;
     -u_y'*A1*skew(v1d)^2*u_z;
     -A1*skew(v1d)^2*s1B + A2*skew(v2d)^2*s2B;
     (-A1*skew(v1d)^2*u_x)'*(A2*u_z)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_z)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_z)...
     + (-A1*u_x)'*(A2*skew(v2d)^2*u_z);
     (-A1*skew(v1d)^2*u_x)'*(A2*u_y)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_y)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_y)...
     + (-A1*u_x)'*(A2*skew(v2d)^2*u_y);
     -A2*skew(v2d)^2*s2C + A3*skew(v3d)^2*s3C;
     (-A2*skew(v2d)^2*u_x)'*(A3*u_z)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_z)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_z)...
     + (-A2*u_x)'*(A3*skew(v3d)^2*u_z);
     (-A2*skew(v2d)^2*u_x)'*(A3*u_y)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_y)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_y)...
```

```
     + (-A2*u_x)'*(A3*skew(v3d)^2*u_y);
      0;
     (-2*(d1d'*d1d) + 2*lt1d(1,1)*lt1d(1,1) + 2*lt1*lt1dd...
     -2*d1'*( A1*skew(v1d)*skew(v1d)*s1D - A2*skew(v2d)*skew(v2d)*s2E));
     (-2*(d2d'*d2d) + 2*lt2d(1,1)*lt2d(1,1) + 2*lt2*lt2dd...
     -2*d2'*( A2*skew(v2d)*skew(v2d)*s2F - A3*skew(v3d)*skew(v3d)*s3G))];

qdd = jac(q)\rh;

%Reaksjonskrefter ====================================================

M(1:18,1:18)    =0;
M(1:3,1:3)      =M1;
M(4:6,4:6)      =J1;
M(7:9,7:9)      =M2;
M(10:12,10:12)  =J2;
M(13:15,13:15)  =M3;
M(16:18,16:18)  =J3;

F_last = [0;0;1000*g(3,1)];      %last
M_last = cross(s3H,A3'*F_last);  %Momentet lasten gir om tyngdepunktet)

F_masse3 = g(3,1)*m3;    %Gravitasjonskrefter masse 3
F_masse2 = g(3,1)*m2;    %Gravitasjonskrefter masse 2
F_masse1 = g(3,1)*m1;    %Gravitasjonskrefter masse 1

g_ext = [0 0 F_masse1 0 0 0 0 0 F_masse2 0 0 0 0 0 (F_masse3+F_last(3,1))
M_last']' ;%-- Eksterne krefter på systemet

lambda = (jac(q)')^-1 * (M*qdd - g_ext) ;    %--Lagranges multiplikator

Jac = jac(q)';          %JACOBIAN transponert


%Krefter joint A ======================================================
FA_x = Jac(1,1)*lambda(1,1);                  %Krefter joint A x-retning
FA_y = Jac(2,2)*lambda(2,1);                  %Krefter joint A y-retning
FA_z = Jac(3,3)*lambda(3,1);                  %Krefter joint A z-retning

%Krefter joint B ======================================================
FB_x = Jac(1,6)*lambda(6,1);                  %Krefter joint B x-retning
FB_y = Jac(2,7)*lambda(7,1);                  %Krefter joint B y-retning
FB_z = Jac(3,8)*lambda(8,1);                  %Krefter joint B z-retning

FB_negativ_x = Jac(7,6)*lambda(6,1);    %Krefter joint B x-retning negativ
FB_negativ_y = Jac(8,7)*lambda(7,1);    %Krefter joint B y-retning negativ
FB_negativ_z = Jac(9,8)*lambda(8,1);    %Krefter joint B z-retning negativ

%Krefter joint C ======================================================
FC_x = Jac(7,11)*lambda(11,1);                %Krefter joint B x-retning
FC_y = Jac(8,12)*lambda(12,1);                %Krefter joint B y-retning
FC_z = Jac(9,13)*lambda(13,1);                %Krefter joint B z-retning

FC_negativ_x = Jac(13,11)*lambda(11,1);     %Krefter joint B x-retning
FC_negativ_y = Jac(14,12)*lambda(12,1);     %Krefter joint B y-retning
FC_negativ_z = Jac(15,13)*lambda(13,1);     %Krefter joint B z-retning

%Sylinderkrefter======================================================
```

```
F_syl1_xyz_dynamisk = Jac(1:3,17)*lambda(17,1);      %Krefter fra sylinder 1
på legem 1
F_syl1_xyz_negativ = Jac(7:9,17)*lambda(17,1);      %Krefter fra sylinder 1
på legem 2

F_syl1 = norm(F_syl1_xyz_dynamisk);                 %Kraft fra sylinder 1
på legem 1

F_syl2_xyz_dynamisk = Jac(7:9,18)*lambda(18,1);             %Krefter fra
sylinder 2 på legem 2
F_syl2_xyz_dynamisk_negativ = Jac(13:15,18)*lambda(18,1);   %Krefter fra
sylinder 2 på legem 3

F_syl2 = norm(F_syl2_xyz_dynamisk);                 %Kraft fra sylinder 1
på legem 1


%Statiske beregninger
2D=================================================================

r_AB = -A1*s1A + A1*s1B;         %Distanse joint A til B
r_AD = -A1*s1A + A1*s1D;         %Distanse joint A til D
r_AI = -A1*s1A;                  %Distanse joint A til tyngdepunktet

r_BC = -A2*s2B + A2*s2C;         %Distanse joint B til C
r_BF = -A2*s2B + A2*s2F;         %Distanse joint B til F
r_BJ = -A2*s2B;                  %Distanse joint B til tyngdepunktet
r_BE = -A2*s2B + A2*s2E;         %Distanse joint B til E

r_CH = -A3*s3C + A3*s3H;         %Distanse joint C til H
r_CK = -A3*s3C;                  %Distanse joint C til tyngdepunktet
r_CG = -A3*s3C + A3*s3G;         %Distanse joint C til G


u1 = -(1/norm(d1))*d1;           %Egenvektor vektor d1 (sylinder 1)
u2 = -(1/norm(d2))*d2;           %Egenvektor vektor d2 (sylinder 2)

F_syl2_statisk = r_CH(2,1)*F_last(3,1)...
    + r_CK(2,1)*F_masse3 / (u2(3,1)*r_CG(2,1) + u2(2,1)*-r_CG(3,1));

FC_y_statisk = -F_syl2_statisk*u2(2,1);
FC_z_statisk = -F_syl2_statisk*u2(3,1) + F_masse3 + F_last(3,1);

F_syl1_statisk = ((r_BC(2,1)*FC_z_statisk) - (r_BC(3,1)*FC_y_statisk)...
    + (r_BF(2,1)*F_syl2_statisk*u2(3,1))...
    - (r_BF(3,1)*F_syl2_statisk*u2(2,1))...
    + (r_BJ(2,1)*F_masse2)) / (u1(2,1)*-r_BE(3,1) + u1(3,1)*r_BE(2,1));

FB_y_statisk = FC_y_statisk + F_syl2_statisk*u2(2,1)...
    - F_syl1_statisk*u1(2,1);
FB_z_statisk = FC_z_statisk + F_syl2_statisk*u2(3,1)...
    + F_masse2 - F_syl1_statisk*u1(3,1);

FA_y_statisk = FB_y_statisk + F_syl1_statisk*u1(2,1);
FA_z_statisk = FB_z_statisk + F_syl1_statisk*u1(3,1) + F_masse1;
```

```
%=========================================================================
Y(i,1:18)=q';
Yd(i,1:18)=qd';
Ydd(i,1:18)=qdd';
%
YA(i,1:3) = [FA_x,FA_y,FA_z];
YB(i,1:3) = [FB_x,FB_y,FB_z];
YC(i,1:3) = [FC_x,FC_y,FC_z];


Ysyl1(i,1:2)=[F_syl1,F_syl1_statisk];
Ysyl2(i,1:2)=[F_syl2,F_syl2_statisk];

Y_syl1(i,1)=F_syl1;
Y_syl2(i,1)=F_syl2;

Y_d1(i,1)=norm(d1);
Y_d2(i,1)=norm(d2);
end

%=========================================================================
% Numerisk kontroll

Ydnum(1:N,1:18) = 0;
Yddnum(1:N,1:18) = 0;
for i=1:N-1
    Ydnum(i,1:18) = 1/dt*(Y(i+1,1:18)-Y(i,1:18));
    Yddnum(i,1:18) = 1/dt*(Yd(i+1,1:18)-Yd(i,1:18));
end
Ydnum(N,1:18) = Ydnum(N-1,1:18);
Yddnum(N,1:18) = Yddnum(N-1,1:18);

% %PLOT====================================================================

figure(1)
subplot(2,3,1)
hold on
    plot(t,Y(:,1),'r')
    plot(t,Y(:,2),'b')
    plot(t,Y(:,3),'y')
hold off
legend('x_1','y_1','z_1')
title('krantårn')
xlabel('tid, [t]')
ylabel('posisjon, [m]')
grid on
box on

subplot(2,3,4)
hold on
    plot(t,Y(:,4),'r')
    plot(t,Y(:,5),'b')
    plot(t,Y(:,6),'y')
hold off
legend('\phi_{1x}','\phi_{1y}','\phi_{1z}')
title('Krantårn')
xlabel('tid, [t]')
ylabel('vinkelposisjon, [rad]')
grid on
box on
```

```
subplot(2,3,2)
hold on
    plot(t,Y(:,7),'r')
    plot(t,Y(:,8),'b')
    plot(t,Y(:,9),'y')
hold off
legend('x_2','y_2','z_2')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('posisjon, [m]')
grid on
box on

subplot(2,3,5)
hold on
    plot(t,Y(:,10),'r')
    plot(t,Y(:,11),'b')
    plot(t,Y(:,12),'y')
hold off
legend('\phi_{2x}','\phi_{2y}','\phi_{2z}')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('vinkelposisjon, [rad]')
grid on
box on


subplot(2,3,3)
hold on
    plot(t,Y(:,13),'r')
    plot(t,Y(:,14),'b')
    plot(t,Y(:,15),'y')
hold off
legend('x_3','y_3','z_3')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('posisjon, [m]')
grid on
box on

subplot(2,3,6)
hold on
    plot(t,Y(:,16),'r')
    plot(t,Y(:,17),'b')
    plot(t,Y(:,18),'y')
hold off
legend('\phi_{3x}','\phi_{3y}','\phi_{3z}')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('vinkelposisjon, [rad]')
grid on
box on

%Hastigheter_____________________________________________________

figure(2)

subplot(2,3,1)
hold on
```

```
    plot(t,Yd(:,1),'r')
    plot(t,Yd(:,2),'b')
    plot(t,Yd(:,3),'y')
    plot(t,Ydnum(:,1),'r*')
    plot(t,Ydnum(:,2),'b*')
    plot(t,Ydnum(:,3),'y*')
hold off
legend('xd_1','yd_1','zd_1')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Hastighet, [m/s]')
grid on
box on

subplot(2,3,4)
hold on
    plot(t,Yd(:,4),'r')
    plot(t,Yd(:,5),'b')
    plot(t,Yd(:,6),'y')

hold off
legend('\phid_{1x}','\phid_{1y}','\phid_{1z}')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Vinkelhastighet, [rad/s]')
grid on
box on

subplot(2,3,2)
hold on
    plot(t,Yd(:,7),'r')
    plot(t,Yd(:,8),'b')
    plot(t,Yd(:,9),'y')

    plot(t,Ydnum(:,7),'r*')
    plot(t,Ydnum(:,8),'b*')
    plot(t,Ydnum(:,9),'y*')
hold off
legend('xd_2','yd_2','zd_2')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Hastighet, [m/s]')
grid on
box on

subplot(2,3,5)
hold on
    plot(t,Yd(:,10),'r')
    plot(t,Yd(:,11),'b')
    plot(t,Yd(:,12),'y')

hold off
legend('\phid_{2x}','\phid_{2y}','\phid_{2z}')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Vinkelhastighet, [rad/s]')
grid on
box on
```

```
subplot(2,3,3)
hold on
    plot(t,Yd(:,13),'r')
    plot(t,Yd(:,14),'b')
    plot(t,Yd(:,15),'y')

    plot(t,Ydnum(:,13),'r*')
    plot(t,Ydnum(:,14),'b*')
    plot(t,Ydnum(:,15),'y*')
hold off
legend('xd_3','yd_3','zd_3')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Hastighet, [m/s]')
grid on
box on

subplot(2,3,6)
hold on
    plot(t,Yd(:,16),'r')
    plot(t,Yd(:,17),'b')
    plot(t,Yd(:,18),'y')


hold off
legend('\phid_{3x}','\phid_{3y}','\phid_{3z}')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Vinkelhastighet, [rad/s]')
grid on
box on


%Akselerasjoner_____________________________________________

figure(3)

subplot(2,3,1)
hold on
    plot(t,Ydd(:,1),'r')
    plot(t,Ydd(:,2),'b')
    plot(t,Ydd(:,3),'y')

    plot(t,Yddnum(:,1),'r*')
    plot(t,Yddnum(:,2),'b*')
    plot(t,Yddnum(:,3),'y*')
hold off
legend('xdd_1','ydd_1','zdd_1')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Akselerasjon, [m^2/s]')
grid on
box on

subplot(2,3,4)
hold on
    plot(t,Ydd(:,4),'r')
    plot(t,Ydd(:,5),'b')
    plot(t,Ydd(:,6),'y')
```

```
hold off
legend('\phidd_{1x}','\phidd_{1y}','\phidd_{1z}')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Vinkelakselerasjon, [rad^2/s]')
grid on
box on

subplot(2,3,2)
hold on
    plot(t,Ydd(:,7),'r')
    plot(t,Ydd(:,8),'b')
    plot(t,Ydd(:,9),'y')

    plot(t,Yddnum(:,7),'r*')
    plot(t,Yddnum(:,8),'b*')
    plot(t,Yddnum(:,9),'y*')
hold off
legend('xdd_2','ydd_2','zdd_2')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Akselerasjon, [m^2/s]')
grid on
box on

subplot(2,3,5)
hold on
    plot(t,Ydd(:,10),'r')
    plot(t,Ydd(:,11),'b')
    plot(t,Ydd(:,12),'y')

hold off
legend('\phidd_{2x}','\phidd_{2y}','\phidd_{2z}')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Vinkelakselerasjon, [rad^2/s]')
grid on
box on


subplot(2,3,3)
hold on
    plot(t,Ydd(:,13),'r')
    plot(t,Ydd(:,14),'b')
    plot(t,Ydd(:,15),'y')

    plot(t,Yddnum(:,13),'r*')
    plot(t,Yddnum(:,14),'b*')
    plot(t,Yddnum(:,15),'y*')
hold off
legend('xdd_3','ydd_3','zdd_3')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Akselerasjon, [m^2/s]')
grid on
box on

subplot(2,3,6)
hold on
```

```
    plot(t,Ydd(:,16),'r')
    plot(t,Ydd(:,17),'b')
    plot(t,Ydd(:,18),'y')


hold off
legend('\phidd_{3x}','\phidd_{3y}','\phidd_{3z}')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Vinkelakselerasjon, [rad^2/s]')
grid on
box on

% % Reaksjonskrefter=====================================================

figure(4)

subplot(3,1,1)
hold on
plot(t,YA(:,1),'r')
plot(t,YA(:,2),'b')
plot(t,YA(:,3),'y')


hold off
xlabel('tid, [t]')
ylabel('Krefter')
legend('F_x','F_y','F_z','F_{stat x}','F_{stat y}','F_{stat z}')
title('Reaksjonskrefter binding A')
grid on
box on


subplot(3,1,2)
hold on
plot(t,YB(:,1),'r')
plot(t,YB(:,2),'b')
plot(t,YB(:,3),'y')

hold off
xlabel('tid, [t]')
ylabel('Krefter')
legend('F_x','F_y','F_z','F_{stat x}','F_{stat y}','F_{stat z}')
title('Reaksjonskrefter binding B')
grid on
box on

subplot(3,1,3)
hold on
plot(t,YC(:,1),'r')
plot(t,YC(:,2),'b')
plot(t,YC(:,3),'y')

hold off
xlabel('tid, [t]')
ylabel('Krefter')
legend('F_x','F_y','F_z','F_{stat x}','F_{stat y}','F_{stat z}')
title('Reaksjonskrefter binding C')
grid on
```

```
box on

for i=1:N

%3D animasjon=================================================================
    cla


r1_plot=Y(i,1:3)';    %x1,y1,z1
r2_plot=Y(i,7:9)';    %x2,y2,z2
r3_plot=Y(i,13:15)';  %x3,y3,z3

v1_plot = Y(i,4:6)';  %w1 (inneholdene phi1, theta1 og tau1)
v2_plot = Y(i,10:12)';%w2 (inneholdene phi2, theta2 og tau2)
v3_plot = Y(i,16:18)';%w2 (inneholdene phi3, theta3 og tau3)

A1 = A(v1_plot); %transformasjonsmatrise for v1
A2 = A(v2_plot); %transformasjonsmatrise for v2
A3 = A(v3_plot); %transformasjonsmatrise for v3

%krantårn
krant=[[b1/2 -D1/2 -h/2]',[b1/2 D1/2 -h/2]',[-b1/2 D1/2 -h/2]',...
    [-b1/2 -D1/2 -h/2]',[b1/2 -D1/2 -h/2]',...
[b1/2 -D1/2 h/2]',[b1/2 D1/2 h/2]',[b1/2 D1/2 -h/2]',[b1/2 D1/2 h/2]',...
[-b1/2 D1/2 h/2]',[-b1/2 D1/2 -h/2]',[-b1/2 D1/2 h/2]',...
[-b1/2 -D1/2 h/2]',[-b1/2 -D1/2 -h/2]',[-b1/2 -D1/2 h/2]',...
[b1/2 -D1/2 h/2]'];

%Kranarm1
kranarm1=[[b1/2 s2B(2,1) s2B(3,1)-D1/2]',[b1/2 s2C(2,1) s2C(3,1)-D1/2]',...
    [-b1/2 s2C(2,1) s2C(3,1)-D1/2]',[-b1/2 s2B(2,1) s2B(3,1)-D1/2]',...
    [b1/2 s2B(2,1) s2B(3,1)-D1/2]',...
[b1/2 s2B(2,1) s2B(3,1)+D1/2]',[b1/2 s2C(2,1) s2C(3,1)+D1/2]',...
[b1/2 s2C(2,1) s2C(3,1)-D1/2]',[b1/2 s2C(2,1) s2C(3,1)+D1/2]',...
[-b1/2 s2C(2,1) s2C(3,1)+D1/2]',[-b1/2 s2C(2,1) s2C(3,1)-D1/2]',...
[-b1/2 s2C(2,1) s2C(3,1)+D1/2]',[-b1/2 s2B(2,1) s2B(3,1)+D1/2]',...
[-b1/2 s2B(2,1) s2B(3,1)-D1/2]',[-b1/2 s2B(2,1) s2B(3,1)+D1/2]',...
[b1/2 s2B(2,1) s2B(3,1)+D1/2]'];

kranarm2=[[b1/2 s3C(2,1) (s3C(3,1)-D1/2)]',...
    [b1/2 s3G(2,1) (s3C(3,1)-D1/2)]',...
    [b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [-b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [-b1/2 s3G(2,1) (s3C(3,1)-D1/2)]',[-b1/2 s3C(2,1) (s3C(3,1)-D1/2)]',...
    [-b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',...
    [-b1/2 s3G(2,1)+D1/2 (s3C(3,1)+D1/2)]',...
    [-b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [-b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [-b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [b1/2 s3G(2,1)+D1/2 (s3C(3,1)+D1/2)]',[b1/2 s3C(2,1)
(s3C(3,1)+D1/2)]',...
    [-b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',[b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',...
    [b1/2 s3C(2,1) (s3C(3,1)-D1/2)]',[-b1/2 s3C(2,1) (s3C(3,1)-D1/2)]'];

plot_krant = A1*krant;
plot_kranarm1 = A2*kranarm1;
```

```
plot_kranarm2 = A3*kranarm2;

plot_krant(1,:)=r1_plot(1,1) + plot_krant(1,:);
plot_krant(2,:)=r1_plot(2,1) + plot_krant(2,:);
plot_krant(3,:)=r1_plot(3,1) + plot_krant(3,:);

plot_kranarm1(1,:)=r2_plot(1,1) + plot_kranarm1(1,:);
plot_kranarm1(2,:)=r2_plot(2,1) + plot_kranarm1(2,:);
plot_kranarm1(3,:)=r2_plot(3,1) + plot_kranarm1(3,:);

plot_kranarm2(1,:)=r3_plot(1,1) + plot_kranarm2(1,:);
plot_kranarm2(2,:)=r3_plot(2,1) + plot_kranarm2(2,:);
plot_kranarm2(3,:)=r3_plot(3,1) + plot_kranarm2(3,:);

figure(8)

hold on
plot3(plot_krant(1,:),plot_krant(2,:),plot_krant(3,:),'b','linewidth',3)
plot3(plot_kranarm1(1,:),plot_kranarm1(2,:),plot_kranarm1(3,:),'r','linewid
th',3)
plot3(plot_kranarm2(1,:),plot_kranarm2(2,:),plot_kranarm2(3,:),'g','linewid
th',3)

view(3)
grid on
axis([-50 50 -50 50 -50 50])
hold off

if i==0.02
    pause

end
    pause(0.001)
end


%Konstanter=====================================================================
function a = konstanter

%Simuleringstid=================================================================
t_min=0;                  %start tid
dt=0.02;                  %tidssteg
t_max=20;                  %slutt tid

%Funksjon (retningsventil)======================================================
t_start = 0;          %startverdi for funksjonen
t_open = 0.2;              %setter åpne og lukketid for ventilen
t_intervall_1 = t_max-(2*t_open);      %setter intervallet for funksjonen


Q_max1 = 0.015;          %900 [l/min]
Q_max2 = 0.015;          %900 [l/min]
Q_max3 = 0.015;          %900 [l/min]
%Kranposisjon A og B============================================================
rH_A = [0 15 5];                %pos A krantupp
rH_B = [0 15 5];                %pos B krantupp

%Sylinder areal=================================================================
```

```
rad1 = 0.32/2;                          %Radius stempel sylinder 1, [m]
rad2 = 0.23/2;                         %Radius stempelstand sylinder 1, [m]
rad3 = 0.30/2;                          %Radius stempel sylinder 2, [m]
rad4 = 0.20/2;                         %Radius stempelstand sylinder 2, [m]

syl1_areal_pos = pi*rad1^2;
syl1_areal_neg = pi*rad1^2-pi*rad2^2;
syl2_areal_pos = pi*rad3^2;
syl2_areal_neg = pi*rad3^2-pi*rad4^2;

%Fortrengningsvolum rotasjon=============================================
displacement = 0.05;          %m^3/omdreining

%Lengder kran============================================================
h =3.85;                     %Høyde krantårn (z-retning)
b1=1;                        %bredde (3D anim)
D1=2;                        %dybde (3D anim)

s1A = [0;0;-h/2];            %lengde fra lokal koordinat 1 til joint A
s1B = [0;0;h/2];             %lengde fra lokal koordinat 1 til joint B
s1D = [0;1.75;-1.475];       %lengde fra lokal koordinat 1 til joint D
s2B = [0;-10.666;-0.37];     %lengde fra lokal koordinat 2 til joint B
s2C = [0;12.834;-0.37];      %lengde fra lokal koordinat 2 til joint C
s2E = [0;-4.666;-1];         %lengde fra lokal koordinat 2 til joint E
s2F = [0;6.484;-1];          %lengde fra lokal koordinat 2 til joint F
s3C = [0;-7.054;2.805];      %lengde fra lokal koordinat 3 til joint C
s3G = [0;-4.804;2.243];     %lengde fra lokal koordinat 3 til joint G
s3H = [0;9.107;-6.326];      %lengde fra lokal koordinat 3 til krantupp


%initial lengder Sylinder 1 og 2=========================================

syl1_lengde = (s1B-s2B+s2E) - (s1D);
l_1_init = norm(syl1_lengde);

syl2_lengde = (s2C-s3C+s3G) - (s2F);
l_2_init = norm(syl2_lengde);

%Hastighet og akselerasjon hydrauliske sylindre==========================
v_1 = 0.2;                          %hastighet sylinder 1
v_2 = -0.2;                         %hastighet sylinder 2
a_1 = 0;                            %akselerasjon sylinder 1
a_2 = 0;                            %akselerasjon sylinder 2

%Masse krantårn og kranarm 1 og 2========================================
m1= 2000;   %masse krantårn
m2= 1500;   %masse kranarm1
m3= 1000;   %masse kranarm2

M1= eye(3)*m1;
M2= eye(3)*m2;
M3= eye(3)*m3;
%Treghetsmoment krantårn og kranarm 1 og 2============================0
% Inertia tensors as given (unit is T*mm^2)
j1 = [8.0028345e7 -9.737385e6 1.9917595e7;...
      -9.737385e6 8.2777918e7 2.1722351e5;...
      1.9917595e7 2.1722351e5 9.8183294e7];

j2 = [8.5127790e8 5.6483637e5 1.3313718e6;...
```

```
      5.6483637e5 8.4813949e8 -9.3820534e6;...
      1.3313718e6 -9.3820534e6 1.3943732e7];

j3 = [3.9622348e8 -1.3399605e6 -1.5215701e6;...
      -1.3399605e6 2.7403523e8 -1.7760645e8;...
      -1.5215701e6 -1.7760645e8 1.2470308e8];
%=========================================================================
%transformation into model coordinate system (unit is kg*m^2)
A_units = [0 0 -1; -1 0 0; 0 1 0];

J1 = A_units*j1*A_units'*1e-3;
J2 = A_units*j2*A_units'*1e-3;
J3 = A_units*j3*A_units'*1e-3;


%=========================================================================

a(1:3,1:45) = 0;
a(1:3,1:10) = [s1A, s1B, s1D,s2B,s2C,s2E,s2F,s3C,s3G,s3H];%lengde koordinat
a(1:3,13)    = [h;b1;D1];     %------------ høgde bredde og dybde krantårn
% a(1:3,14)    = [L1;L2;0];      %------------------- Lengde kranarm 1 og 2
a(1:3,15)    = [0;v_1;v_2];    %----------------- hastighet sylinder 1 og 2
a(1:3,16)    = [0;a_1;a_2];    %-------------- akselerasjon sylinder 1 og 2
a(1:3,17)    = [m1;m2;m3];  %--------------- masse krantårn, kranarm 1 og 2
a(1:3,18:20) = J1;   %---------------------------- treghetsmoment krantårn
a(1:3,21:23) = J2;   %---------------------------- treghetsmoment kranarm 1
a(1:3,24:26) = J3;   %---------------------------- treghetsmoment kranarm 2
a(1:3,27:29) = M1;
a(1:3,30:32) = M2;
a(1:3,33:35) = M3;
a(1:3,36)    = [0;0;9.81];  %---------------------- gravitasjon [9,81m/s^2]
a(1:3,37)    = rH_A;      %-----------------------------------startposisjon
a(1:3,38)    = rH_B;        %----------------------------------sluttposisjon
a(1,39)      = syl1_areal_pos;
a(2,39)      = syl1_areal_neg;
a(1,40)      = syl2_areal_pos;
a(2,40)      = syl2_areal_neg;
a(3,40)      = displacement;
a(1,41)      = t_start;
a(2,41)      = t_intervall_1;
a(3,41)      = t_open;
a(1,42)      = t_min;
a(2,42)      = dt;
a(3,42)      = t_max;
a(1,43)      = l_1_init;
a(2,43)      = l_2_init;
a(1:3,44)    = [Q_max1; Q_max2; Q_max3];



%Newton Raphson ================================
function NewR=NewtonR(jac,phi,q,t)

tol=1e-10;
error=1;
n_max=150;
i=0;

while error>tol || i<n_max
```

```
    i=i+1;
J=feval(jac,q);
P=feval(phi,q,t);
delta_q = J\P;
q= q - delta_q;
error=norm(delta_q);

end
NewR=q;



%Phi=================================================================
function fi=phi(q,t)

%Henter konstanter===================================================
k=konstanter;

s1A = k(1:3,1);      %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);      %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);      %lengde fra lokal koordinat 1 til joint D
s2B = k(1:3,4);      %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5);      %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6);      %lengde fra lokal koordinat 2 til joint E
s2F = k(1:3,7);      %lengde fra lokal koordinat 2 til joint F
s3C = k(1:3,8);      %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9);      %lengde fra lokal koordinat 3 til joint G


l1 = k(1,43);        %initial lengde sylinder 1
l2 = k(2,43);        %initial lengde sylinder 2

v_1 = k(2,15);       %hastighet sylinder 1
v_2 = k(3,15);       %hastighet sylinder 2

a_1 = k(2,16);       %akselerasjon sylinder 1
a_2 = k(3,16);       %akselerasjon sylinder 2
%====================================================================

r1=q(1:3,1);     %x1,y1,z1
r2=q(7:9,1);     %x2,y2,z2
r3=q(13:15,1);   %x3,y3,z3

v1 = q(4:6,1);   %w1 (inneholdene phi1, theta1 og tau1)
v2 = q(10:12,1);%w2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);%w3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1); %transformasjonsmatrise for v1
A2 = A(v2); %transformasjonsmatrise for v2
A3 = A(v3); %transformasjonsmatrise for v3

u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';

d1 = r1 + A1*s1D - A2*s2E - r2;   %vektorkoordinat longitudinal driver 1
```

```
d2 = r2 + A2*s2F - A3*s3G - r3;   %ve longitudinal driver 2

lt1 = Cyl1_length(t,l1,v_1,a_1);  %Sylinder 1 driver (funksjon)
lt2 = Cyl2_length(t,l2,v_2,a_2);  %Sylinder 2 driver (funksjon)

fi(1:18,1) = 0;
fi(1:3,1) = r1 + A1*s1A;
fi(4,1) = u_x'*A1*u_z;
fi(5,1) = u_y'*A1*u_z;
fi(6:8,1) = r1 + A1*s1B - A2*s2B - r2;
fi(9,1) = (A1*u_x)'*(A2*u_z);
fi(10,1) = (A1*u_x)'*(A2*u_y);
fi(11:13,1) = r2 + A2*s2C - r3 - A3*s3C;
fi(14,1) = (A2*u_x)'*(A3*u_z);
fi(15,1) = (A2*u_x)'*(A3*u_y);
fi(16,1) = v1(3,1) - 1*t;
fi(17,1) = (d1'*d1) - lt1^2;
fi(18,1) = (d2'*d2) - lt2^2;


if (norm(d1)<0.4)
    hjelp = 1;
    pause
end

if (norm(d2)<0.4)
    hjelp = 2;
    pause
end




%Jacobian ================================================================
function jacobi=jac(q)

%Henter konstanter=========================================================

k=konstanter;

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s2B = k(1:3,4);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6);     %lengde fra lokal koordinat 2 til joint E
s2F = k(1:3,7);     %lengde fra lokal koordinat 2 til joint F
s3C = k(1:3,8);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9);     %lengde fra lokal koordinat 3 til joint G

%==========================================================================
r1=q(1:3,1);    %x1,y1,z1
r2=q(7:9,1);    %x2,y2,z2
r3=q(13:15,1);  %x3,y3,z3


v1 = q(4:6,1);  %w1 (inneholdene phi1, theta1 og tau1)
```

```
v2 = q(10:12,1);%w2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);%w3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1); %transformasjonsmatrise for v1
A2 = A(v2); %transformasjonsmatrise for v2
A3 = A(v3); %transformasjonsmatrise for v3

u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';


d1 = r1 + A1*s1D - A2*s2E - r2;   %lengde longitudinal driver 1
d2 = r2 + A2*s2F - A3*s3G - r3;   %lengde longitudinal driver 2

skew_s1A = skew(s1A);
skew_s1B = skew(s1B);
skew_s1D = skew(s1D);

skew_s2B = skew(s2B);
skew_s2C = skew(s2C);
skew_s2E = skew(s2E);
skew_s2F = skew(s2F);

skew_s3C = skew(s3C);
skew_s3G = skew(s3G);


skew_u_x = skew(u_x);
skew_u_y = skew(u_y);
skew_u_z = skew(u_z);


jacobi(1:18,1:18)=0;
jacobi(1:3,1:6)=    [eye(3), -A1*skew_s1A];
jacobi(4,4:6) =     -u_x'*A1*skew_u_z;
jacobi(5,4:6) =     -u_y'*A1*skew_u_z;
jacobi(6:8,1:12) =  [eye(3), -A1*skew_s1B, -eye(3), A2*skew_s2B];
jacobi(9,4:12) =    [(A2*u_z)'*(-A1*skew_u_x), zeros(1,3),...
    (A1*u_x)'*(-A2*skew_u_z)];
jacobi(10,4:12) =   [(A2*u_y)'*(-A1*skew_u_x), zeros(1,3),...
    (A1*u_x)'*(-A2*skew_u_y)];
jacobi(11:13,7:18) =[eye(3), -A2*skew_s2C, -eye(3), A3*skew_s3C];
jacobi(14,10:18) =  [(A3*u_z)'*(-A2*skew_u_x), zeros(1,3),...
    (A2*u_x)'*(-A3*skew_u_z)];
jacobi(15,10:18) =  [(A3*u_y)'*(-A2*skew_u_x), zeros(1,3),...
    (A2*u_x)'*(-A3*skew_u_y)];
jacobi(16,6) =      1;
jacobi(17,1:12) =   [2*d1', -2*d1'*A1*skew_s1D, -2*d1', 2*d1'*A2*skew_s2E];
jacobi(18,7:18) =   [2*d2', -2*d2'*A2*skew_s2F, -2*d2', 2*d2'*A3*skew_s3G];



%Transformasjonsmatrise=================================================0
function A_matrix=A(v)

phi = v(1,1);
theta = v(2,1);
```

```
tau = v(3,1);

A_x=[1 0 0;0 cos(phi) -sin(phi);0 sin(phi) cos(phi)]; %rotasjon om x aksen
A_y=[cos(theta) 0 sin(theta);0 1 0;-sin(theta) 0 cos(theta)];%rotasjon om y
aksen
A_z=[cos(tau) -sin(tau) 0;sin(tau) cos(tau) 0;0 0 1];    %rotasjon om z
aksen
A_matrix = A_z*A_y*A_x;



%Skew matrise=====================================
function s = skew(a)
s = [0, -a(3,1), a(2,1);...
    a(3,1), 0, -a(1,1);...
    -a(2,1), a(1,1), 0];




%==========================================
function l = Cyl1_length(t,l_ini,v,a)
l = l_ini + v*t + 1/2*a*t^2;


function l = Cyl1_length_d(t,l_ini,v,a)
l = v + a*t;


function l = Cyl1_length_dd(t,l_ini,v,a)
l = a;




%==========================================
function l = Cyl2_length(t,l_ini,v,a)
l = l_ini + v*t + 1/2*a*t^2;


function l = Cyl2_length_d(t,l_ini,v,a)
l = v + a*t;

function l = Cyl2_length_dd(t,l_ini,v,a)
l = a;
```

Vedlegg 2: Optimering av krankonstruksjon, Matlab

```
%Optimering av krankonstruksjon==============================

x0 = [1,1,1,1,1,1,1,1,1,1]';    %Initialverdier


f = @(x)OptimQ_max(x);          %nonlineær begrensning
ff = @(x)main_script(x);        %objektsfunksjon

lb = [0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8];
ub = [1.2,1.2,1.2,1.2,1.2,1.2,1.2,1.2,1.2,1.2];


options=optimset('Algorithm','interior-point');

[xopt,val,exitflag,output] = fmincon(ff,x0,[],[],[],[],lb,ub,f);

anim = resultat(xopt);



%constraints=========================================================

function [c,ceq]=OptimQ_max(x)

global l_SYLINDER1A l_SYLINDER2A l_SYLINDER1B l_SYLINDER2B...
    VINKEL2_A VINKEL3_A VINKEL2_B VINKEL3_B

c = [l_SYLINDER1A-9.05;l_SYLINDER2A-8.24;-l_SYLINDER1B+5.075;...
    -l_SYLINDER2B+4.67; -l_SYLINDER1A+5.075;-l_SYLINDER2A+4.67;...
    l_SYLINDER1B-9.05;l_SYLINDER2B-8.24;-VINKEL2_A; VINKEL2_A-(pi/2);...
    -VINKEL2_B; VINKEL2_B-(pi/2); -VINKEL3_A; VINKEL3_A-pi;...
    -VINKEL3_B; VINKEL3_B-pi];

ceq = [];
end



%Objeksfunksjon======================================================

function e=main_script(x)
%Henter konstanter===================================================

k=konstanter;

h = k(1,13);       %Høyde krantårn (z-retning)
b1 = k(2,13);        %bredde (y-retning)
D1 = k(3,13);        %dybde (x-retning)

s1A = k(1:3,1);      %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);      %lengde fra lokal koordinat 1 til joint B
```

```
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H

m1 = k(1,17);       %masse krantårn
m2 = x(1)*k(2,17);       %masse kranarm1
m3 = x(2)*k(3,17);       %masse kranarm2

M1 = k(1:3,27:29);
M2 = k(1:3,30:32);
M3 = k(1:3,33:35);

J1 = k(1:3,18:20);
J2 = x(1)^3*k(1:3,21:23);
J3 = x(2)^3*k(1:3,24:26);

g = k(1:3,36);        %gravitasjonskraft
last = k(1,45);

%=========================================================================
u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';

% Areal sylinderstempel positiv og negativ retning + rotdriver========0
syl1_areal_pos = k(1,39);
syl1_areal_neg = k(2,39);
syl2_areal_pos = k(1,40);
syl2_areal_neg = k(2,40);
displacement = k(3,40);    %  m^3/rotasjon

%Beregner nødvendig oljevolum (forflyttning fra A til B)==============
posA = pos_A(x);

l_syl1_A = posA(1,1);
l_syl2_A = posA(2,1);
v_A = posA(3,1);

posB = pos_B(x);

l_syl1_B = posB(1,1);
l_syl2_B = posB(2,1);
v_B = posB(3,1);

sylinder1 = l_syl1_B - l_syl1_A;       %forflyttning sylinder 1
sylinder2 = l_syl2_B - l_syl2_A;       %forflyttning sylinder 2
```

```
rotasjon = v_B - v_A;                   %vinkel-rotasjon [rad]

if sylinder1 >= 0
    V1 = syl1_areal_pos * abs(sylinder1);
else
    V1 = syl1_areal_neg * abs(sylinder1);
end

if sylinder2 >= 0
    V2 = syl2_areal_pos * abs(sylinder2);
else
    V2 = syl2_areal_neg * abs(sylinder2);
end
 V3 =displacement * (abs(rotasjon)/2*pi);

%Initial verdier ========================================================
l_1_init = l_syl1_A;
l_2_init = l_syl2_A;
q = posA(4:21,1);

l_syl1 = sylinder1;
l_syl2 = sylinder2;
rot_driv = rotasjon;

% Simulering  ============================================================
t_min= k(1,42);               %start tid
dt= k(2,42);                  %tidssteg
t_max= x(1);                  %slutt tid
t=[t_min:dt:t_max]';
N=size(t,1);                  %antall iterasjoner


Y(1:N,1:18)=0;
Yd(1:N,1:18)=0;
Ydd(1:N,1:18)=0;

YA(1:N,1:6)=0;
YB(1:N,1:6)=0;
YC(1:N,1:6)=0;

Ysyl1(1:N,1:2)=0;
Ysyl2(1:N,1:2)=0;

Y_syl1(1:N,1)=0;
Y_syl2(1:N,1)=0;

Y_d1(1:N)=0;
Y_d2(1:N)=0;

Y_p1(1:N,1)=0;
Y_p2(1:N,1)=0;

Y_vinkel1(1:N,1) = 0;
Y_vinkel2(1:N,1) = 0;


Y_p1(1:N,1) = 0;
Y_p1(1:N,1) = 0;
```

```
%=========================================================

for i=1:N

q = fzerom_par('phi',q,V1,V2,V3,l_1_init,l_2_init,l_syl1,l_syl2,x,t(i,1));

    r1=q(1:3,1);        %x1,y1,z1
    r2=q(7:9,1);        %x2,y2,z2
    r3=q(13:15,1);      %x3,y3,z3

    v1 = q(4:6,1);      %v1 (inneholdene phi1, theta1 og tau1)
    v2 = q(10:12,1);    %v2 (inneholdene phi2, theta2 og tau2)
    v3 = q(16:18,1);    %v3 (inneholdene phi3, theta3 og tau3)

    A1 = A(v1);         %transformasjonsmatrise for v1
    A2 = A(v2);         %transformasjonsmatrise for v2
    A3 = A(v3);         %transformasjonsmatrise for v3


        % Drivefunksjon Sylinder 1====================================

        if l_syl1 >= 0
        lt1=(Q_syl1_pos(t(i,1),V1,x)/syl1_areal_pos) + l_1_init;
        else
        lt1=(-Q_syl1_pos(t(i,1),V1,x)/syl1_areal_neg) + l_1_init;
        end

        if l_syl1 >= 0
        lt1d=(Q_syl1_vel(t(i,1),V1,x)/syl1_areal_pos);
        else
        lt1d=(-Q_syl1_vel(t(i,1),V1,x)/syl1_areal_neg);
        end

        if l_syl1 >= 0
        lt1dd=(Q_syl1_acc(t(i,1),V1,x)/syl1_areal_pos);
        else
        lt1dd=(-Q_syl1_acc(t(i,1),V1,x)/syl1_areal_neg);
        end
        %Drivefunksjon Sylinder 2====================================
        if l_syl2 >= 0
        lt2 = (Q_syl2_pos(t(i,1),V2,x)/syl2_areal_pos) + l_2_init;
        else
        lt2 = (-Q_syl2_pos(t(i,1),V2,x)/syl2_areal_neg) + l_2_init;
        end

        if l_syl2 >=0
        lt2d = (Q_syl2_vel(t(i,1),V2,x)/syl2_areal_pos);
        else
        lt2d = (-Q_syl2_vel(t(i,1),V2,x)/syl2_areal_neg);
        end

        if l_syl2 >=0
        lt2dd = (Q_syl2_acc(t(i,1),V2,x)/syl2_areal_pos);
        else
        lt2dd = (-Q_syl2_acc(t(i,1),V2,x)/syl2_areal_neg);
        end

        %Drivefunksjon rotasjon======================================
```

```
        if rot_driv >=0
        lt3 = (Q_rot_pos(t(i,1),V3,x)/displacement);
        else
        lt3 = (-Q_rot_pos(t(i,1),V3,x)/displacement);
        end

        if rot_driv >=0
        lt3d = (Q_rot_vel(t(i,1),V3,x)/displacement);
        else
        lt3d = (-Q_rot_vel(t(i,1),V3,x)/displacement);
        end

        if rot_driv >=0
        lt3dd = (Q_rot_acc(t(i,1),V3,x)/displacement);
        else
        lt3dd = (-Q_rot_acc(t(i,1),V3,x)/displacement);
        end

%Hastighet ===========================================================
qd = jac(q,x)\[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 lt3d(1,1) 2*lt1*lt1d(1,1)
2*lt2*lt2d(1,1)]';

    r1d=qd(1:3,1);          %x1,y1,z1
    r2d=qd(7:9,1);          %x2,y2,z2
    r3d=qd(13:15,1);        %x3,y3,z3

    v1d = qd(4:6,1);        %v1 (inneholdene phi1, theta1 og tau1)
    v2d = qd(10:12,1);      %v2 (inneholdene phi2, theta2 og tau2)
    v3d = qd(16:18,1);      %v3 (inneholdene phi3, theta3 og tau3)



     d1 = r1 + A1*s1D - A2*s2E - r2; %vektorkoordinat longitudinal driver 1
     d2 = r2 + A2*s2F - A3*s3G - r3; %vektorkoordinat longitudinal driver 2

     d1d = r1d + A1*skew(v1d)*s1D - A2*skew(v2d)*s2E - r2d;    % derivert
     d2d = r2d + A2*skew(v2d)*s2F - A3*skew(v3d)*s3G - r3d;    % derivert

%Akselerasjon=========================================================
rh = [-A1*skew(v1d)^2*s1A;
     -u_x'*A1*skew(v1d)^2*u_z;
     -u_y'*A1*skew(v1d)^2*u_z;
     -A1*skew(v1d)^2*s1B + A2*skew(v2d)^2*s2B;
     (-A1*skew(v1d)^2*u_x)'*(A2*u_z)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_z)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_z)...
     + (-A1*u_x)'*(A2*skew(v2d)^2*u_z);
     (-A1*skew(v1d)^2*u_x)'*(A2*u_y)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_y)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_y)...
     + (-A1*u_x)'*(A2*skew(v2d)^2*u_y);
     -A2*skew(v2d)^2*s2C + A3*skew(v3d)^2*s3C;
     (-A2*skew(v2d)^2*u_x)'*(A3*u_z)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_z)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_z)...
     + (-A2*u_x)'*(A3*skew(v3d)^2*u_z);
     (-A2*skew(v2d)^2*u_x)'*(A3*u_y)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_y)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_y)...
```

```
     + (-A2*u_x)'*(A3*skew(v3d)^2*u_y);
      lt3dd;
     (-2*(d1d'*d1d) + 2*lt1d(1,1)*lt1d(1,1)...
     + 2*lt1*lt1dd -2*d1'*( A1*skew(v1d)*skew(v1d)*s1D...
     - A2*skew(v2d)*skew(v2d)*s2E));
     (-2*(d2d'*d2d) + 2*lt2d(1,1)*lt2d(1,1)...
     + 2*lt2*lt2dd -2*d2'*( A2*skew(v2d)*skew(v2d)*s2F...
     - A3*skew(v3d)*skew(v3d)*s3G))];


qdd = jac(q,x)\rh;

%Reaksjonskrefter
================================================================================
================================================================================
    M(1:18,1:18)    =0;
    M(1:3,1:3)      =M1;
    M(4:6,4:6)      =J1;
    M(7:9,7:9)      =M2;
    M(10:12,10:12)  =J2;
    M(13:15,13:15)  =M3;
    M(16:18,16:18)  =J3;


    F_last = [0;0;k(1,45)*g(3,1)];   %Last
    M_last = cross(s3H,A3'*F_last);  %Momentet lasten gir om tyngdepunktet)

    F_masse3 = g(3,1)*m3;    %Gravitasjonskrefter masse 3
    F_masse2 = g(3,1)*m2;    %Gravitasjonskrefter masse 2
    F_masse1 = g(3,1)*m1;    %Gravitasjonskrefter masse 1

    g_ext = [0 0 F_masse1 0 0 0 0 0 F_masse2 0 0 0 0 0
(F_masse3+F_last(3,1)) M_last']' ;%-- Eksterne krefter

    lambda = (jac(q,x)')^-1 * (M*qdd - g_ext) ;    %Lagranges multiplikator

    Jac = jac(q,x)';                                %JACOBIAN transponert


    %Krefter joint A ====================================================
    FA_x = Jac(1,1)*lambda(1,1);                  %Krefter joint A x-retning
    FA_y = Jac(2,2)*lambda(2,1);                  %Krefter joint A y-retning
    FA_z = Jac(3,3)*lambda(3,1);                  %Krefter joint A z-retning

    %Krefter joint B ====================================================
    FB_x = Jac(1,6)*lambda(6,1);                  %Krefter joint B x-retning
    FB_y = Jac(2,7)*lambda(7,1);                  %Krefter joint B y-retning
    FB_z = Jac(3,8)*lambda(8,1);                  %Krefter joint B z-retning


    %Krefter joint C ====================================================
    FC_x = Jac(7,11)*lambda(11,1);                %Krefter joint B x-retning
    FC_y = Jac(8,12)*lambda(12,1);                %Krefter joint B y-retning
    FC_z = Jac(9,13)*lambda(13,1);                %Krefter joint B z-retning


    %Sylinderkrefter=====================================================
```

```
    F_syl1_xyz_dynamisk = Jac(1:3,17)*lambda(17,1);     %Krefter fra
sylinder 1 på legem 1
    F_syl1_xyz_negativ = Jac(7:9,17)*lambda(17,1);      %Krefter fra
sylinder 1 på legem 2

    F_syl1 = norm(F_syl1_xyz_dynamisk);                 %Kraft fra sylinder
1 på legem 1

    F_syl2_xyz_dynamisk = Jac(7:9,18)*lambda(18,1);              %Krefter
fra sylinder 2 på legem 2
    F_syl2_xyz_dynamisk_negativ = Jac(13:15,18)*lambda(18,1);    %Krefter
fra sylinder 2 på legem 3

    F_syl2 = norm(F_syl2_xyz_dynamisk);                 %Kraft fra sylinder
1 på legem 1


     %Oljetrykk=============================================================

    kraftretning1 = dot(F_syl1_xyz_dynamisk,d1);
    if kraftretning1 >=0
        p1=(F_syl1*10 + 15*syl1_areal_neg)/syl1_areal_pos;
    else
        p1=(F_syl1*10 + 15*syl1_areal_pos)/syl1_areal_neg;
    end

    kraftretning2 = dot(F_syl2_xyz_dynamisk,d2);
    if kraftretning2 >= 0
        p2=(F_syl2*10 + 15*syl2_areal_neg)/syl2_areal_pos;
    else
        p2=(F_syl2*10 + 15*syl2_areal_pos)/syl2_areal_neg;
    end

%   %=================================================================


Y_syl1(i,1)=F_syl1;
Y_syl2(i,1)=F_syl2;

Y_d1(i,1)=norm(d1);
Y_d2(i,1)=norm(d2);

Y_vinkel1(i,1) = q(10,1);
Y_vinkel2(i,1) = q(17,1);

Y_p1(i,1) = p1;
Y_p2(i,1) = p2;

end

%Designkrav ============================================================
global F_SYLINDER1 F_SYLINDER2 REAKSJONSKREFTER_B REAKSJONSKREFTER_C...
    OLJETRYKK1 OLJETRYKK2

F_SYLINDER1 = max(Y_syl1(:,1));
F_SYLINDER2 = max(Y_syl2(:,1));
REAKSJONSKREFTER_B = max(max(YB(:,1:3)));
REAKSJONSKREFTER_C = max(max(YC(:,1:3)));
```

```
OLJETRYKK1 = max(Y_p1(:,1));
OLJETRYKK2 = max(Y_p1(:,1));

%Objektfunksjon============================================================
% e = -x(1) - x(2);
e = F_SYLINDER1 + F_SYLINDER2;



%=========================================================================
function a = pos_A(x)
%Henter konstanter=========================================================

k=konstanter;

h = k(1,13);      %Høyde krantårn (z-retning)
b1 = k(2,13);       %bredde (y-retning)
D1 = k(3,13);       %dybde (x-retning)

rH_A = k(1:3,37);

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H


% %initial verdier posisjon A   ========================================

fi_z = atan2(rH_A(2,1),rH_A(1,1))-pi/2       %rotasjon om z aksen

l3 = norm(-s1B+s1A+rH_A);
R1 = norm(-s2B+s2C);
R2 = norm(-s3C+s3H);

alpha_1 = acos((R2^2 - R1^2 - l3^2)/(-2*R1*l3));
alpha_2 = acos((l3^2 - R1^2 - R2^2)/(-2*R1*R2));

l3_xyz = -s1B+s1A+rH_A;
l3_norm = norm([l3_xyz(2,1);l3_xyz(1,1)]);
alpha_3 = atan(l3_xyz(3,1)/l3_norm);

fi_2x = alpha_1 + alpha_3       %Initial rotasjon om x-aksen for kranarm 2
```

```
s_AB = -s1A + s1B;
s_CH = -s3C + s3H;
s_BC = -s2B + s2C;

alpha_4 = abs(atan(s_CH(3,1)/s_CH(2,1)));   %Vinkel koordinatsys 3 i
forhold til R2

fi_3x = alpha_1+alpha_3-pi+alpha_2+alpha_4  % Vinkelorientering
koordinatsys 3

v_1 = [0;0;fi_z];
v_2 = [fi_2x;0;fi_z];
v_3 = [fi_3x;0;fi_z];

pos_2_xyz = A(v_1)*s_AB + A(v_2)*(-s2B);

pos_3_xyz = A(v_1)*s_AB + A(v_2)*s_BC + A(v_3)*(-s3C);

pos_D = A(v_1)*(-s1A + s1D)

pos_E = A(v_1)*s_AB + A(v_2)*(-s2B+s2E)

pos_F = A(v_1)*s_AB + A(v_2)*(-s2B+s2F)

pos_G = A(v_1)*s_AB + A(v_2)*s_BC + A(v_3)*(-s3C+s3G)

d1 = norm(pos_E-pos_D)
d2 = norm(pos_G-pos_F)

q=[-s1A', v_1', pos_2_xyz', v_2', pos_3_xyz', v_3']'

% Beregner posisjon ======================================================

% q_A=NewtonR_AB('jac_AB','phi_A',q,x)
q_A = fzerom_par_AB('phi_A',q,x)

r1 = q_A(1:3,1);
r2 = q_A(7:9,1);
r3 = q_A(13:15,1);

v1 = q_A(4:6,1);       %v1 (inneholdene phi1, theta1 og tau1)
v2 = q_A(10:12,1);     %v2 (inneholdene phi2, theta2 og tau2)
v3 = q_A(16:18,1);     %v3 (inneholdene phi3, theta3 og tau3)

    A1 = A(v1);          %transformasjonsmatrise for v1
    A2 = A(v2);          %transformasjonsmatrise for v2
    A3 = A(v3);          %transformasjonsmatrise for v3

d1_A = r2 + A2*s2E - A1*s1D - r1;
d2_A = r3 + A3*s3G - A2*s2F - r2;

global VINKEL2_A VINKEL3_A l_SYLINDER1A l_SYLINDER2A
VINKEL2_A = q_A(10,1);
VINKEL3_A = q_A(10,1)-q_A(17,1)+alpha_4;
l_SYLINDER1A = norm(d1_A)
l_SYLINDER2A = norm(d2_A)

a(1:21,1) =0;
```

```
a(1,1) = norm(d1_A);                    %lengde sylinder 1 pos A
a(2,1) = norm(d2_A);                    %lengde sylinder 2 pos A
a(3,1) = atand(rH_A(1,1)/rH_A(2,1));     %rotasjonsvinkel pos A
a(4:21,1) = q_A;



%Konstanter=============================================================
function a = konstanter

%Simuleringstid=========================================================
t_min=0;                 %start tid
dt=0.2;                 %tidssteg
t_max=10;                 %slutt tid

%Funksjon (rettningsventil)=============================================
t_start = 0;          %startverdi for funksjonen
t_open = 0.2;             %setter åpne og lukketid for ventilen
t_intervall_1 = t_max-(2*t_open);      %setter intervallet for funksjonen


Q_max1 = 0.015;         %900 [l/min]
Q_max2 = 0.015;         %900 [l/min]
Q_max3 = 0.015;         %900 [l/min]
%Kranposisjon A og B====================================================
rH_A = [0 35 15];               %pos A krantupp
rH_B = [0 15 5];               %pos B krantupp

last = 10000;
%Sylinder areal=========================================================
rad1 = 0.32/2;                      %Radius stempel sylinder 1, [m]
rad2 = 0.23/2;                     %Radius stempelstand sylinder 1, [m]
rad3 = 0.30/2;                      %Radius stempel sylinder 2, [m]
rad4 = 0.20/2;                     %Radius stempelstand sylinder 2, [m]

syl1_areal_pos = 2*pi*rad1;
syl1_areal_neg = 2*pi*(rad1-rad2);
syl2_areal_pos = 2*pi*rad3;
syl2_areal_neg = 2*pi*(rad3-rad4);

%Fortrengningsvolum rotasjon============================================
displacement = 0.05;         %m^3/omdreining

%Lengder kran===========================================================
h =3.85;                    %Høyde krantårn (z-retning)
b1=1;                       %bredde (3D anim)
D1=2;                       %dybde (3D anim)

% L1 = 23.5;                  %Lengde kranarm 1
% L2 = 12;                    %Lengde kranarm 2

k = 1;

s1A = k*[0;0;-h/2];           %lengde fra lokal koordinat 1 til joint A
s1B = k*[0;0;h/2];            %lengde fra lokal koordinat 1 til joint B
s1D = k*[0;1.75;-1.475];      %lengde fra lokal koordinat 1 til joint D
s2B = k*[0;-10.666;-0.37];    %lengde fra lokal koordinat 2 til joint B
```

```
s2C = k*[0;12.834;-0.37];     %lengde fra lokal koordinat 2 til joint C
s2E = k*[0;-4.666;-1];        %lengde fra lokal koordinat 2 til joint E
s2F = k*[0;6.484;-1];         %lengde fra lokal koordinat 2 til joint F
s3C = k*[0;-7.054;2.805];     %lengde fra lokal koordinat 3 til joint C
s3G = k*[0;-4.804;2.243];    %lengde fra lokal koordinat 3 til joint G
s3H = k*[0;9.107;-6.326];     %lengde fra lokal koordinat 3 til krantupp


%initial lengder Sylinder 1 og 2=======================================

syl1_lengde = (s1B-s2B+s2E) - (s1D);
l_1_init = norm(syl1_lengde);

syl2_lengde = (s2C-s3C+s3G) - (s2F);
l_2_init = norm(syl2_lengde);

%Hastighet og akselerasjon hydrauliske sylindre====================
v_1 = 0.2;                          %hastighet sylinder 1
v_2 = -0.2;                         %hastighet sylinder 2
a_1 = 0;                            %akselerasjon sylinder 1
a_2 = 0;                            %akselerasjon sylinder 2

%Masse krantårn og kranarm 1 og 2======================================
m1= 2000;   %masse krantårn
m2= 1500;   %masse kranarm1
m3= 1000;   %masse kranarm2

M1= eye(3)*m1;
M2= eye(3)*m2;
M3= eye(3)*m3;
%Treghetsmoment krantårn og kranarm 1 og 2============================0
% Inertia tensors as given (unit is T*mm^2)
j1 = [8.0028345e7 -9.737385e6 1.9917595e7;...
      -9.737385e6 8.2777918e7 2.1722351e5;...
      1.9917595e7 2.1722351e5 9.8183294e7];

j2 = [8.5127790e8 5.6483637e5 1.3313718e6;...
      5.6483637e5 8.4813949e8 -9.3820534e6;...
      1.3313718e6 -9.3820534e6 1.3943732e7];

j3 = [3.9622348e8 -1.3399605e6 -1.5215701e6;...
      -1.3399605e6 2.7403523e8 -1.7760645e8;...
      -1.5215701e6 -1.7760645e8 1.2470308e8];
%=========================================================================
%transformation into model coordinate system (unit is kg*m^2)
A_units = [0 0 -1; -1 0 0; 0 1 0];

J1 = A_units*j1*A_units'*1e-3;
J2 = A_units*j2*A_units'*1e-3;
J3 = A_units*j3*A_units'*1e-3;


%=========================================================================

a(1:3,1:45) = 0;
a(1:3,1:10)  = [s1A, s1B, s1D,s2B,s2C,s2E,s2F,s3C,s3G,s3H]; %pos koordinat
a(1:3,13)    = [h;b1;D1];     %------------ høgde bredde og dybde krantårn
a(1:3,15)    = [0;v_1;v_2];    %--------------- hastighet sylinder 1 og 2
a(1:3,16)    = [0;a_1;a_2];    %---------------akselerasjon sylinder 1 og 2
```

```
a(1:3,17)    = [m1;m2;m3];  %--------------- masse krantårn, kranarm 1 og 2
a(1:3,18:20) = J1;   %---------------------------- treghetsmoment krantårn
a(1:3,21:23) = J2;   %---------------------------- treghetsmoment kranarm 1
a(1:3,24:26) = J3;   %---------------------------- treghetsmoment kranarm 2
a(1:3,27:29) = M1;
a(1:3,30:32) = M2;
a(1:3,33:35) = M3;
a(1:3,36)    = [0;0;-9.81];  %----------------------- gravitasjon 9,81m/s^2
a(1:3,37)    = rH_A;      %-----------------------------Startposisjon
a(1:3,38)    = rH_B;        %----------------------------Sluttposisjon
a(1,39)      = syl1_areal_pos;
a(2,39)      = syl1_areal_neg;
a(1,40)      = syl2_areal_pos;
a(2,40)      = syl2_areal_neg;
a(3,40)      = displacement;
a(1,41)      = t_start;
a(2,41)      = t_intervall_1;
a(3,41)      = t_open;
a(1,42)      = t_min;
a(2,42)      = dt;
a(3,42)      = t_max;
a(1,43)      = l_1_init;
a(2,43)      = l_2_init;
a(1:3,44)    = [Q_max1; Q_max2; Q_max3];
a(1,45)      = last;
```

```
%=========================================================================
function a = pos_B(x)

k=konstanter;

h = k(1,13);      %Høyde krantårn (z-retning)
b1 = k(2,13);       %bredde (y-retning)
D1 = k(3,13);       %dybde (x-retning)

rH_B = k(1:3,38);

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H
```

```
% %initial verdier posisjon B   ========================================

fi_z = atan2(rH_B(2,1),rH_B(1,1))-pi/2       %rotasjon om z aksen

l3 = norm(-s1B+s1A+rH_B);
R1 = norm(-s2B+s2C);
R2 = norm(-s3C+s3H);

alpha_1 = acos((R2^2 - R1^2 - l3^2)/(-2*R1*l3));
alpha_2 = acos((l3^2 - R1^2 - R2^2)/(-2*R1*R2));

l3_xyz = -s1B+s1A+rH_B;
l3_norm = norm([l3_xyz(2,1);l3_xyz(1,1)]);
alpha_3 = atan(l3_xyz(3,1)/l3_norm);

fi_2x = alpha_1 + alpha_3       %Initial rotasjon om x-aksen for kranarm 2

s_AB = -s1A + s1B;
s_CH = -s3C + s3H;
s_BC = -s2B + s2C;

alpha_4 = abs(atan(s_CH(3,1)/s_CH(2,1)));   %Vinkel koordinatsys 3 i
forhold til R2

fi_3x = alpha_1+alpha_3-pi+alpha_2+alpha_4  % Vinkelorientering
koordinatsys 3

v_1 = [0;0;fi_z];
v_2 = [fi_2x;0;fi_z];
v_3 = [fi_3x;0;fi_z];

pos_2_xyz = A(v_1)*s_AB + A(v_2)*(-s2B);

pos_3_xyz = A(v_1)*s_AB + A(v_2)*s_BC + A(v_3)*(-s3C);

pos_D = A(v_1)*(-s1A + s1D)

pos_E = A(v_1)*s_AB + A(v_2)*(-s2B+s2E)

pos_F = A(v_1)*s_AB + A(v_2)*(-s2B+s2F)

pos_G = A(v_1)*s_AB + A(v_2)*s_BC + A(v_3)*(-s3C+s3G)

d1 = norm(pos_E-pos_D)
d2 = norm(pos_G-pos_F)

q=[-s1A', v_1', pos_2_xyz', v_2', pos_3_xyz', v_3']'

% Beregner posisjon ======================================================

% q_A=NewtonR_AB('jac_AB','phi_A',q,x)
q_B = fzerom_par_AB('phi_B',q,x)


r1 = q_B(1:3,1);
r2 = q_B(7:9,1);
```

```
r3 = q_B(13:15,1);

v1 = q_B(4:6,1);       %v1 (inneholdene phi1, theta1 og tau1)
v2 = q_B(10:12,1);     %v2 (inneholdene phi2, theta2 og tau2)
v3 = q_B(16:18,1);     %v3 (inneholdene phi3, theta3 og tau3)

    A1 = A(v1);          %transformasjonsmatrise for v1
    A2 = A(v2);          %transformasjonsmatrise for v2
    A3 = A(v3);          %transformasjonsmatrise for v3

d1_B = r2 + A2*s2E - A1*s1D - r1;
d2_B = r3 + A3*s3G - A2*s2F - r2;

global VINKEL2_B VINKEL3_B l_SYLINDER1B l_SYLINDER2B
VINKEL2_B = q_B(10,1);
VINKEL3_B = q_B(10,1)-q_B(17,1)+alpha_4;
l_SYLINDER1B = norm(d1_B)
l_SYLINDER2B = norm(d2_B)

a(1:3,1) = 0;
a(1,1) = norm(d1_B);                    %lengde sylinder 1 pos B
a(2,1) = norm(d2_B);                    %lengde sylinder 2 pos B
a(3,1) = atand(rH_B(1,1)/rH_B(2,1));      %rotasjonsvinkel pos B


function fi=phi_A(q,x)


%Newton Raphson ====================================
function NewR=NewtonR_AB(jac_AB,phi,q,x)

tol=1e-10;
error=1;
n_max=150;
i=0;

while error>tol || i<n_max

    i=i+1;
J=feval(jac_AB,q,x);
P=feval(phi,q,x);
delta_q = J\P;
q= q - delta_q;
error=norm(delta_q);

end
NewR=q;



%Phi_A=============================================================================
%Henter konstanter=========================================================
k=konstanter;

rH_A = k(1:3,37);

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
```

```
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H
%=================================================================
r1=q(1:3,1);    %x1,y1,z1
r2=q(7:9,1);    %x2,y2,z2
r3=q(13:15,1);  %x3,y3,z3

v1 = q(4:6,1);  %w1 (inneholdene phi1, theta1 og tau1)
v2 = q(10:12,1);%w2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);%w3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1); %transformasjonsmatrise for v1
A2 = A(v2); %transformasjonsmatrise for v2
A3 = A(v3); %transformasjonsmatrise for v3

u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';


fi(1:18,1) = 0;
fi(1:3,1) = r1 + A1*s1A;
fi(4,1) = u_x'*A1*u_z;
fi(5,1) = u_y'*A1*u_z;
fi(6:8,1) = r1 + A1*s1B - A2*s2B - r2;
fi(9,1) = (A1*u_x)'*(A2*u_z);
fi(10,1) = (A1*u_x)'*(A2*u_y);
fi(11:13,1) = r2 + A2*s2C - r3 - A3*s3C;
fi(14,1) = (A2*u_x)'*(A3*u_z);
fi(15,1) = (A2*u_x)'*(A3*u_y);
fi(16:18,1) = r3 + A3*s3H - rH_A;




%Phi_B=============================================================
function fi=phi_B(q,x)

%Henter konstanter=================================================
k=konstanter;


rH_B = k(1:3,38);

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
```

```
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H


%====================================================================
r1=q(1:3,1);    %x1,y1,z1
r2=q(7:9,1);    %x2,y2,z2
r3=q(13:15,1);  %x3,y3,z3

v1 = q(4:6,1);  %w1 (inneholdene phi1, theta1 og tau1)
v2 = q(10:12,1);%w2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);%w3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1); %transformasjonsmatrise for v1
A2 = A(v2); %transformasjonsmatrise for v2
A3 = A(v3); %transformasjonsmatrise for v3

u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';


fi(1:18,1) = 0;
fi(1:3,1) = r1 + A1*s1A;
fi(4,1) = u_x'*A1*u_z;
fi(5,1) = u_y'*A1*u_z;
fi(6:8,1) = r1 + A1*s1B - A2*s2B - r2;
fi(9,1) = (A1*u_x)'*(A2*u_z);
fi(10,1) = (A1*u_x)'*(A2*u_y);
fi(11:13,1) = r2 + A2*s2C - r3 - A3*s3C;
fi(14,1) = (A2*u_x)'*(A3*u_z);
fi(15,1) = (A2*u_x)'*(A3*u_y);
fi(16:18,1) = r3 + A3*s3H - rH_B;



%Jackobian=============================================
function jacobi=jac_AB(q,x)

%Henter konstanter=========================================================
k=konstanter;

rH = k(1:3,37);
```

```
h = k(1,13);      %Høyde krantårn (z-retning)
b1 = k(2,13);       %bredde (y-retning)
D1 = k(3,13);       %dybde (x-retning)

L1 = k(1,14);       %Lengde kranarm 1
L2 = k(2,14);       %Lengde kranarm 2

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H


%=========================================================================
r1=q(1:3,1);    %x1,y1,z1
r2=q(7:9,1);    %x2,y2,z2
r3=q(13:15,1);  %x3,y3,z3


v1 = q(4:6,1);  %w1 (inneholdene phi1, theta1 og tau1)
v2 = q(10:12,1);%w2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);%w3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1); %transformasjonsmatrise for v1
A2 = A(v2); %transformasjonsmatrise for v2
A3 = A(v3); %transformasjonsmatrise for v3

u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';

skew_s1A = skew(s1A);
skew_s1B = skew(s1B);
skew_s1D = skew(s1D);

skew_s2B = skew(s2B);
skew_s2C = skew(s2C);
skew_s2E = skew(s2E);
skew_s2F = skew(s2F);

skew_s3C = skew(s3C);
skew_s3G = skew(s3G);
skew_s3H = skew(s3H);
```

```
skew_u_x = skew(u_x);
skew_u_y = skew(u_y);
skew_u_z = skew(u_z);


jacobi(1:18,1:18)=0;
jacobi(1:3,1:6)=    [eye(3), -A1*skew_s1A];
jacobi(4,4:6) =     -u_x'*A1*skew_u_z;
jacobi(5,4:6) =     -u_y'*A1*skew_u_z;
jacobi(6:8,1:12) =  [eye(3), -A1*skew_s1B, -eye(3), A2*skew_s2B];
jacobi(9,4:12) =    [(A2*u_z)'*(-A1*skew_u_x), zeros(1,3),...
    (A1*u_x)'*(-A2*skew_u_z)];
jacobi(10,4:12) =   [(A2*u_y)'*(-A1*skew_u_x), zeros(1,3),...
    (A1*u_x)'*(-A2*skew_u_y)];
jacobi(11:13,7:18) =[eye(3), -A2*skew_s2C, -eye(3), A3*skew_s3C];
jacobi(14,10:18) =  [(A3*u_z)'*(-A2*skew_u_x), zeros(1,3),...
    (A2*u_x)'*(-A3*skew_u_z)];
jacobi(15,10:18) =  [(A3*u_y)'*(-A2*skew_u_x), zeros(1,3),...
    (A2*u_x)'*(-A3*skew_u_y)];
jacobi(16:18,13:18)=[eye(3), -A3*skew_s3H];

 %Akselerasjon sylinder 1==================================
function y = Q_syl1_acc(t,V1,x)


k = konstanter;

Q_max1 = k(1,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V1 ==0
    u_max = 0;
else
u_max = V1 / ((t_open + t_intervall) * Q_max1);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;

    if t>=t1 && t<t2

y=(0.5*u_max*Q_max1*(pi/t_open))*cos((pi*t/t_open)-((pi*t1)/t_open) -pi/2);

        elseif t>=t2 && t<t3
y=0;
        elseif t>=t3 && t<t4

y=(-0.5*u_max*Q_max1*(pi/t_open))*cos((pi*t/t_open)-((pi*t3)/t_open)-pi/2);

    else
        y=0;
```

```
    end


%Hastighet sylinder 1================================
function y = Q_syl1_vel(t,V1,x)


k = konstanter;

Q_max1 = k(1,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V1 ==0
    u_max = 0;
else
u_max = V1 / ((t_open + t_intervall) * Q_max1);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;



    if t>=t1 && t<t2

   y=[(sin((pi/t_open)*t-pi/2-(pi/t_open)*t1)/2+0.5)*u_max*Q_max1,u_max];

        elseif t>=t2 && t<t3
y=[1*u_max*Q_max1,u_max];
        elseif t>=t3 && t<t4

  y=[(sin((pi/t_open)*t-3*pi/2-(pi/t_open)*t3)/2+0.5)*u_max*Q_max1,u_max];

    else
        y=[0,u_max];

    end




%Akselerasjon sylinder 1==================================
function y = Q_syl1_acc(t,V1,x)


k = konstanter;

Q_max1 = k(1,44);
```

```
t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V1 ==0
    u_max = 0;
else
u_max = V1 / ((t_open + t_intervall) * Q_max1);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;

    if t>=t1 && t<t2

y=(0.5*u_max*Q_max1*(pi/t_open))*cos((pi*t/t_open)-((pi*t1)/t_open) -pi/2);

        elseif t>=t2 && t<t3
y=0;
        elseif t>=t3 && t<t4

y=(-0.5*u_max*Q_max1*(pi/t_open))*cos((pi*t/t_open)-((pi*t3)/t_open)-pi/2);

    else
        y=0;

    end


%Posisjon sylinder2==================================
function y = Q_syl2_pos(t,V2,x)




k = konstanter;

Q_max2 = k(2,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V2 ==0
    u_max = 0;
else
u_max = V2 / ((t_open + t_intervall) * Q_max2);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
```

```
t4 = t3 + t_open;


if t<t1
    y=0;
elseif t>=t1 && t<=t2

        y=((-t_open*u_max*Q_max2)/(2*pi))*cos(((pi*t)/t_open)...
            - ((pi*t1)/t_open) - pi/2) + u_max*Q_max2*0.5*(t-t1);

       elseif t>=t2 && t<t3
         y1=((-t_open*u_max*Q_max2)/(2*pi))*cos(((pi*t2)/t_open)...
             - ((pi*t1)/t_open) - pi/2) + u_max*Q_max2*0.5*(t2-t1);

        y=((t-t2)*u_max*Q_max2)+ y1;

        elseif t>=t3 && t<t4
             y1=((-t_open*u_max*Q_max2)/(2*pi))*cos(((pi*t2)/t_open)...
                 - ((pi*t1)/t_open) - pi/2) + u_max*Q_max2*0.5*(t2-t1);
            y2=((t3-t2)*u_max*Q_max2)+ y1;

        y=((-t_open*u_max*Q_max2)/(2*pi))*cos(((pi*t)/t_open)...
            - ((pi*t3)/t_open) + pi/2) + u_max*Q_max2*0.5*(t-t3) + y2;
else
y1=((-t_open*u_max*Q_max2)/(2*pi))*cos(((pi*t2)/t_open)...
    - ((pi*t1)/t_open) - pi/2) + u_max*Q_max2*0.5*(t2-t1);
        y2=((t3-t2)*u_max*Q_max2)+ y1;

        y=((-t_open*u_max*Q_max2)/(2*pi))*cos(((pi*t4)/t_open)...
            - ((pi*t3)/t_open) + pi/2) + u_max*Q_max2*0.5*(t4-t3) + y2;

end



%Hastighet sylinder 2=====================================
function y = Q_syl2_vel(t,V2,x)

k = konstanter;

Q_max2 = k(2,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V2 ==0
    u_max = 0;
else
u_max = V2 / ((t_open + t_intervall) * Q_max2);
end



t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;
```

```
    if t>=t1 && t<t2

    y=[(sin((pi/t_open)*t-pi/2-(pi/t_open)*t1)/2+0.5)*u_max*Q_max2,u_max];

        elseif t>=t2 && t<t3
y=[1*u_max*Q_max2,u_max];
        elseif t>=t3 && t<t4

    y=[(sin((pi/t_open)*t-3*pi/2-
(pi/t_open)*t3)/2+0.5)*u_max*Q_max2,u_max];

    else
        y=[0,u_max];

    end



%Akselerasjon sylinder 2====================================
function y = Q_syl2_acc(t,V2,x)

k = konstanter;

Q_max2 = k(2,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V2 ==0
    u_max = 0;
else
u_max = V2 / ((t_open + t_intervall) * Q_max2);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;

    if t>=t1 && t<t2

y=(0.5*u_max*Q_max2*(pi/t_open))*cos((pi*t/t_open)-((pi*t1)/t_open)- pi/2);

        elseif t>=t2 && t<t3
y=0;
        elseif t>=t3 && t<t4

y=(-0.5*u_max*Q_max2*(pi/t_open))*cos((pi*t/t_open)-((pi*t3)/t_open)-pi/2);

    else
        y=0;

    end
```

```
%Posisjon rotsjonsdriver==============================
function y = Q_rot_pos(t,V3,x)

k = konstanter;

Q_max3 = k(3,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V3 ==0
    u_max = 0;
else
u_max = V3 / ((t_open + t_intervall) * Q_max3);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;


if t<t1
    y=0;
elseif t>=t1 && t<=t2

        y=((-t_open*u_max*Q_max3)/(2*pi))*cos(((pi*t)/t_open)...
            - ((pi*t1)/t_open) - pi/2) + u_max*Q_max3*0.5*(t-t1);

       elseif t>=t2 && t<t3
         y1=((-t_open*u_max*Q_max3)/(2*pi))*cos(((pi*t2)/t_open)...
             - ((pi*t1)/t_open) - pi/2) + u_max*Q_max3*0.5*(t2-t1);

        y=((t-t2)*u_max*Q_max3)+ y1;

        elseif t>=t3 && t<t4
             y1=((-t_open*u_max*Q_max3)/(2*pi))*cos(((pi*t2)/t_open)...
                 - ((pi*t1)/t_open) - pi/2) + u_max*Q_max3*0.5*(t2-t1);
            y2=((t3-t2)*u_max*Q_max3)+ y1;

        y=((-t_open*u_max*Q_max3)/(2*pi))*cos(((pi*t)/t_open)...
            - ((pi*t3)/t_open) + pi/2) + u_max*Q_max3*0.5*(t-t3) + y2;
else
y1=((-t_open*u_max*Q_max3)/(2*pi))*cos(((pi*t2)/t_open)...
    - ((pi*t1)/t_open) - pi/2) + u_max*Q_max3*0.5*(t2-t1);
        y2=((t3-t2)*u_max*Q_max3)+ y1;

        y=((-t_open*u_max*Q_max3)/(2*pi))*cos(((pi*t4)/t_open)...
            - ((pi*t3)/t_open) + pi/2) + u_max*Q_max3*0.5*(t4-t3) + y2;

end
```

```
%Hastighet rotsjonsdriver==================================
function y = Q_rot_vel(t,V3,x)


k = konstanter;

Q_max3 = k(3,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
t_open = k(3,41);             %setter åpne og lukketid for ventilen

if V3 ==0
    u_max = 0;
else
u_max = V3 / ((t_open + t_intervall) * Q_max3);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;


    if t>=t1 && t<t2

   y=[(sin((pi/t_open)*t-pi/2-(pi/t_open)*t1)/2+0.5)*u_max*Q_max3,u_max];

        elseif t>=t2 && t<t3
y=[1*u_max*Q_max3,u_max];
        elseif t>=t3 && t<t4

   y=[(sin((pi/t_open)*t-3*pi/2-(pi/t_open)*t3)/2+0.5)*u_max*Q_max3,u_max];

    else
        y=[0,u_max];
    end




%Akselerasjon rotasjonsdriver==============================
function y = Q_rot_acc(t,V3,x)

k = konstanter;

Q_max3 = k(3,44);

t_start = k(1,41);          %startverdi for funksjonen
t_intervall = k(2,41);      %setter intervallet for funksjonen
```

```
t_open = k(3,41);              %setter åpne og lukketid for ventilen

if V3 ==0
    u_max = 0;
else
u_max = V3 / ((t_open + t_intervall) * Q_max3);
end


t1 = t_start;
t2 = t1 + t_open;
t3 = t2 + t_intervall;
t4 = t3 + t_open;

    if t>=t1 && t<t2

y=(0.5*u_max*Q_max3*(pi/t_open))*cos((pi*t/t_open)-((pi*t1)/t_open)- pi/2);

        elseif t>=t2 && t<t3
y=0;
        elseif t>=t3 && t<t4

y=(-0.5*u_max*Q_max3*(pi/t_open))*cos((pi*t/t_open)-((pi*t3)/t_open)-pi/2);

    else
        y=0;

    end


%Resultat=============================================================
function e=resultat(x)

k=konstanter;

h = k(1,13);       %Høyde krantårn (z-retning)
b1 = k(2,13);        %bredde (y-retning)
D1 = k(3,13);        %dybde (x-retning)

% L1 = k(1,14);        %Lengde kranarm 1
% L2 = k(2,14);        %Lengde kranarm 2

s1A = k(1:3,1);      %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);      %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);      %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);      %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);      %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);      %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);     %endring i y
s2E(3,1) = s2E(3,1)*x(6);     %endring i z
s2F = k(1:3,7)*x(1);      %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);     %endring i y
s2F(3,1) = s2F(3,1)*x(8);     %endring i z
s3C = k(1:3,8)*x(2);      %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);      %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);    %endring i y
```

```
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H


m1 = k(1,17);        %masse krantårn
m2 = x(1)*k(2,17);       %masse kranarm1
m3 = x(2)*k(3,17);       %masse kranarm2

M1 = k(1:3,27:29);
M2 = k(1:3,30:32);
M3 = k(1:3,33:35);

J1 = k(1:3,18:20);
J2 = x(1)^3*k(1:3,21:23);
J3 = x(2)^3*k(1:3,24:26);

g = k(1:3,36);         %gravitasjonskraft

%=========================================================================
====
u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';

% Areal sylinderstempel positiv og negativ retning +
rotdriver==================================0
syl1_areal_pos = k(1,39);
syl1_areal_neg = k(2,39);
syl2_areal_pos = k(1,40);
syl2_areal_neg = k(2,40);
displacement = k(3,40);    %  m^3/rotasjon

%Beregner nødvendig oljevolum (forflyttning fra A til
B)================================================
posA = pos_A(x);

l_syl1_A = posA(1,1);
l_syl2_A = posA(2,1);
v_A = posA(3,1);

posB = pos_B(x);

l_syl1_B = posB(1,1);
l_syl2_B = posB(2,1);
v_B = posB(3,1);

sylinder1 = l_syl1_B - l_syl1_A;       %forflyttning sylinder 1
sylinder2 = l_syl2_B - l_syl2_A;       %forflyttning sylinder 2
rotasjon = v_B - v_A;                  %vinkel-rotasjon [rad]

if sylinder1 >= 0
    V1 = syl1_areal_pos * abs(sylinder1);
else
    V1 = syl1_areal_neg * abs(sylinder1);
end

if sylinder2 >= 0
    V2 = syl2_areal_pos * abs(sylinder2);
else
```

```
    V2 = syl2_areal_neg * abs(sylinder2);
end
 V3 =displacement * (abs(rotasjon)/2*pi);


%Initial verdier
=====================================================================
l_1_init = l_syl1_A;
l_2_init = l_syl2_A;
q = posA(4:21,1);

l_syl1 = sylinder1;
l_syl2 = sylinder2;
rot_driv = rotasjon;



% Simulering
=========================================================================
==================
t_min= k(1,42);                 %start tid
dt= k(2,42);                    %tidssteg
t_max= k(3,42);                 %slutt tid
t=[t_min:dt:t_max]';
N=size(t,1);                    %antall iterasjoner

Y(1:N,1:18)=0;
Yd(1:N,1:18)=0;
Ydd(1:N,1:18)=0;

YA(1:N,1:6)=0;
YB(1:N,1:6)=0;
YC(1:N,1:6)=0;

Ysyl1(1:N,1:2)=0;
Ysyl2(1:N,1:2)=0;

Y_syl1(1:N,1)=0;
Y_syl2(1:N,1)=0;

Y_d1(1:N,1)=0;
Y_d2(1:N,1)=0;

Y_d1d(1:N,1)=0;
Y_d2d(1:N,1)=0;

Y_lt1(1:N,1) = 0;
Y_lt1d(1:N,1) = 0;
Y_lt1dd(1:N,1) = 0;

Y_lt2(1:N,1) = 0;
Y_lt2d(1:N,1) = 0;
Y_lt2dd(1:N,1) = 0;

Y_p1(1:N,1)=0;
Y_p2(1:N,1)=0;
```

```
for i=1:N

q = fzerom_par('phi',q,V1,V2,V3,l_1_init,l_2_init,l_syl1,l_syl2,x,t(i,1));

    r1=q(1:3,1);        %x1,y1,z1
    r2=q(7:9,1);        %x2,y2,z2
    r3=q(13:15,1);      %x3,y3,z3

    v1 = q(4:6,1);      %v1 (inneholdene phi1, theta1 og tau1)
    v2 = q(10:12,1);    %v2 (inneholdene phi2, theta2 og tau2)
    v3 = q(16:18,1);    %v3 (inneholdene phi3, theta3 og tau3)

    A1 = A(v1);         %transformasjonsmatrise for v1
    A2 = A(v2);         %transformasjonsmatrise for v2
    A3 = A(v3);         %transformasjonsmatrise for v3

        % Drivefunksjon Sylinder 1=====================================

        if l_syl1 >= 0
        lt1=(Q_syl1_pos(t(i,1),V1,x)/syl1_areal_pos) + l_1_init;
        else
        lt1=(-Q_syl1_pos(t(i,1),V1,x)/syl1_areal_neg) + l_1_init;
        end

        if l_syl1 >= 0
        lt1d=(Q_syl1_vel(t(i,1),V1,x)/syl1_areal_pos);
        else
        lt1d=(-Q_syl1_vel(t(i,1),V1,x)/syl1_areal_neg);
        end

        if l_syl1 >= 0
        lt1dd=(Q_syl1_acc(t(i,1),V1,x)/syl1_areal_pos);
        else
        lt1dd=(-Q_syl1_acc(t(i,1),V1,x)/syl1_areal_neg);
        end
        %Drivefunksjon Sylinder 2=====================================
        if l_syl2 >= 0
        lt2 = (Q_syl2_pos(t(i,1),V2,x)/syl2_areal_pos) + l_2_init;
        else
        lt2 = (-Q_syl2_pos(t(i,1),V2,x)/syl2_areal_neg) + l_2_init;
        end

        if l_syl2 >=0
        lt2d = (Q_syl2_vel(t(i,1),V2,x)/syl2_areal_pos);
        else
        lt2d = (-Q_syl2_vel(t(i,1),V2,x)/syl2_areal_neg);
        end

        if l_syl2 >=0
        lt2dd = (Q_syl2_acc(t(i,1),V2,x)/syl2_areal_pos);
        else
        lt2dd = (-Q_syl2_acc(t(i,1),V2,x)/syl2_areal_neg);
        end

        %Drivefunksjon rotasjon=======================================
        if rot_driv >=0
        lt3 = (Q_rot_pos(t(i,1),V3,x)/displacement);
        else
        lt3 = (-Q_rot_pos(t(i,1),V3,x)/displacement);
```

```
        end

        if rot_driv >=0
        lt3d = (Q_rot_vel(t(i,1),V3,x)/displacement);
        else
        lt3d = (-Q_rot_vel(t(i,1),V3,x)/displacement);
        end

        if rot_driv >=0
        lt3dd = (-Q_rot_acc(t(i,1),V3,x)/displacement);
        else
        lt3dd = (Q_rot_acc(t(i,1),V3,x)/displacement);
        end

%Hastighet=============================================================
qd = jac(q,x)\[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 lt3d(1,1) 2*lt1*lt1d(1,1)
2*lt2*lt2d(1,1)]';

    r1d=qd(1:3,1);          %x1,y1,z1
    r2d=qd(7:9,1);          %x2,y2,z2
    r3d=qd(13:15,1);        %x3,y3,z3

    v1d = qd(4:6,1);        %v1 (inneholdene phi1, theta1 og tau1)
    v2d = qd(10:12,1);      %v2 (inneholdene phi2, theta2 og tau2)
    v3d = qd(16:18,1);      %v3 (inneholdene phi3, theta3 og tau3)



     d1 = r1 + A1*s1D - A2*s2E - r2;          %longitudinal driver 1
     d2 = r2 + A2*s2F - A3*s3G - r3;          %longitudinal driver 2

     d1d = r1d + A1*skew(v1d)*s1D - A2*skew(v2d)*s2E - r2d;    % derivert
     d2d = r2d + A2*skew(v2d)*s2F - A3*skew(v3d)*s3G - r3d;    % derivert

%Akselerasjon===========================================================

rh = [-A1*skew(v1d)^2*s1A;
     -u_x'*A1*skew(v1d)^2*u_z;
     -u_y'*A1*skew(v1d)^2*u_z;
     -A1*skew(v1d)^2*s1B + A2*skew(v2d)^2*s2B;
     (-A1*skew(v1d)^2*u_x)'*(A2*u_z)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_z)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_z)...
     + (-A1*u_x)'*(A2*skew(v2d)^2*u_z);
     (-A1*skew(v1d)^2*u_x)'*(A2*u_y)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_y)...
     + (-A1*skew(v1d)*u_x)'*(A2*skew(v2d)*u_y)...
     + (-A1*u_x)'*(A2*skew(v2d)^2*u_y);
     -A2*skew(v2d)^2*s2C + A3*skew(v3d)^2*s3C;
     (-A2*skew(v2d)^2*u_x)'*(A3*u_z)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_z)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_z)...
     + (-A2*u_x)'*(A3*skew(v3d)^2*u_z);
     (-A2*skew(v2d)^2*u_x)'*(A3*u_y)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_y)...
     + (-A2*skew(v2d)*u_x)'*(A3*skew(v3d)*u_y)...
     + (-A2*u_x)'*(A3*skew(v3d)^2*u_y);
      lt3dd;
     (-2*(d1d'*d1d) + 2*lt1d(1,1)*lt1d(1,1)...
```

```
     + 2*lt1*lt1dd -2*d1'*( A1*skew(v1d)*skew(v1d)*s1D...
     - A2*skew(v2d)*skew(v2d)*s2E));
     (-2*(d2d'*d2d) + 2*lt2d(1,1)*lt2d(1,1)...
     + 2*lt2*lt2dd -2*d2'*( A2*skew(v2d)*skew(v2d)*s2F...
     - A3*skew(v3d)*skew(v3d)*s3G))];

qdd = jac(q,x)\rh;

%Reaksjonskrefter ====================================================
    M(1:18,1:18)    =0;
    M(1:3,1:3)      =M1;
    M(4:6,4:6)      =J1;
    M(7:9,7:9)      =M2;
    M(10:12,10:12)  =J2;
    M(13:15,13:15)  =M3;
    M(16:18,16:18)  =J3;


    F_last = [0;0;k(1,45)*g(3,1)];   %Last
    M_last = cross(s3H,A3'*F_last);  %Momentet lasten gir om tyngdepunktet

    F_masse3 = g(3,1)*m3;    %Gravitasjonskrefter masse 3
    F_masse2 = g(3,1)*m2;    %Gravitasjonskrefter masse 2
    F_masse1 = g(3,1)*m1;    %Gravitasjonskrefter masse 1

    g_ext = [0 0 F_masse1 0 0 0 0 0 F_masse2 0 0 0 0 0
(F_masse3+F_last(3,1)) M_last']' ;%-- Eksterne krefter på systemet

    lambda = (jac(q,x)')^-1 * (M*qdd - g_ext) ;  %Lagranges multiplikator


    Jac = jac(q,x)';          %JACOBIAN transponert


    %Krefter joint A ==================================================
    FA_x = Jac(1,1)*lambda(1,1);                  %Krefter joint A x-retning
    FA_y = Jac(2,2)*lambda(2,1);                  %Krefter joint A y-retning
    FA_z = Jac(3,3)*lambda(3,1);                  %Krefter joint A z-retning

    %Krefter joint B ==================================================
    FB_x = Jac(1,6)*lambda(6,1);                  %Krefter joint B x-retning
    FB_y = Jac(2,7)*lambda(7,1);                  %Krefter joint B y-retning
    FB_z = Jac(3,8)*lambda(8,1);                  %Krefter joint B z-retning

    %Krefter joint C ==================================================
    FC_x = Jac(7,11)*lambda(11,1);                %Krefter joint B x-retning
    FC_y = Jac(8,12)*lambda(12,1);                %Krefter joint B y-retning
    FC_z = Jac(9,13)*lambda(13,1);                %Krefter joint B z-retning

    %Sylinderkrefter==================================================

    F_syl1_xyz_dynamisk = Jac(1:3,17)*lambda(17,1);              %Krefter
fra sylinder 1 på legem 1
    F_syl1_xyz_negativ = Jac(7:9,17)*lambda(17,1);               %Krefter
fra sylinder 1 på legem 2

    F_syl1 = norm(F_syl1_xyz_dynamisk);   %Kraft fra sylinder 1 på legem 1
```

```
    F_syl2_xyz_dynamisk = Jac(7:9,18)*lambda(18,1);              %Krefter
fra sylinder 2 på legem 2
    F_syl2_xyz_dynamisk_negativ = Jac(13:15,18)*lambda(18,1);    %Krefter
fra sylinder 2 på legem 3

    F_syl2 = norm(F_syl2_xyz_dynamisk);   %Kraft fra sylinder 1 på legem 1


     %Oljetrykk=============================================================

    kraftretning1 = dot(F_syl1_xyz_dynamisk,d1);
    if kraftretning1 >=0
        p1=(F_syl1*10 + 15*syl1_areal_neg)/syl1_areal_pos;
    else
        p1=(F_syl1*10 + 15*syl1_areal_pos)/syl1_areal_neg;
    end

    kraftretning2 = dot(F_syl2_xyz_dynamisk,d2);
    if kraftretning2 >= 0
        p2=(F_syl2*10 + 15*syl2_areal_neg)/syl2_areal_pos;
    else
        p2=(F_syl2*10 + 15*syl2_areal_pos)/syl2_areal_neg;
    end


%     %Statiske beregninger 2D=========================================

    r_AB = -A1*s1A + A1*s1B;        %Distanse joint A til B
    r_AD = -A1*s1A + A1*s1D;        %Distanse joint A til D
    r_AI = -A1*s1A;                 %Distanse joint A til tyngdepunktet

    r_BC = -A2*s2B + A2*s2C;        %Distanse joint B til C
    r_BF = -A2*s2B + A2*s2F;        %Distanse joint B til F
    r_BJ = -A2*s2B;                 %Distanse joint B til tyngdepunktet
    r_BE = -A2*s2B + A2*s2E;        %Distanse joint B til E

    r_CH = -A3*s3C + A3*s3H;        %Distanse joint C til H
    r_CK = -A3*s3C;                 %Distanse joint C til tyngdepunktet
    r_CG = -A3*s3C + A3*s3G;        %Distanse joint C til G

    u1 = -(1/norm(d1))*d1;           %Egenvektor vektor d1 (sylinder 1)
    u2 = -(1/norm(d2))*d2;           %Egenvektor vektor d2 (sylinder 2)

    F_syl2_statisk = r_CH(2,1)*F_last(3,1)...
        + r_CK(2,1)*F_masse3 / (u2(3,1)*r_CG(2,1) + u2(2,1)*-r_CG(3,1));

    FC_y_statisk = -F_syl2_statisk*u2(2,1);
    FC_z_statisk = -F_syl2_statisk*u2(3,1) + F_masse3 + F_last(3,1);

    F_syl1_statisk = ((r_BC(2,1)*FC_z_statisk)...
        - (r_BC(3,1)*FC_y_statisk) + (r_BF(2,1)*F_syl2_statisk*u2(3,1))...
        - (r_BF(3,1)*F_syl2_statisk*u2(2,1))...
        + (r_BJ(2,1)*F_masse2)) / (u1(2,1)*-r_BE(3,1) + u1(3,1)*r_BE(2,1));

    FB_y_statisk = FC_y_statisk + F_syl2_statisk*u2(2,1)...
        - F_syl1_statisk*u1(2,1);
    FB_z_statisk = FC_z_statisk + F_syl2_statisk*u2(3,1)...
        + F_masse2 - F_syl1_statisk*u1(3,1);
```

```
    FA_y_statisk = FB_y_statisk + F_syl1_statisk*u1(2,1);
    FA_z_statisk = FB_z_statisk + F_syl1_statisk*u1(3,1) + F_masse1;

   %=====================================================================

%
%   %=====================================================================
    Y(i,1:18)=q';
    Yd(i,1:18)=qd';
    Ydd(i,1:18)=qdd';

    YA(i,1:6) = [FA_x,FA_y,FA_z,0,FA_y_statisk,FA_z_statisk];
    YB(i,1:6) = [FB_x,FB_y,FB_z,0,FB_y_statisk,FB_z_statisk];
    YC(i,1:6) = [FC_x,FC_y,FC_z,0,FC_y_statisk,FC_z_statisk];

    Ysyl1(i,1:3)=F_syl1_xyz_dynamisk';
    Ysyl2(i,1:3)=F_syl2_xyz_dynamisk';

    Y_syl1(i,1) = F_syl1;
    Y_syl2(i,1) = F_syl2;

    Y_d1(i,1) = norm(d1);
    Y_d2(i,1) = norm(d2);

    Y_d1d(i,1) = norm(d1d);
    Y_d2d(i,1) = norm(d2d);

    Y_lt1(i,1) = lt1;
    Y_lt1d(i,1) = lt1d(1,1);
    Y_lt1dd(i,1) = lt1dd;

    Y_lt2(i,1) = lt2;
    Y_lt2d(i,1) = lt2d(1,1);
    Y_lt2dd(i,1) = lt2dd;

    Y_p1(i,1)=p1;
    Y_p2(i,1)=p2;

end

% ======================================================================
% Numerisk kontroll

Ydnum(1:N,1:18) = 0;
Yddnum(1:N,1:18) = 0;
for i=1:N-1
    Ydnum(i,1:18) = 1/dt*(Y(i+1,1:18)-Y(i,1:18));
    Yddnum(i,1:18) = 1/dt*(Yd(i+1,1:18)-Yd(i,1:18));
end
Ydnum(N,1:18) = Ydnum(N-1,1:18);
Yddnum(N,1:18) = Yddnum(N-1,1:18);
%
% %PLOT=================================================================

figure(17)

hold on
```

```
plot(t,Y_p1(:,1),'b')
plot(t,Y_p2(:,1),'r')
title('Oljetrykk')
xlabel('tid')
ylabel('trykk, [bar]')
legend('p1','p2')
grid on
box on


figure(14)

hold on
plot(t,Y_lt1(:,1),'b')
plot(t,Y_lt2(:,1),'r')
title('Sylinderlengde')
xlabel('tid, [t]')
ylabel('lengde, [m]')
legend('lt1','lt2')
grid on
box on

figure(15)

hold on
plot(t,Y_lt1d(:,1),'b')
plot(t,Y_lt2d(:,1),'r')
title('Sylinderhastighet')
xlabel('tid, [t]')
ylabel('lengde, [m]')
legend('lt1d','lt2d')
grid on
box on


figure(16)

hold on
plot(t,Y_lt1dd(:,1),'b')
plot(t,Y_lt2dd(:,1),'r')
title('Sylinderakselerasjon')
xlabel('tid, [t]')
ylabel('lengde, [m]')
legend('lt1dd','lt2dd')
grid on
box on


figure(11)

hold on
plot(t,Y_syl1(:,1),'b')
plot(t,Y_syl2(:,1),'r')
title('Sylinderkrefter')
xlabel('tid, [t]')
ylabel('kraft, [N]')
legend('F_{syl1}','F_{syl2}')
grid on
box on
```

```
figure(13)

hold on
plot(t,Y_d1(:,1),'b')
plot(t,Y_d2(:,1),'r')
title('Sylinderlengde')
xlabel('tid, [t]')
ylabel('lengde, [m]')
legend('l_{syl1}','l_{syl2}')
grid on
box on

figure(1)
subplot(2,3,1)
hold on
    plot(t,Y(:,1),'r')
    plot(t,Y(:,2),'b')
    plot(t,Y(:,3),'y')
hold off
legend('x_1','y_1','z_1')
title('krantårn')
xlabel('tid, [t]')
ylabel('posisjon, [m]')
axis([t_min, t_max, min(min(Y(:,1:3)))-2, max(max(Y(:,1:3)))+2])
grid on
box on

subplot(2,3,4)
hold on
    plot(t,Y(:,4),'r')
    plot(t,Y(:,5),'b')
    plot(t,Y(:,6),'y')
hold off
legend('\phi_{1x}','\phi_{1y}','\phi_{1z}')
title('Krantårn')
xlabel('tid, [t]')
ylabel('vinkelposisjon, [rad]')
axis([t_min, t_max, min(min(Y(:,4:6)))-2, max(max(Y(:,4:6)))+2])
grid on
box on

subplot(2,3,2)
hold on
    plot(t,Y(:,7),'r')
    plot(t,Y(:,8),'b')
    plot(t,Y(:,9),'y')
hold off
legend('x_2','y_2','z_2')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('posisjon, [m]')
axis([t_min, t_max, min(min(Y(:,7:9)))-2, max(max(Y(:,7:9)))+2])
grid on
box on

subplot(2,3,5)
hold on
    plot(t,Y(:,10),'r')
    plot(t,Y(:,11),'b')
```

```
    plot(t,Y(:,12),'y')
hold off
legend('\phi_{2x}','\phi_{2y}','\phi_{2z}')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('vinkelposisjon, [rad]')
axis([t_min, t_max, min(min(Y(:,10:12)))-2, max(max(Y(:,10:12)))+2])
grid on
box on


subplot(2,3,3)
hold on
    plot(t,Y(:,13),'r')
    plot(t,Y(:,14),'b')
    plot(t,Y(:,15),'y')
hold off
legend('x_3','y_3','z_3')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('posisjon, [m]')
axis([t_min, t_max, min(min(Y(:,13:15)))-2, max(max(Y(:,13:15)))+2])
grid on
box on

subplot(2,3,6)
hold on
    plot(t,Y(:,16),'r')
    plot(t,Y(:,17),'b')
    plot(t,Y(:,18),'y')
hold off
legend('\phi_{3x}','\phi_{3y}','\phi_{3z}')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('vinkelposisjon, [rad]')
axis([t_min, t_max, min(min(Y(:,16:18)))-2, max(max(Y(:,16:18)))+2])
grid on
box on

%Hastigheter_____________________________________________________

figure(2)

subplot(2,3,1)
hold on
    plot(t,Yd(:,1),'r')
    plot(t,Yd(:,2),'b')
    plot(t,Yd(:,3),'y')

hold off
legend('xd_1','yd_1','zd_1')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Hastighet, [m/s]')
grid on
box on

subplot(2,3,4)
hold on
    plot(t,Yd(:,4),'r')
```

```
    plot(t,Yd(:,5),'b')
    plot(t,Yd(:,6),'y')

hold off
legend('\phid_{1x}','\phid_{1y}','\phid_{1z}')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Vinkelhastighet, [rad/s]')
grid on
box on

subplot(2,3,2)
hold on
    plot(t,Yd(:,7),'r')
    plot(t,Yd(:,8),'b')
    plot(t,Yd(:,9),'y')

hold off
legend('xd_2','yd_2','zd_2')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Hastighet, [m/s]')
grid on
box on

subplot(2,3,5)
hold on
    plot(t,Yd(:,10),'r')
    plot(t,Yd(:,11),'b')
    plot(t,Yd(:,12),'y')

hold off
legend('\phid_{2x}','\phid_{2y}','\phid_{2z}')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Vinkelhastighet, [rad/s]')
grid on
box on


subplot(2,3,3)
hold on
    plot(t,Yd(:,13),'r')
    plot(t,Yd(:,14),'b')
    plot(t,Yd(:,15),'y')

hold off
legend('xd_3','yd_3','zd_3')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Hastighet, [m/s]')
grid on
box on

subplot(2,3,6)
hold on
    plot(t,Yd(:,16),'r')
    plot(t,Yd(:,17),'b')
    plot(t,Yd(:,18),'y')
```

```
hold off
legend('\phid_{3x}','\phid_{3y}','\phid_{3z}')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Vinkelhastighet, [rad/s]')
grid on
box on


%Akselerasjoner_____________________________________________

figure(3)

subplot(2,3,1)
hold on
    plot(t,Ydd(:,1),'r')
    plot(t,Ydd(:,2),'b')
    plot(t,Ydd(:,3),'y')

hold off
legend('xdd_1','ydd_1','zdd_1')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Akselerasjon, [m^2/s]')
grid on
box on

subplot(2,3,4)
hold on
    plot(t,Ydd(:,4),'r')
    plot(t,Ydd(:,5),'b')
    plot(t,Ydd(:,6),'y')

hold off
legend('\phidd_{1x}','\phidd_{1y}','\phidd_{1z}')
title('Krantårn')
xlabel('tid, [t]')
ylabel('Vinkelakselerasjon, [rad^2/s]')
grid on
box on

subplot(2,3,2)
hold on
    plot(t,Ydd(:,7),'r')
    plot(t,Ydd(:,8),'b')
    plot(t,Ydd(:,9),'y')

hold off
legend('xdd_2','ydd_2','zdd_2')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Akselerasjon, [m^2/s]')
grid on
box on

subplot(2,3,5)
hold on
    plot(t,Ydd(:,10),'r')
```

```
    plot(t,Ydd(:,11),'b')
    plot(t,Ydd(:,12),'y')

hold off
legend('\phidd_{2x}','\phidd_{2y}','\phidd_{2z}')
title('Kranarm 1')
xlabel('tid, [t]')
ylabel('Vinkelakselerasjon, [rad^2/s]')
grid on
box on


subplot(2,3,3)
hold on
    plot(t,Ydd(:,13),'r')
    plot(t,Ydd(:,14),'b')
    plot(t,Ydd(:,15),'y')

hold off
legend('xdd_3','ydd_3','zdd_3')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Akselerasjon, [m^2/s]')
grid on
box on

subplot(2,3,6)
hold on
    plot(t,Ydd(:,16),'r')
    plot(t,Ydd(:,17),'b')
    plot(t,Ydd(:,18),'y')

hold off
legend('\phidd_{3x}','\phidd_{3y}','\phidd_{3z}')
title('Kranarm 2')
xlabel('tid, [t]')
ylabel('Vinkelakselerasjon, [rad^2/s]')
grid on
box on

% %
Reaksjonskrefter============================================================

figure(4)

subplot(3,1,1)
hold on
plot(t,YA(:,1),'r')
plot(t,YA(:,2),'b')
plot(t,YA(:,3),'y')

hold off
xlabel('tid, [t]')
ylabel('Krefter')
legend('F_x','F_y','F_z');
title('Reaksjonskrefter binding A')
grid on
box on
```

```
subplot(3,1,2)
hold on
plot(t,YB(:,1),'r')
plot(t,YB(:,2),'b')
plot(t,YB(:,3),'y')

hold off
xlabel('tid, [t]')
ylabel('Krefter')
legend('F_x','F_y','F_z');
title('Reaksjonskrefter binding B')
grid on
box on

subplot(3,1,3)
hold on
plot(t,YC(:,1),'r')
plot(t,YC(:,2),'b')
plot(t,YC(:,3),'y')

hold off
xlabel('tid, [t]')
ylabel('Krefter')
legend('F_x','F_y','F_z');
title('Reaksjonskrefter binding C')
grid on
box on

for i=1:N

%3D animasjon=================================================================
    cla



r1_plot=Y(i,1:3)';    %x1,y1,z1
r2_plot=Y(i,7:9)';    %x2,y2,z2
r3_plot=Y(i,13:15)';  %x3,y3,z3

v1_plot = Y(i,4:6)';  %w1 (inneholdene phi1, theta1 og tau1)
v2_plot = Y(i,10:12)';%w2 (inneholdene phi2, theta2 og tau2)
v3_plot = Y(i,16:18)';%w2 (inneholdene phi3, theta3 og tau3)

A1 = A(v1_plot); %transformasjonsmatrise for v1
A2 = A(v2_plot); %transformasjonsmatrise for v2
A3 = A(v3_plot); %transformasjonsmatrise for v3

%krantårn
krant=[[b1/2 -D1/2 -h/2]',[b1/2 D1/2 -h/2]',[-b1/2 D1/2 -h/2]',...
    [-b1/2 -D1/2 -h/2]',[b1/2 -D1/2 -h/2]',...
[b1/2 -D1/2 h/2]',[b1/2 D1/2 h/2]',[b1/2 D1/2 -h/2]',[b1/2 D1/2 h/2]',...
[-b1/2 D1/2 h/2]',[-b1/2 D1/2 -h/2]',[-b1/2 D1/2 h/2]',[-b1/2 -D1/2
h/2]',...
[-b1/2 -D1/2 -h/2]',[-b1/2 -D1/2 h/2]',[b1/2 -D1/2 h/2]'];

%Kranarm1
kranarm1=[[b1/2 s2B(2,1) s2B(3,1)-D1/2]',[b1/2 s2C(2,1) s2C(3,1)-D1/2]',...
    [-b1/2 s2C(2,1) s2C(3,1)-D1/2]',[-b1/2 s2B(2,1) s2B(3,1)-D1/2]',...
    [b1/2 s2B(2,1) s2B(3,1)-D1/2]',...
```

```
[b1/2 s2B(2,1) s2B(3,1)+D1/2]',[b1/2 s2C(2,1) s2C(3,1)+D1/2]',...
[b1/2 s2C(2,1) s2C(3,1)-D1/2]',[b1/2 s2C(2,1) s2C(3,1)+D1/2]',...
[-b1/2 s2C(2,1) s2C(3,1)+D1/2]',[-b1/2 s2C(2,1) s2C(3,1)-D1/2]',...
[-b1/2 s2C(2,1) s2C(3,1)+D1/2]',[-b1/2 s2B(2,1) s2B(3,1)+D1/2]',...
[-b1/2 s2B(2,1) s2B(3,1)-D1/2]',[-b1/2 s2B(2,1) s2B(3,1)+D1/2]',...
[b1/2 s2B(2,1) s2B(3,1)+D1/2]'];

kranarm2=[[b1/2 s3C(2,1) (s3C(3,1)-D1/2)]',...
    [b1/2 s3G(2,1) (s3C(3,1)-D1/2)]',[b1/2 s3H(2,1) -
s3G(3,1)+s3H(3,1)]',...
    [-b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [-b1/2 s3G(2,1) (s3C(3,1)-D1/2)]',[-b1/2 s3C(2,1) (s3C(3,1)-D1/2)]',...
    [-b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',...
    [-b1/2 s3G(2,1)+D1/2 (s3C(3,1)+D1/2)]',...
    [-b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [-b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [-b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [b1/2 s3H(2,1) -s3G(3,1)+s3H(3,1)]',...
    [b1/2 s3H(2,1)+D1/2 -s3G(3,1)+s3H(3,1)+D1/2]',...
    [b1/2 s3G(2,1)+D1/2 (s3C(3,1)+D1/2)]',...
    [b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',...
    [-b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',...
    [b1/2 s3C(2,1) (s3C(3,1)+D1/2)]',...
    [b1/2 s3C(2,1) (s3C(3,1)-D1/2)]',...
    [-b1/2 s3C(2,1) (s3C(3,1)-D1/2)]'];

plot_krant = A1*krant;
plot_kranarm1 = A2*kranarm1;
plot_kranarm2 = A3*kranarm2;

plot_krant(1,:)=r1_plot(1,1) + plot_krant(1,:);
plot_krant(2,:)=r1_plot(2,1) + plot_krant(2,:);
plot_krant(3,:)=r1_plot(3,1) + plot_krant(3,:);

plot_kranarm1(1,:)=r2_plot(1,1) + plot_kranarm1(1,:);
plot_kranarm1(2,:)=r2_plot(2,1) + plot_kranarm1(2,:);
plot_kranarm1(3,:)=r2_plot(3,1) + plot_kranarm1(3,:);

plot_kranarm2(1,:)=r3_plot(1,1) + plot_kranarm2(1,:);
plot_kranarm2(2,:)=r3_plot(2,1) + plot_kranarm2(2,:);
plot_kranarm2(3,:)=r3_plot(3,1) + plot_kranarm2(3,:);

figure(8)

hold on
plot3(plot_krant(1,:),plot_krant(2,:),plot_krant(3,:),'b','linewidth',3)
plot3(plot_kranarm1(1,:),plot_kranarm1(2,:),plot_kranarm1(3,:),'r','linewid
th',3)
plot3(plot_kranarm2(1,:),plot_kranarm2(2,:),plot_kranarm2(3,:),'g','linewid
th',3)

view(3)
grid on
axis([-50 50 -50 50 -50 50])
hold off

if i==0.02
    pause
```

```
end
    pause(0.001)
end

e=1;



%Phi ========================================================
function fi=phi(q,V1,V2,V3,l_1_init,l_2_init,l_syl1,l_syl2,x,t)

%Henter konstanter=========================================================
k=konstanter;

h = k(1,13);      %Høyde krantårn (z-retning)
b1 = k(2,13);       %bredde (y-retning)
D1 = k(3,13);       %dybde (x-retning)

L1 = k(1,14);       %Lengde kranarm 1
L2 = k(2,14);       %Lengde kranarm 2

s1A = k(1:3,1);     %lengde fra lokal koordinat 1 til joint A
s1B = k(1:3,2);     %lengde fra lokal koordinat 1 til joint B
s1D = k(1:3,3);     %lengde fra lokal koordinat 1 til joint D
s1D(2,1) = s1D(2,1)*x(3);  %endring i y
s1D(3,1) = s1D(3,1)*x(4);  %endring i z
s2B = k(1:3,4)*x(1);     %lengde fra lokal koordinat 2 til joint B
s2C = k(1:3,5)*x(1);     %lengde fra lokal koordinat 2 til joint C
s2E = k(1:3,6)*x(1);     %lengde fra lokal koordinat 2 til joint E
s2E(2,1) = s2E(2,1)*x(5);    %endring i y
s2E(3,1) = s2E(3,1)*x(6);    %endring i z
s2F = k(1:3,7)*x(1);     %lengde fra lokal koordinat 2 til joint F
s2F(2,1) = s2F(2,1)*x(7);    %endring i y
s2F(3,1) = s2F(3,1)*x(8);    %endring i z
s3C = k(1:3,8)*x(2);     %lengde fra lokal koordinat 3 til joint C
s3G = k(1:3,9)*x(2);     %lengde fra lokal koordinat 3 til joint G
s3G(2,1) = s3G(2,1)*x(9);   %endring i y
s3G(3,1) = s3G(3,1)*x(10);   %endring i z
s3H = k(1:3,10)*x(2);    %lengde fra lokal koordinat 3 til joint H


m1 = k(1,17);       %masse krantårn
m2 = k(2,17);       %masse kranarm1
m3 = k(3,17);       %masse kranarm2

M1 = k(1:3,27:29);
M2 = k(1:3,30:32);
M3 = k(1:3,33:35);

J1 = k(1:3,18:20);
J2 = k(1:3,21:23);
J3 = k(1:3,24:26);

l1 = l_1_init;
l2 = l_2_init;

v_1 = k(2,15);      %hastighet sylinder 1
v_2 = k(3,15);      %hastighet sylinder 2
```

```
a_1 = k(2,16);      %akselerasjon sylinder 1
a_2 = k(3,16);      %akselerasjon sylinder 2

g = k(3,36);        %gravitasjonskraft


% Areal sylinderstempel positiv og negativ retning + rotdriver==========0
syl1_areal_pos = k(1,39);
syl1_areal_neg = k(2,39);
syl2_areal_pos = k(1,40);
syl2_areal_neg = k(2,40);
% rot_areal   = k(3,39);
displacement = k(3,40);    %  m^3/rotasjon

%================================================================
r1=q(1:3,1);    %x1,y1,z1
r2=q(7:9,1);    %x2,y2,z2
r3=q(13:15,1);  %x3,y3,z3

v1 = q(4:6,1);  %w1 (inneholdene phi1, theta1 og tau1)
v2 = q(10:12,1);%w2 (inneholdene phi2, theta2 og tau2)
v3 = q(16:18,1);%w3 (inneholdene phi3, theta3 og tau3)

A1 = A(v1); %transformasjonsmatrise for v1
A2 = A(v2); %transformasjonsmatrise for v2
A3 = A(v3); %transformasjonsmatrise for v3

u_x=[1 0 0]';
u_y=[0 1 0]';
u_z=[0 0 1]';

d1 = r1 + A1*s1D - A2*s2E - r2;   %lengde longitudinal driver 1
d2 = r2 + A2*s2F - A3*s3G - r3;   %lengde longitudinal driver 2

% Drivefunksjon Sylinder 1=======================================

        if l_syl1 >= 0
        lt1=(Q_syl1_pos(t,V1,x)/syl1_areal_pos) + l_1_init;
        else
        lt1=(-Q_syl1_pos(t,V1,x)/syl1_areal_neg) + l_1_init;
        end


        %Drivefunksjon Sylinder 2=================================
        if l_syl2 >= 0
        lt2 = (Q_syl2_pos(t,V2,x)/syl2_areal_pos) + l_2_init;
        else
        lt2 = (-Q_syl2_pos(t,V2,x)/syl2_areal_neg) + l_2_init;
        end



        %Drivefunksjon rotasjon===================================
        lt3 = (Q_rot_pos(t,V3,x)/displacement);


fi(1:18,1) = 0;
```

```
fi(1:3,1) = r1 + A1*s1A;
fi(4,1) = u_x'*A1*u_z;
fi(5,1) = u_y'*A1*u_z;
fi(6:8,1) = r1 + A1*s1B - A2*s2B - r2;
fi(9,1) = (A1*u_x)'*(A2*u_z);
fi(10,1) = (A1*u_x)'*(A2*u_y);
fi(11:13,1) = r2 + A2*s2C - r3 - A3*s3C;
fi(14,1) = (A2*u_x)'*(A3*u_z);
fi(15,1) = (A2*u_x)'*(A3*u_y);
fi(16,1) = v1(3,1) - lt3;
fi(17,1) = (d1'*d1) - lt1^2;
fi(18,1) = (d2'*d2) - lt2^2;


if (norm(d1)<0.4)
    hjelp = 1;
    pause
end

if (norm(d2)<0.4)
    hjelp = 2;
    pause
end
```

Vedlegg 3: Optimering av kjøretid, Matlab

```
%Optimering av kjørehastighet=============================================
clear all
close all
clc

k=konstanter;

s1D = k(1:3,3);
s2E = k(1:3,6);
s2F = k(1:3,7);
s3G = k(1:3,9);

x0 = [20, 0, 0.4, 15, 0, 0.4, 15, 0, 0.4, 15, s1D(2,1), s1D(3,1), s2E(2,1),
s2E(3,1), s2F(2,1), s2F(3,1), s3G(2,1), s3G(3,1)]';

A = [-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     -1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     -1,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0
     -1,0,0,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0
     -1,0,0,0,0,2,1,0,0,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0
     -1,0,0,0,0,0,0,1,2,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
     0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0
     -1,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,0
     0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
     0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0
     0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
     0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1];

b = [-10;inf;0;0;-0.2;3;0;0;0;0;-0.2;3;0;0;0;0;-0.2;3;0;0;s1D(2,1)+1;...
     -s1D(2,1)+1;s1D(3,1)+1;-s1D(3,1)+1;s2E(2,1)+1;-s2E(2,1)+1;...
     s2E(3,1)+1;s2E(3,1)+1;s2F(2,1)+1;-s2F(2,1)+1;s2F(3,1)+1;...
     -s2F(3,1)+1;s3G(2,1)+1;-s3G(2,1)+1;s3G(3,1)+1;s3G(3,1)+1];
```

```
f = @(x)OptimQ_max(x);
ff = @(x)main_script(x);

[xopt,val,exitflag,output] = fmincon(ff,x0,A,b,[],[],[],[],f);

anim = resultat(xopt);



%constraints=======================================================

function [c,ceq]=OptimQ_max(x)

global l_SYLINDER1A l_SYLINDER2A l_SYLINDER1B l_SYLINDER2B...
    VINKEL2_A VINKEL3_A VINKEL2_B VINKEL3_B HPU_max

c = [l_SYLINDER1A-9.05;l_SYLINDER2A-8.24;-l_SYLINDER1B+5.075;...
    -l_SYLINDER2B+4.67; -l_SYLINDER1A+5.075;-l_SYLINDER2A+4.67;...
    l_SYLINDER1B-9.05;l_SYLINDER2B-8.24;-VINKEL2_A; VINKEL2_A-(pi/2);...
    -VINKEL2_B; VINKEL2_B-(pi/2); -VINKEL3_A; VINKEL3_A-pi;...
    -VINKEL3_B; VINKEL3_B-pi;HPU_max-1];

ceq = [];

HPU_max-1
end
```

For detaljer om øvrige funksjoner henvises det til vedlegg 2 da disse er like for de to optimeringene.