

Modellering av roterende legeme

Raymond Bakken

Forsvarets forskningsinstitutt (FFI)

8. september 2009

FFI-rapport 2009/01503

102003

P: ISBN 978-82-464-1774-5

E: ISBN 978-82-464-1775-2

Godkjent av

Halvor Ajer

Prosjektleder

Johnny Bardal

Avdelingssjef

Sammendrag

Denne oppgaven gikk ut på å lage en matematisk modell av et 155 mm spinnstabilisert prosjektil som har mulighet til å korrigere sin egen bane. For å få til en banekorreksjon er det tenkt at man kan bytte ut brannrøret med et banekorrigerende brannrør. Dette brannrøret er festet på prosjektilkroppen via to kulelager slik at brannrøret og prosjektilkroppen kan ha forskjellig spinn. Det har blitt utledet en matematisk modell som beskriver prosjektilkroppens bevegelser i 6 frihetsgrader. Denne modellen tilsvarende et vanlig prosjektil og ble validert mot skytedata [8] med akseptable resultater. Så ble den testet med krefter på brannrøret som representerer styrepådrag for å se når granaten ble ustabil, slik at det kunne bli brukt for å finne en grei finnestørrelse som ikke kommer i konflikt med stabiliteten. Prosjektilet ser ut til å tåle rundt 140 N normalt på lengdeaksen. Denne matematiske modellen (med 6 frihetsgrader) ble så modifisert for å ta med spinn til det banekorrigerende brannrøret og beskriver nå prosjektillets 7 frihetsgrader. Det banekorrigerende brannrøret er utstyrt med to sett finner som er hentet fra [1]. Et finnesett tar seg av rullstabiliseringen av korreksjonsbrannrøret mens det andre finnesettet brukes til å korrigere banen til prosjektillet. Det er gjort en del tester på størrelsen til disse finnene for å finne toleransegrensen for hvor store disse finnene kan være før prosjektillet blir ustabil. Simuleringsresultater viser at de finnene som er designet i [1] ser ut til å være godt innenfor toleransegrensen for hva prosjektillet aksepterer før det blir ustabil.

Det er ikke hensiktsmessig å legge opp til å regulere banen til prosjektillet fra starten av. Man antar at skytset er korrekt innrettet og at man regulerer inn eventuelle avdrift underveis. Hvis reguleringen av banen starter etter 10 sekunder, viser simuleringer at man har en korreksjonsevne i sideretning på over 1000 meter og en korreksjonsevne på over 500 meter i lengderetning. Disse korreksjonsevnene vil avta jo senere man begynner å regulere. Disse simuleringene var gjort med en konstant momentpåvirkning mellom korreksjonsbrannrøret og prosjektilkroppen. I virkeligheten vil man ha en varierende momentpåvirkning mellom korreksjonsbrannrøret og prosjektilkroppen via kulelageropphenget og eventuelt en generator som lager strøm til elektronikken av rotasjonsforskjellen mellom korreksjonsbrannrøret og prosjektilkroppen. Det er vanskelig å få gode simuleringsresultater med denne momentpåvirkningen uten et ordentlig reguleringssystem, og det ble derfor besluttet å simulere med et fast moment mellom korreksjonsbrannrøret og prosjektilkroppen.

Rapporten inneholder et forslag for å beregne kulelagermomentet og eventuelt generatormoment, og det anbefales at disse likningene blir testet og eventuelt implementert når man har et reguleringssystem, slik at man får en så komplett modell som mulig.

English summary

The objective of this work was to make a mathematical model of a 155 mm spin-stabilised projectile that has the possibility to correct its own trajectory. To obtain course correction the idea is to change the fuse with a course-correction fuse (CCF). This CCF is connected to the shell through two roll bearings. In this way, the CCF and the shell can have different spin velocities. A mathematical model has been developed that describe the projectile's movements in six degrees of freedom. This model corresponds to a regular projectile and was validated against a firing table [8] with acceptable results. Then it was tested with forces on the CCF to test when the shell became unstable. The result was going to be used to find an ok size on the fins. The projectile seems to handle around 140 N perpendicular to its length axis, on the tip of its CCF. This model was then modified to include the spin of the CCF, and describes now the projectile's seven degrees of freedom. The CCF is equipped with two sets of fins from [1]. One set of fins controls the angular velocity of the CCF, while the other set controls the course correction of the projectile. It has been done a few tests with various fin sizes to find out how big they might be before the projectile gets unstable. The results show that the fins that are designed in [1] are well within the limits of how big they can be before the projectile gets unstable.

It is not practical to deliberately aim so that you must regulate the course from the beginning. Normally you aim at a target and you correct drift along as the projectile flies. If the guidance starts 10 seconds after start, simulations show a course-correction ability of over 1000 meter in deflection, and over 500 meters in range. This ability will be decreased with increasing delay in the startup of the guidance system. These simulations were done with a constant torque between the CCF and the shell. In the reality this torque will vary. It is difficult to get a good simulation result with this varying torque without a good representation of the guidance system, and it was therefore decided to run simulations with a fixed torque between the CCF and the shell.

This report includes an expression to calculate the torque from the roll bearings and an expression to calculate the torque from a generator. It is recommended that these equations be tested and eventually implemented when a regulation system is available. In this way the model will be as complete as possible.

Innhold

	Forord	8
1	Innledning	9
2	Notasjoner og koordinatsystemer	11
2.1	Koordinatsystemer	11
2.2	Vektornotasjoner	12
2.3	Matrisenotasjoner	14
2.4	Skalarnotasjon og forkorting	15
2.5	Koeffisientnotasjoner	17
3	Bevegelseslikningene	18
3.1	Noen definisjoner	18
3.2	Eulervinklene mellom aksesystemene	18
3.3	Likningene for translatoriske bevegelser	22
3.4	Likningene for rotasjonsbevegelser	24
3.5	Posisjonen	25
3.6	Likningssettet oppsummert	26
3.6.1	Eulervinklene mellom rammene	26
3.6.2	Likningene for translatoriske bevegelser	26
3.6.3	Likningene for rotasjonsbevegelser	27
3.6.4	Posisjonen	27
4	Krefter	28
4.1	Machtallet	29
4.2	Kraft- og momentkoeffisienter på objektets kropp	30
4.2.1	Gravitasjonskraften	30
4.2.2	Normal og aksialkraftkoeffisienter på grunn av vinden	30
4.2.3	Tippmomentkoeffisienter på grunn av vinden	31
4.2.4	Magnuskraftkoeffisienter	31
4.2.5	Magnusmomentkoeffisienter	32
4.2.6	Pitch -dempemomentkoeffisienter	33
4.2.7	Rullmomentet	33
4.3	Kraft- og momentkoeffisienter fra objektets finner, generator og kulelager	34
4.3.1	Styrefinnene	34
4.3.2	Rullstabiliseringsfinner	37
4.3.3	Moment fra generator	39

4.3.4	Moment fra kulelager	41
4.4	Oppsummering	45
5	Forklaring på simuleringsprogrammet og filer	46
5.1	Simuleringsprogrammet	46
5.1.1	Returparameter nummer 1	47
5.1.2	Returparameter nummer 2 og 3	48
5.1.3	Returparameter nummer 4	48
5.1.4	Inngangsparameter nummer 1	49
5.1.5	Inngangsparameter nummer 2	49
5.1.6	Inngangsparameter nummer 3	49
5.1.7	Inngangsparameter nummer 4	49
5.1.8	Inngangsparameter nummer 5	49
5.1.9	Inngangsparameter nummer 6	49
5.2	Plotteprogrammet	50
5.3	Filmprogrammet	51
6	Tester og resultater	53
6.1	Sammenlikning av treffpunkt med skytetabell	54
6.2	Test av finnestørrelser	55
6.2.1	Testoppsett	55
6.2.2	Resultater	57
6.3	Test av finnepådrag	57
6.4	Styrefinnestørrelsens innflytelse på banen.	58
6.5	Momentpåvirkningen mellom for- og bakseksjonens innvirkning på spinnets	60
6.6	Momentpåvirkningen mellom for- og bakseksjonens innvirkning på banen	61
6.7	For stor momentpåvirkningen mellom for- og bakseksjonen	62
6.8	Styrefinneutslags påvirkning på eulervinklens	63
6.8.1	Rullvinkelen til forseksjonen	64
6.8.2	Theta vinkelen	65
6.8.3	Psi-vinkelen	66
6.9	Styrefinneutslags påvirkning på vinkelhastighetene	66
6.9.1	Spinnets bak	67
6.9.2	Vinkelhastighetene q og r	68
6.10	Krefter på kroppen	69
6.10.1	Krefter langs x-aksen representert i ikke-rullsystemet	70
6.10.2	Krefter langs y-aksen representert i ikke-rullsystemet	71
6.10.3	Krefter langs z-aksen representert i ikke-rullsystemet	72
6.11	Momentene på kroppen	72

6.11.1	Momentet om x-aksen	73
6.11.2	Moment om y-aksen	74
6.11.3	Moment om z-aksen	75
7	Konklusjon	76
	Referanser	77
	Appendix A Tabell over aerodynamiske koeffisienter	78
	Appendix B Dataprogram og filer	80
B.1	Skript	82
B.2	SimProsjekttilMFinner	83
B.3	CoefficientsMF	86
B.4	InfluenceMF	88
B.5	RK6dofMF	90
B.6	PlotResults	92
B.7	Film	101
B.8	HentEXCELData	102
B.9	Konfig_OEF3HB_UF	103
B.10	Prosjekttilbak	104
B.11	Prosjektilkropp	105
B.12	BrannrorUtenFinner	106
B.13	BrannrorMedFinner	108
B.14	Rot	111

Forord

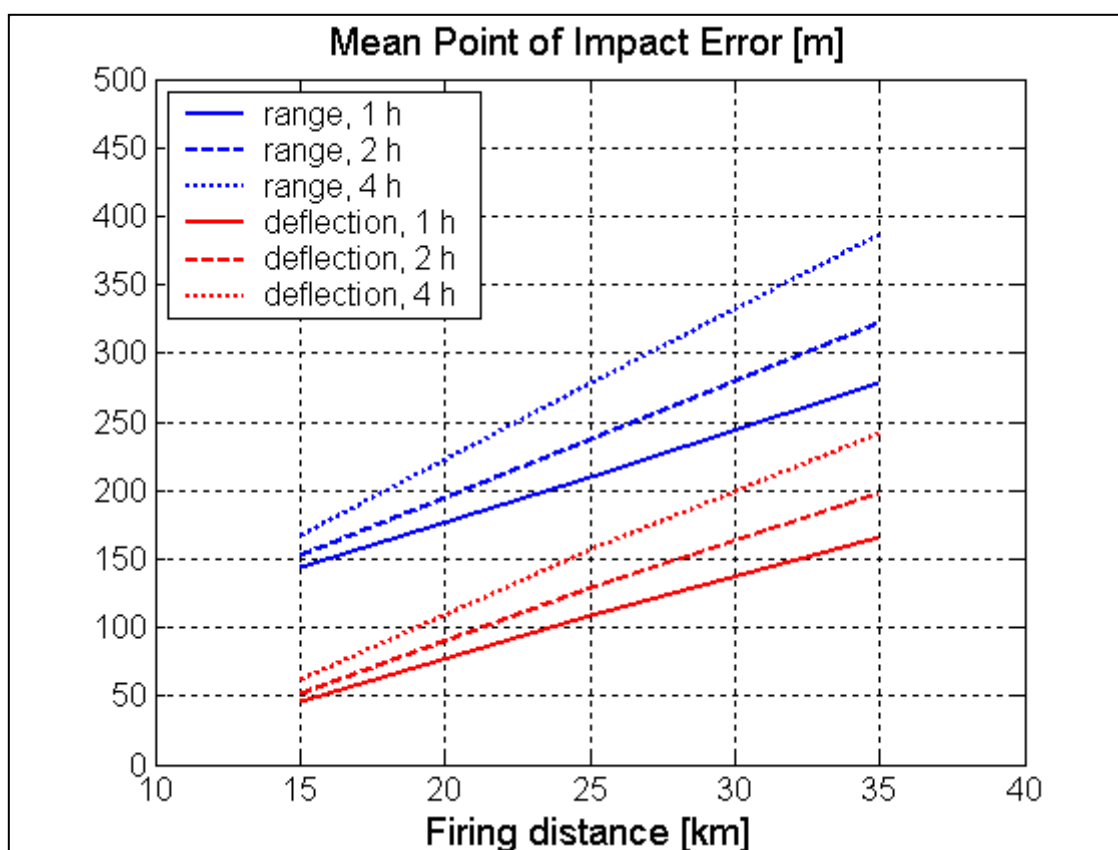
Jeg vil takke Oddvar Hallingstad, min eksterne hovedveileder og faglærer for gode råd og veiledning underveis. Han har vært min viktigste kontaktperson ved universitetet under hele utdannelsen. Han har hjulpet meg med nesten alt, fra det å sette sammen fagene jeg trenger og helt til å definere denne oppgaven. Faget hans UNIK4540 – Matematisk modellering av dynamiske system er et av de morsomste fagene jeg har tatt. Jeg vil også takke Øyvind Grandum, min eksterne veileder på FFI for gode råd og veiledning med oppgaven. Øyvind ser alltid ut til å ha et godt svar når jeg har kjørt meg skikkelig fast og kommer til han for hjelp. Jeg har et inntrykk av man skal lete lenge for å finne en dyktigere mann innen sitt felt. Grete Stavik-Døvle på fysisk institutt fortjener en takk for hjelpen med det praktiske sett fra universitetets side. Takk til Morten Nakjem min tidligere prosjektleder og John-Mikal Størdal forskningssjef på maritime systemer. Det er anbefalinger fra dem som gjorde det mulig for meg å ta mastergraden ved siden av jobben. Takk til Elin Bakke på FFI for hjelpen og tipsene jeg fikk da jeg skulle søke om tillatelse til å ta utdannelsen ved siden av jobben. Du hjalp meg mer en du aner Elin. Til slutt vil jeg takke min prosjektleder Halvor Ajer for den generelle forståelsen han har vist for at jeg har tatt utdannelsen ved siden av jobben. Mine veiledere var Oddvar Hallingstad, Øyvind Grandum og Torfinn Lindem.

Kjeller, 2009-05-19

Raymond Bakken

1 Innledning

Skyting med artilleri foregår i dag ved at man beregner asimut og elevasjon til løpet ved å bruke en modifisert massepunktmodell, for så å skyte. De feilene man får på treffpunktet bruker man så til å oppdatere modellen, slik at man får korrigert asimut og elevasjon til de neste skytinger. På denne måten kan man etter et viss antall skudd få ned virkningen av de systematiske feilene som gjør at middeltreffpunktet ikke er på siktepunktet. De tilfeldige feilene (spredningen) har man pr i dag ingen mulighet til å korrigere for. I massepunktmodellen kan man ta høyde for værdata. Men jo eldre værdata man har, jo større sannsynlighet har man for feil i middeltreffpunktet. Man har altså større sannsynlighet for at de systematiske feilene har økt. Figuren under er hentet fra [6] og viser et eksempel på gjennomsnittlig bomavstand som en funksjon av skyteavstand og tid siden værdata ble oppdatert.



Figur 1.1 Eksempel på middeltreffpunktfeil som en funksjon av skyteavstand og alder på meteorologisk data.

Det er ønskelig å se på muligheten til å korrigere banen til et prosjektil ved å bytte ut brannrøret med et banekorrigerende brannrør. Artilleriet vil sannsynligvis fremdeles være et verktøy man bruker mot områdemål og ikke presisjonsmål. Men, man kan tenke seg at feilområdet innsnevres og at innskytning blir unødvendig. Fordelene med banekorrigerende brannrør vil være mange. Man vil bruke færre skudd pr ildoppdrag for å oppnå ønsket effekt. Følgen av det er at man får mindre logistikkbelastning. Man kan også skyte nærmere egne styrker med mindre risiko. Man slipper innskytningsfasen, og dermed har man bedre mulighet for å bevare overraskelsesmomentet.

Formålet med denne oppgaven er å lage en matematisk modell av et prosjektil med et banekorrigerende brannrør. På brannrøret antas det at det blir montert noen finner som gjør det mulig å korrigere banen. Denne oppgaven vil også ta for seg en del tester med forskjellige størrelser på disse finnene, for å få en indikasjon på hvilke finnestørrelser som kan være mulig og hensiktsmessig å bruke.

2 Notasjoner og koordinatsystemer

I dette kapitlet blir det en gjennomgang av hvilket koordinatsystem som blir brukt under utledning av modellen. Det blir også forklart hvordan notasjonen til vektorer og matriser vil være.

Notasjonsbruken til vektorer og matriser er hentet fra [7]. Da de aerodynamiske koeffisientene har tilstrekkelig god forklaring i kapittel 4 vil de bare bli nevnt kort her. Vanlige skalarer blir bare satt opp i en tabell med forklarende tekst.

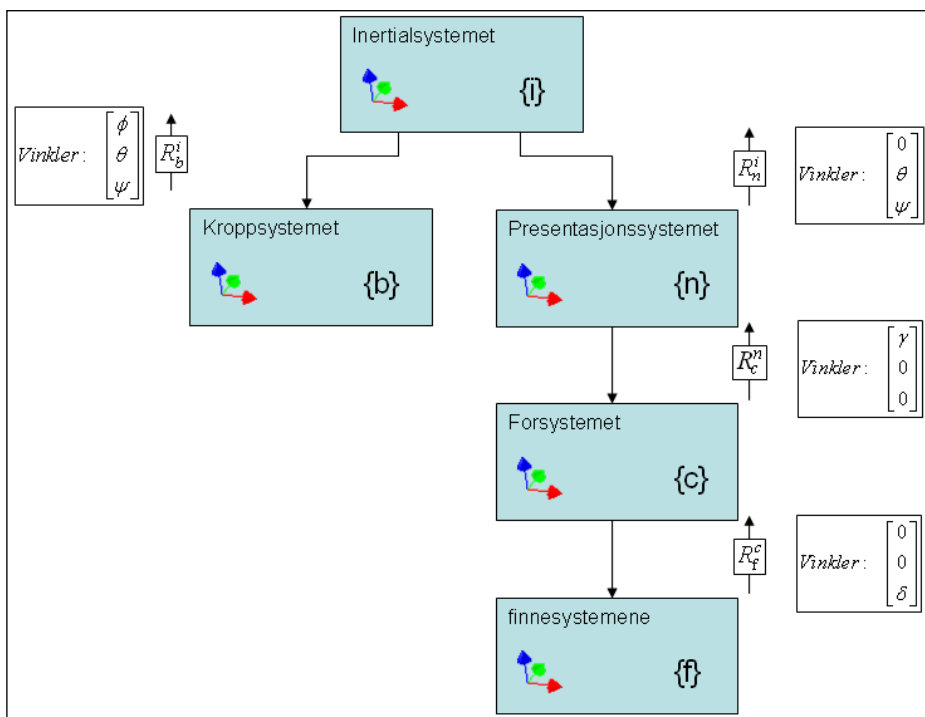
2.1 Koordinatsystemer

Systemet er bygd opp av 5 forskjellige koordinatsystemer som vist på figuren under. Det tas ikke hensyn til om jorden roterer, og det antas en flat jord. Inertialsystemet $\{i\}$, har x-aksen nordover, y-aksen østover og z-aksen peker nedover. Origo til $\{i\}$ -systemet befinner seg på startstedet. Det vil si at bortsett fra $\{f\}$ -ramma er det ikke er noen avstandsvektor mellom $\{i\}$ og noen andre rammer i startøyeblikket. Kroppssystemet $\{b\}$ ligger fast i bakdelen av objektet med x-aksen langs symmetriaksen til objektet og origo i massemidelpunktet til hele objektet. Med 0 vinkelutslag mellom $\{b\}$ og $\{i\}$ -rammene ligger aksene til $\{b\}$ og $\{i\}$ -rammene parallelt.

Presentasjonsrammen $\{n\}$ er helt lik $\{b\}$ -ramma bortsett fra at den ikke har noen rullbevegelse. Det vil si at for $\{n\}$ så er $\phi = 0$. Forsystemet $\{c\}$ ligger som $\{b\}$ bare at det har sin egen

rullbevegelse rundt x-aksen til $\{n\}$ -ramma. Egentlig består forsystemet av to rammer som er 90 grader på hverandre. Det ene er for styrefinnene og det andre er for rullstabiliseringsfinnene. Det er ikke behov for å gå innpå systemet til rullstabiliseringsfinnene da dette blir løst matematisk med antakelser fra styrefinnene. Det er allikevel nyttig å vite dette for forståelsen sin del.

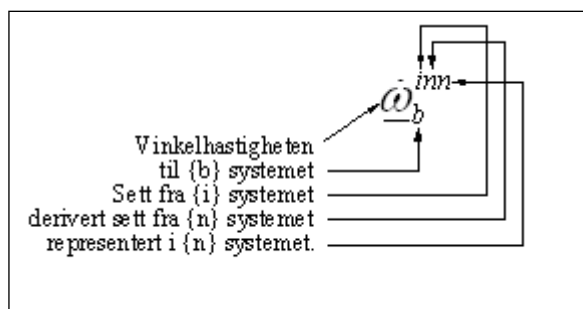
Finnesystemene består da også av to rammer. En ramme for styrefinnene og en for rullstabiliseringsfinnene. Påvirkning fra rullstabiliseringsfinnene vil som sagt bli løst matematisk med antagelser fra styrefinnene. Finnesystemene $\{f\}$ har origo i trykksenteret til finneparet. Dette trykksenteret ligger på akslingen til finneparet og på senterlinjen til prosjektilet. Styrefinnene har x-aksen framover og y-aksen ligger normalt på finneflaten. Hvis det ikke er noen eulervinkelutslag på noen av rammene, vil aksene til $\{f\}$ ligge parallelt med aksene til $\{i\}$. Styrefinnesystemet kan ha et eulervinkelutslag δ_1 om z-aksen til $\{c\}$.



Figur 2.1 Koordinatsystemhierarkiet.

2.2 Vektornotasjoner

En vektor skrives her som regel med liten bokstav. Unntakene som er gjort her, er på krefter, momenter og deres koeffisienter. De skrives med store bokstaver. Det brukes her algebraiske vektorer. Videre har vektorene noen hevede og senkede bokstavsymboler. Disse bokstavsymbolene forklarer stort sett noe om hvilken rolle de forskjellige rammene har på vektorene. Et eksempel på dette kan man se på figuren under.



Figur 2.2 Eksempel på vektornotasjon.

Man kan generelt si at hvert hevede bokstavsymbol bortsett fra den siste forteller hvor vektoren er derivert i fra. Den siste hevede bokstavsymbolen forteller hvor vektoren til slutt er representert. Symbolet $\underline{\omega}$ er et generelt symbol for en vinkelhastighetsvektor, mens her er det gitt av de små bokstavsymbolene hvilken vinkelhastighet det gjelder. I tilfellet med figuren over er det vinkelhastigheten til {b} -ramma sett fra {i} -ramma, men dette er en hastighet og det hevede bokstavsymbolen i representerer også at $\underline{\omega}$ inneholder en derivasjon sett fra {i}.

De vektornotasjonene som blir brukt er som følger:

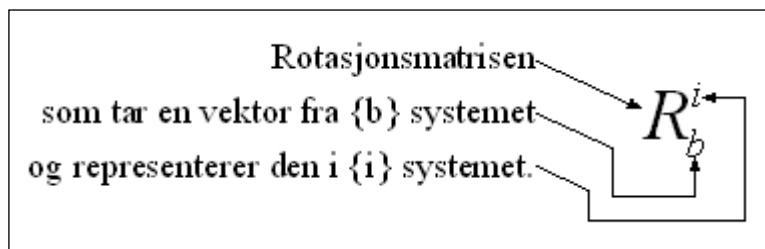
Symbol	Forklaring	komponenter
$\underline{\varphi}_{ib}$	Eulervinklene mellom {i} og {b}	$[\phi \ \theta \ \psi]^T$
$\underline{\varphi}_{in}$	Eulervinklene mellom {i} og {n}	$[0 \ \theta \ \psi]^T$
$\underline{\omega}_b^{ib}$	Vinkelhastigheten til {b} sett fra {i} representert i {b}	
$\underline{\omega}_b^{in}$	Vinkelhastigheten til {b} sett fra {i} representert i {n}	$[p \ q \ r]^T$
$\underline{\omega}_b^{ii}$	Vinkelhastigheten til {b} sett fra {i} representert i {i}	
$\underline{\omega}_n^{in}$	Vinkelhastigheten til {n} sett fra {i} representert i {n}	$[d \ q \ r]^T$
\underline{a}_F	En tilnærming til Sentripetalakselerasjonen	
\underline{a}_s	Sentripetalakselerasjon forårsaket av sentripetalhastigheten	
\underline{a}_b^{ii}	Akselerasjonen til {b} sett fra {i} representert i {i}	
\underline{a}_b^{in}	Akselerasjonen til {b} sett fra {i} representert i {n}	
\underline{c}	Avstandsvektor mellom {b} og {f} - rammene	$[c \ 0 \ 0]^T$
\underline{C}^{FF1A}	Aksialkraftkoeffisient for styrefinner	$[C_X^{FF1A} \ C_Y^{FF1A} \ C_Z^{FF1A}]^T$
\underline{C}^{FF1N}	Normalkraftkoeffisient for styrefinner	$[C_X^{FF1N} \ C_Y^{FF1N} \ C_Z^{FF1N}]^T$
\underline{C}^{FF2}	Total kraftkoeffisient for rullstabiliseringsfinner	
\underline{C}^{FF2A}	Aksialkraftkoeffisient for rullstabiliseringsfinner	$[C_X^{FF2A} \ C_Y^{FF2A} \ C_Z^{FF2A}]^T$
\underline{C}^{FF2N}	Normalkraftkoeffisient for rullstabiliseringsfinner	$[C_X^{FF2N} \ C_Y^{FF2N} \ C_Z^{FF2N}]^T$
\underline{C}^{MF1T}	Momentkoeffisient på grunn av styrefinner	$[C_X^{MF1T} \ C_Y^{MF1T} \ C_Z^{MF1T}]^T$
\underline{C}^{MF1T}	Momentkoeffisient på grunn av styrefinner	$[C_X^{MF1T} \ C_Y^{MF1T} \ C_Z^{MF1T}]^T$
\underline{C}^{MF2T}	Momentkoeffisient på grunn av rullstabiliseringsfinner	
\underline{e}_{xf}	Normalvektoren til aksialkraften til et finnepar	
\underline{F}_b^n	Kraften som påvirker {b} representert i {n}	$[F_X \ F_Y \ F_Z]^T$
\underline{F}_G^n	Gravitasjonskraften representert i {n} - ramma	
\underline{F}_F	Totale translatoriske krefter som virker på forseksjonen	$[F_{FX} \ F_{FY} \ F_{FZ}]^T$
\underline{F}_A	Totale translatoriske krefter som virker på bakseksjonen	$[F_{AX} \ F_{AY} \ F_{AZ}]^T$
\underline{F}_{YZ}^F	Kraften som virker på objektets y, z plan ved CP_F	
\underline{g}^i	Gravitasjonskraften i {i} - ramma.	$[0 \ 0 \ g]^T$
\underline{h}_b^{ii}	Spinnet om {b} sett fra {i} representert i {i}.	
\underline{n}_{fl}^n	Normalen til styrefinnene representert i {n} - ramma.	
\underline{n}_{fl}^f	Normalen til styrefinnene representert i {f} - ramma.	$[0 \ 1 \ 0]^T$
\underline{n}_b^{in}	Momentet om {b} sett fra {i} representert i {n}.	$[l \ m \ n]^T$
\underline{p}_{ib}^i	Avstandsvektor mellom {i} og {b} representert i {i}.	$[x \ y \ z]^T$

\underline{v}_F	Sentripetalhastighet på {f}, på grunn av vinkelhastigheten.	
\underline{v}_{F1}	Sentripetalhastighet på {f} i et diskret steg.	
\underline{v}_{F2}	Sentripetalhastighet på {f} i et neste diskrete steg.	
\underline{v}_{bw}^{in}	Hastighet til {b} sett fra {i} representert i {n}, med vind.	$[u_{bw} \ v_{bw} \ w_{bw}]^T$
\underline{v}_{fw}^{in}	Hastighet til {f} sett fra {i} representert i {n}, med vind.	$[u_{fw} \ v_{fw} \ w_{fw}]^T$
\underline{v}_w^{in}	Hastigheten til vinden sett fra {i} representert i {n}	
\underline{v}_b^{ii}	Hastighet til {b} sett fra {i} representert i {i}.	
\underline{v}_b^{in}	Hastighet til {b} sett fra {i} representert i {n}.	$[u \ v \ w]^T$

Tabell 2.1 Vektorbeskrivelse.

2.3 Matrisenotasjoner

Matrisene skrives her alltid med stor bokstav. Som vektorene har ofte også de noen hevede og senkede bokstavsymboler. Disse bokstavsymbolene forklarer noe om hvilken rolle de forskjellige rammene har på matrisene. Et eksempel på dette kan man se på figuren under.



Figur 2.3 Eksempel på matrisenotasjon.

Symbolet R er et generelt symbol for en rotasjonsmatrise mellom ortonormale rammer, mens her er det gitt av de små bokstavsymbolene hvilke rammer det gjelder. Eulervinklene mellom {b} og {i} -rammene kan man av figuren for koordinatsystemhierarkiet se heter $[\phi \ \theta \ \psi]^T$. Måten som R_b^i er oppbygd på vil da være slik:

$$\begin{aligned}
 R_b^i &= \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \\
 &= \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (0.1)
 \end{aligned}$$

Her står $c()$ for $\cos()$, og $s()$ for $\sin()$.

En vektor som skal krysses med en annen, kan bli representert som en skjevsymmetrisk matrise. Matrisen har symbol $S(\)$, og inni parentesen skrives den vektoren som skal krysses. Skal $\underline{\omega}_b^{in}$ krysses med \underline{p}_{ib}^i , kan man i stedet for å skrive $\underline{\omega}_b^{in} \times \underline{p}_{ib}^i$ bruke $S(\underline{\omega}_b^{in})\underline{p}_{ib}^i$. Noe mer utskrevet blir dette:

$$\begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (0.2)$$

Treghetstensoren skrives med symbolet T . Da kroppskoordinatsystemet er valgt med origo i massemidtpunktet til hele prosjektilet og med aksene slik at massefordelingen blir symmetrisk om x-aksen, vil T bli symmetrisk med elementene I_{xx} , I_{yy} og I_{zz} på diagonalen slik:

$$T = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (0.3)$$

2.4 Skalarnotasjon og forkorting

De skalarer som ikke er en del av en matrise eller en vektor, vil for ordens skyld bli ramset opp her.

Symbol	forklaring
α	Vinkel mellom rotasjonsakse og x-akse til {b}
Δt	Tiden til et diskret tidssteg
ε_1	Angrepsvinkel til styrefinnene
μ	Friksjonskoeffisient til kulelager
ρ	Lufttrykk ved objektet
ω	Vinkelhastighet
a	Translatorisk akselerasjon
$c(\)$	$\cos(\)$
CG	Massemiddelpunktet til objektet
CG_F	Massemiddelpunktet til forseksjonen
CP	Luftens trykksenter på objektet
CP_F	Trykksenteret til styrefinnene
CP_{magnus}	Trykksenteret for magnuskraften til objektet
D	Den tykkeste delen av objektet normalt på dens x-akse (referansediameter)
d	Hulldiameter i et kulelager
E	Symbol for energi

e	Koeffisient for lagerlikning
F_1	Den radielle kraften som påvirker kulelager 1
F_2	Den radielle kraften som påvirker kulelager 2
F_A	1) Total aksialkraft som påvirker kulelagrene
F_A	2) Summen av alle translatoriske krefter som virker på bakseksjonen
F_F	Summen av alle translatoriske krefter som virker på forseksjonen
F_{L1}	Den totale kraften på kulelager 1
F_{L2}	Den totale kraften på kulelager 2
F_R	Radiell lagerlast
g	Tyngdens akselerasjon
i	Elektrisk strøm
I	Treghetstensor
k	Fordelingsfaktor til aksialkraften på for og bakseksjonen
l_1	Avstand mellom kulelagrene, senter til senter
l_2	Avstand fra kulelager 2 til massesenteret på forseksjonen
M_G	Moment fra generator
M_L	Moment fra kulelagrene, benevning Nm
M_{LS}	Moment fra kulelagrene, benevning Nmm
M_{L1}	Moment fra kulelager 1
M_{L2}	Moment fra kulelager 2
M_T	Testmoment som virker mellom for og bakseksjonen
m	Masse
m_F	Massen til forseksjonen
P	Effekt som blir omgjort i øyeblikket. / Dynamisk lagerlast
r	Avstand fra momentakse til kraftpåvirkningspunkt
S	Referansearealet til objektet
$s()$	$\sin()$
T_W	Riflestigning i løpet (=20 kalibere pr omdreining)
t	Tid
$t()$	$\tan()$
u	Elektrisk spenning
v_0	Starthastigheten til {b} sett fra {i} representert i {n}. Langs {b} sin x-akse
v	En translatorisk hastighet
X	Skaleringsfaktor for kulelagerkraft
Y_1	Skaleringsfaktor for kulelagerkraft
Y_2	Skaleringsfaktor for kulelagerkraft

Tabell 2.2 Symbolbeskrivelse.

2.5 Koeffisientnotasjoner

Da alle de aerodynamiske koeffisientene har en forklarende overskrift og forklarende tekst, vil ikke det bli noen tabell over dem her. En liten forklaring på notasjonen vil allikevel bli gjort med et eksempel. Formelen for normalkraftkoeffisienten i y-retning på prosjektilet er:

$$C_Y^{FA} = -C_{na} \frac{v_{bw}}{|v_{bw}^{in}|} \quad (0.4)$$

Koeffisienten C_{na} er hentet fra tabellen i appendiks A og kalles der for CNa. Slik kan man finne alle de aerodynamiske koeffisientene på høyre side av likningene. Koeffisienten C_Y^{FA} er den koeffisienten som blir brukt videre i kraftberegningen. Her er notasjonen valgt slik at den senkede skriften forteller hvilken akse kreftene virker langs, eller momentet virker rundt. Den hevede skriften begynner alltid med M eller F som betyr moment eller kraft. Resten av bokstavene i den hevede skriften forteller noe om hvor koeffisienten hører hjemme. I koeffisienten C_Y^{FA} vil A si at denne koeffisienten er på grunn av angrepsvinkelen og i C_Y^{FM} vil M si at koeffisienten er på grunn av Magnuseffekten.

3 Bevegelseslikningene

I [2] ligger de ferdige likningene, men uten noen gode utledninger. For å få klarhet i hva disse likningene mente, var det nødvendig å utlede dem. Det er det som er gjort i dette kapitlet.

I utledningen av bevegelseslikningene forholder vi oss til 3 rammer. Origo til alle tre rammene er sammenfallende i startøyeblikket. Den ene er en jordfast ramme med x-aksen nordover, y-aksen østover og z-aksen nedover. Denne ramma kalles $\{i\}$. Den andre kalles $\{b\}$ og er fast i prosjektilet med origo i massemiddepunktet (også kalt CG). Positiv x-akse til $\{b\}$ er ut av nesen til objektet og i startøyeblikket er y-aksen horisontalt ut til høyre. Startvinkelen og høyrehåndsregelen vil da gi z-aksen. Den siste rammen kalles $\{n\}$ og er en ikke-rullende ramme. Denne rammen har origo i CG til objektet, x-aksen sammenfallende med x-aksen til $\{b\}$ -ramma, og y-aksen ligger alltid horisontalt ut til høyre for positiv x-akse. I denne ramma blir vektorene representert.

3.1 Noen definisjoner

Eulervinklene mellom $\{i\}$ og $\{b\}$ -rammene skrives $[\phi \quad \theta \quad \psi]^T$. Den translatoriske hastigheten mellom $\{i\}$ og $\{b\}$ -rammene på komponentform kalles $\underline{v}_b^i = [u \quad v \quad w]^T$ og er representert i $\{n\}$ -rammen. Vinkelhastigheten mellom $\{i\}$ og $\{b\}$ -rammene kalles $\underline{\omega}_b^i = [p \quad q \quad r]^T$ og er representert i $\{n\}$ -rammen. Vinklene mellom $\{i\}$ og $\{n\}$ -rammene skrives $[0 \quad \theta \quad \psi]^T$.

Vinkelhastigheten mellom $\{i\}$ og $\{n\}$ -rammene kalles $\underline{\omega}_n^i = [d \quad q \quad r]^T$ og er representert i $\{n\}$ -rammen. Rammen $\{n\}$ har ingen rullbevegelse, men er ellers lik $\{b\}$ -rammen.

Avstandsvektoren som oppstår mellom $\{i\}$ og $\{b\}$ -rammene kalles \underline{p}_{ib}^i og er representert i $\{i\}$ -rammen, Se figur 3.1.

3.2 Eulervinklene mellom aksesystemene

Stillingen til objektet er definert ved hjelp av eulervinklene $[\phi \quad \theta \quad \psi]^T$ der ϕ beskriver rotasjon rundt x-aksen, θ rotasjonen rundt y-aksen og ψ rotasjonen rundt z-aksen til $\{i\}$ -ramma.

Hvis alle eulervinklene er lik null radianer og $\underline{p}_{ib}^i = \underline{0}$ vil aksesystemene $\{i\}$, $\{n\}$ og $\{b\}$ være sammenfallende.

Likningen for den deriverte av retningscosinusmatrisa er:

$$\dot{R}_b^i = S(\underline{\omega}_b^i) R_b^i \quad (0.5)$$

Som hver enkelt utskrevet blir:

$$R_b^i = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (0.6)$$

$$\dot{R}_b^i = \begin{bmatrix} \frac{d(c(\psi)c(\theta))}{dt} & \frac{d(c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi))}{dt} & \frac{d(s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi))}{dt} \\ \frac{d(s(\psi)c(\theta))}{dt} & \frac{d(c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi))}{dt} & \frac{d(s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi))}{dt} \\ \frac{d(-s(\theta))}{dt} & \frac{d(c(\theta)s(\phi))}{dt} & \frac{d(c(\theta)c(\phi))}{dt} \end{bmatrix} \quad (0.7)$$

$$S(\underline{\omega}_b^{ii}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \quad (0.8)$$

Her er $\underline{\omega}_b^{ii} = [\omega_x \quad \omega_y \quad \omega_z]^T$. Skriver man ut $S(\underline{\omega}_b^{ii})R_b^i$ blir det:

$$\begin{bmatrix} \begin{pmatrix} -\omega_z s(\psi)c(\theta) - \\ \omega_y s(\theta) \end{pmatrix} & \begin{pmatrix} -\omega_z (c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi)) + \\ \omega_y c(\theta)s(\phi) \end{pmatrix} & \begin{pmatrix} -\omega_z (s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)) + \\ \omega_y c(\theta)c(\phi) \end{pmatrix} \\ \begin{pmatrix} \omega_z c(\psi)c(\theta) + \\ \omega_x s(\theta) \end{pmatrix} & \begin{pmatrix} \omega_z (c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)) - \\ \omega_x c(\theta)s(\phi) \end{pmatrix} & \begin{pmatrix} \omega_z (s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)) - \\ \omega_x c(\theta)c(\phi) \end{pmatrix} \\ \begin{pmatrix} -\omega_y c(\psi)c(\theta) + \\ \omega_x s(\psi)c(\theta) \end{pmatrix} & \begin{pmatrix} -\omega_y (c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)) + \\ \omega_x (c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi)) \end{pmatrix} & \begin{pmatrix} -\omega_y (s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)) + \\ \omega_x (s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)) \end{pmatrix} \end{bmatrix} \quad (0.9)$$

Dette er 9 likninger og 3 ukjente. Eulervinklene blir her bare utledet med 3 av ligningene.

Vi tar for oss likningene som tilsvarer plass [1,1], [3,1] og [3,2] i (3.3).

$$\frac{d(c(\psi)c(\theta))}{dt} = -\omega_z s(\psi)c(\theta) - \omega_y s(\theta) \quad (0.10)$$

$$\frac{d(-s(\theta))}{dt} = -\omega_y c(\psi)c(\theta) + \omega_x s(\psi)c(\theta) \quad (0.11)$$

$$\frac{d(c(\theta)s(\phi))}{dt} = -\omega_y (c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)) + \omega_x (c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi)) \quad (0.12)$$

Disse omskriver vi slik at vi får frem $[\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$

$$s(\psi)c(\theta)\dot{\psi} + c(\psi)s(\theta)\dot{\theta} = \omega_z s(\psi)c(\theta) + \omega_y s(\theta) \quad (0.13)$$

$$c(\theta)\dot{\theta} = \omega_y c(\psi)c(\theta) - \omega_x s(\psi)c(\theta) \quad (0.14)$$

$$-s(\theta)s(\phi)\dot{\theta} + c(\phi)c(\theta)\dot{\phi} = -\omega_y (c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)) + \omega_x (c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi)) \quad (0.15)$$

Så løses de med hensyn på $\begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$

$$\dot{\phi} = \frac{-c(\psi)s(\theta)s(\phi)}{c(\phi)c(\theta)}\omega_y + \frac{s(\psi)}{c(\theta)}\omega_y + \frac{c(\psi)}{c(\theta)}\omega_x + \frac{s(\psi)s(\theta)s(\phi)}{c(\phi)c(\theta)}\omega_x + \frac{s(\theta)s(\phi)}{c(\phi)c(\theta)}\dot{\theta} \quad (0.16)$$

$$\dot{\theta} = -s(\psi)\omega_x + c(\psi)\omega_y \quad (0.17)$$

$$\dot{\psi} = \omega_z + \frac{s(\theta)}{s(\psi)c(\theta)}\omega_y - \frac{c(\psi)s(\theta)}{s(\psi)c(\theta)}\dot{\theta} \quad (0.18)$$

Setter (3.13) inn i (3.12):

$$\begin{aligned} \dot{\phi} &= \frac{-c(\psi)s(\theta)s(\phi)}{c(\phi)c(\theta)}\omega_y + \frac{s(\psi)}{c(\theta)}\omega_y + \frac{s(\theta)s(\phi)c(\psi)}{c(\phi)c(\theta)}\omega_y \\ &= \left(\frac{c(\psi)}{c(\theta)} + \frac{s(\psi)s(\theta)s(\phi)}{c(\phi)c(\theta)} - \frac{s(\theta)s(\phi)s(\psi)}{c(\phi)c(\theta)} \right)\omega_x + \left(\frac{s(\psi)}{c(\theta)} + \frac{s(\theta)s(\phi)c(\psi)}{c(\phi)c(\theta)} - \frac{c(\psi)s(\theta)s(\phi)}{c(\phi)c(\theta)} \right)\omega_y \\ &= \left(\frac{c(\psi)}{c(\theta)} \right)\omega_x + \left(\frac{s(\psi)}{c(\theta)} \right)\omega_y \end{aligned} \quad (0.19)$$

Setter (3.13) inn i (3.14):

$$\begin{aligned} \dot{\psi} &= \omega_z + \frac{s(\theta)}{s(\psi)c(\theta)}\omega_y - \frac{c(\psi)s(\theta)}{s(\psi)c(\theta)}(-s(\psi)\omega_x + c(\psi)\omega_y) \\ &= \omega_z + \frac{s(\theta)}{s(\psi)c(\theta)}\omega_y + \frac{c(\psi)s(\theta)s(\psi)}{s(\psi)c(\theta)}\omega_x - \frac{c(\psi)s(\theta)c(\psi)}{s(\psi)c(\theta)}\omega_y \\ &= \frac{c(\psi)s(\theta)}{c(\theta)}\omega_x + \left(\frac{s(\theta)}{s(\psi)c(\theta)} - \frac{c(\psi)s(\theta)c(\psi)}{s(\psi)c(\theta)} \right)\omega_y + \omega_z \\ &= c(\psi)t(\theta)\omega_x + \frac{s(\theta) - c(\psi)s(\theta)c(\psi)}{s(\psi)c(\theta)}\omega_y + \omega_z \\ &= c(\psi)t(\theta)\omega_x + \frac{t(\theta) - c(\psi)t(\theta)c(\psi)}{s(\psi)}\omega_y + \omega_z \\ &= c(\psi)t(\theta)\omega_x + \frac{t(\theta)(1 - c(\psi)^2)}{s(\psi)}\omega_y + \omega_z \\ &= c(\psi)t(\theta)\omega_x + t(\theta)s(\psi)\omega_y + \omega_z \end{aligned} \quad (0.20)$$

Uttrykket ser nå slik ut:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{c(\psi)}{c(\theta)} & \frac{s(\psi)}{c(\theta)} & 0 \\ -s(\psi) & c(\psi) & 0 \\ c(\psi)t(\theta) & t(\theta)s(\psi) & 1 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (0.21)$$

Vi bruker $\underline{\omega}_b^{in}$ som er representert i {n}-systemet. Eulervinklene og $\underline{\omega}_b^{ii}$ er representert i {i}-systemet. Vi må derfor multiplisere med transformasjonsmatrisen:

$$\underline{\omega}_b^{ii} = R_n^i \underline{\omega}_b^{in} \quad (0.22)$$

Utskrevet blir likningen:

$$\begin{aligned} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} \frac{c(\psi)}{c(\theta)} & \frac{s(\psi)}{c(\theta)} & 0 \\ -s(\psi) & c(\psi) & 0 \\ c(\psi)t(\theta) & t(\theta)s(\psi) & 1 \end{bmatrix} \begin{bmatrix} c(\psi)c(\theta) & -s(\psi) & c(\psi)s(\theta) \\ s(\psi)c(\theta) & c(\psi) & s(\psi)s(\theta) \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & t(\theta) \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{c(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \end{aligned} \quad (0.23)$$

Når forseksjonen blir tatt med blir likningen slik:

$$\begin{bmatrix} \dot{\phi}_F \\ \dot{\phi}_A \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t(\theta) \\ 0 & 1 & 0 & t(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{c(\theta)} \end{bmatrix} \begin{bmatrix} p_F \\ p_A \\ q \\ r \end{bmatrix} \quad (0.24)$$

Representasjonsrammen {n} har ingen rullbevegelse så hvis denne likningen hadde vært utviklet med hensyn på den, ville $\dot{\phi} = \phi = 0$, og likningen for de vinklene ser slik ut:

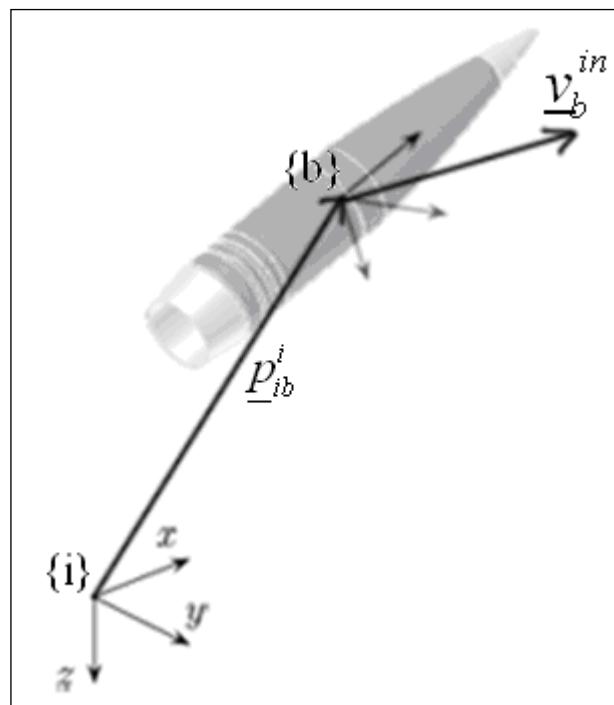
$$\begin{bmatrix} 0 \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t(\theta) \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{c(\theta)} \end{bmatrix} \begin{bmatrix} d \\ q \\ r \end{bmatrix} \quad (0.25)$$

Herfra ser vi at $d = -t(\theta)r$ og $[d \quad q \quad r]^T = \underline{\omega}_n^{in}$ på vektorform.

3.3 Likningene for translatoriske bevegelser

Likningene for translatorisk bevegelse er Newtons 2. lov for kraften på et legeme i tre dimensjoner. Det spiller ingen rolle hvor på legemet man påfører denne kraften, men hvis ikke kraftvektoren danner en linje som går gjennom massemidtpunktet vil man i tillegg få et moment på legemet.

På figuren under er det ikke tatt med $\{n\}$ -ramma. Den er allikevel fast i objektet som beskrevet over.



Figur 3.1 Objektet med noen koordinatsystemer.

Her er \underline{v}_b^{in} hastigheten til $\{b\}$ -ramma sett fra $\{i\}$ -ramma representert i $\{n\}$ -ramma og \underline{p}_{ib}^i avstandsvektoren fra $\{i\}$ -ramma til $\{b\}$ -ramma representert i $\{i\}$ -ramma. Sammenhengen mellom \underline{v}_b^{in} og \underline{p}_{ib}^i er:

$$\underline{v}_b^{in} = R_i^n \frac{d^i(\underline{p}_{ib}^i)}{dt} \quad (0.26)$$

Før vi finner akselerasjonen kan det være hensiktsmessig å først representere hastigheten i $\{i\}$ - ramma.

$$\underline{v}_b^{ii} = R_n^i \underline{v}_b^{in} \quad (0.27)$$

Vi kan nå derivere \underline{v}_b^{ii} fra $\{i\}$ -ramma

$$\begin{aligned} \underline{a}_b^{ii} &= \dot{\underline{v}}_b^{ii} = \dot{R}_n^i \underline{v}_b^{in} + R_n^i \dot{\underline{v}}_b^{inn} \\ &= S(\underline{\omega}_n^{ii}) R_n^i \underline{v}_b^{in} + R_n^i \dot{\underline{v}}_b^{inn} \\ &= R_n^i S(\underline{\omega}_n^{in}) \underline{v}_b^{in} + R_n^i \dot{\underline{v}}_b^{inn} \\ &= R_n^i \left(S(\underline{\omega}_n^{in}) \underline{v}_b^{in} + \dot{\underline{v}}_b^{inn} \right) \end{aligned} \quad (0.28)$$

Så kan vi representere det i $\{n\}$ -ramma

$$\underline{a}_b^{inn} = S(\underline{\omega}_n^{in}) \underline{v}_b^{in} + \dot{\underline{v}}_b^{inn} \quad (0.29)$$

Ordnet med hensyn på $\dot{\underline{v}}_b^{inn}$ og med objektets masse blir det

$$\dot{\underline{v}}_b^{inn} = \frac{F_b^n}{m} - S(\underline{\omega}_n^{in}) \underline{v}_b^{in} \quad (0.30)$$

Utskrevet blir dette:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \frac{1}{m} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -d \\ -q & d & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (0.31)$$

Av likning (3.21) ser vi at vi kan sette $d = -t(\theta)r$.

Utregnet blir likningen da

$$\dot{u} = \frac{F_x}{m} - qw + rv \quad (0.32)$$

$$\dot{v} = \frac{F_y}{m} - ru - t(\theta)rw \quad (0.33)$$

$$\dot{w} = \frac{F_z}{m} + t(\theta)rv + qu \quad (0.34)$$

3.4 Likningene for rotasjonsbevegelsene

Likningene for rotasjonsbevegelsen tar for seg momentet og spinnet til objektet.

Her trenger vi ikke å tenke på at den ene rammen flytter seg (translatorisk) i forhold til den andre, vi er mer opptatt av hvordan rammene roterer i forhold til hverandre. Vi starter med likningen for spinnet:

$$\underline{h}_b^{in} = T \underline{\omega}_b^{in} \quad (0.35)$$

Her er \underline{h}_b^{in} spinnet (eller bevegelsesmengdemomentet) om {b} sett fra {i} representert i {n}, T er treghetsmatrisa hvor treghetsmomentene I_{xx} , I_{yy} og I_{zz} befinner seg på diagonalen og treghetsproduktene I_{xy} o.s.v. er lik 0 og $\underline{\omega}_b^{in}$ er vinkelhastigheten til {b} sett fra {i} representert i {n}.

Hvis vi deriverer likning (3.31) fra {i} får vi:

$$\begin{aligned} \dot{\underline{h}}_b^{ii} &= \dot{R}_n^i T \underline{\omega}_b^{in} + R_n^i T \dot{\underline{\omega}}_b^{inn} \\ &= S(\underline{\omega}_n^{ii}) R_n^i T \underline{\omega}_b^{in} + R_n^i T \dot{\underline{\omega}}_b^{inn} \\ &= R_n^i S(\underline{\omega}_n^{in}) T \underline{\omega}_b^{in} + R_n^i T \dot{\underline{\omega}}_b^{inn} \end{aligned} \quad (0.36)$$

Representert i {n}-ramma:

$$\dot{\underline{h}}_b^{in} = S(\underline{\omega}_n^{in}) T \underline{\omega}_b^{in} + T \dot{\underline{\omega}}_b^{inn} \quad (0.37)$$

Momentet til {b} sett fra {i} representert i {n} er:

$$\underline{n}_b^n = \dot{\underline{h}}_b^{in} \quad (0.38)$$

Løst med hensyn på $\dot{\underline{\omega}}_b^{inn}$ får vi:

$$\dot{\underline{\omega}}_b^{inn} = T^{-1} \left(\underline{n}_b^n - S(\underline{\omega}_n^{in}) T \underline{\omega}_b^{in} \right) \quad (0.39)$$

Utskrevet blir det:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix}^{-1} \left(\begin{bmatrix} l \\ m \\ n \end{bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -d \\ -q & d & 0 \end{bmatrix} \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \quad (0.40)$$

Tar man med likning (3.21) og regner ut får man:

$$\dot{p} = \frac{l}{I_{XX}} \quad (0.41)$$

$$\dot{q} = \frac{m}{I_{YY}} - \frac{I_{XX}pr + I_{ZZ}t(\theta)r^2}{I_{YY}} \quad (0.42)$$

$$\dot{r} = \frac{n}{I_{ZZ}} + \frac{I_{XX}pq + I_{YY}t(\theta)qr}{I_{ZZ}} \quad (0.43)$$

Med roterende brannrør blir det:

$$\dot{p}_F = \frac{l_F}{I_{XX}^F} \quad (0.44)$$

$$\dot{p}_A = \frac{l_A}{I_{XX}^A} \quad (0.45)$$

$$\dot{q} = \frac{m}{I_{YY}} - \frac{(I_{XX}^A p_A + I_{XX}^F p_F)}{I_{YY}} r - \frac{I_{ZZ}}{I_{YY}} t(\theta) r^2 \quad (0.46)$$

$$\dot{r} = \frac{n}{I_{ZZ}} + \frac{(I_{XX}^A p_A + I_{XX}^F p_F)}{I_{ZZ}} q + \frac{I_{YY}}{I_{ZZ}} t(\theta) qr \quad (0.47)$$

Her er p_A vinkelhastigheten til bakseksjonen og p_F vinkelhastigheten til forseksjonen. På likning (3.42) og (3.43) er det $I_{XX}p$ som er splittet opp i $(I_{XX}^A p_A + I_{XX}^F p_F)$ og behandles hver for seg som et superposisjonsprinsipp. Her er I_{XX}^F treghetsmomentet til forseksjonen og I_{XX}^A treghetsmomentet til bakseksjonen rundt x aksen. I_{YY} og I_{ZZ} er de totale treghetsmomenta rundt de gjeldende akser. Det vil si at $I_{YY} = I_{YY}^F + I_{YY}^A$ og $I_{ZZ} = I_{ZZ}^F + I_{ZZ}^A$. Generelt kan vi si at $T = T_F + T_A$ og at T er diagonal med I_{XX} , I_{YY} og I_{ZZ} på diagonalen. For at dette skal gå, må origo til forseksjonen være sammenfallende med origo til bakseksjonen.

3.5 Posisjonen

Posisjonen finnes ved å integrere opp hastigheten over en brukt tid. Vi må selvsagt vite starthastigheten og startposisjonen for å finne sluttposisjonen. Her har man v_b^{in} som er

hastigheten til {b} -ramma sett fra {i} -ramma representert i {n} -ramma. Denne kan man gjerne representere som den deriverte av posisjonen sett fra {i} -ramma representert i {n} -ramma.

$$\underline{v}_b^{in} = R_i^n \underline{\dot{p}}_{ib}^{ii} \quad (0.48)$$

Som snudd med hensyn på $\underline{\dot{p}}_{ib}^{ii}$ blir

$$\underline{\dot{p}}_{ib}^{ii} = R_n^i \underline{v}_b^{in} \quad (0.49)$$

Eller utskrevet:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi) & c(\psi)s(\theta) \\ s(\psi)c(\theta) & c(\psi) & s(\psi)s(\theta) \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (0.50)$$

3.6 Likningssettet oppsummert

3.6.1 Eulervinklene mellom rammene

Eulervinklene mellom {b} -ramma og {i} -ramma.

$$\begin{bmatrix} \dot{\phi}_F \\ \dot{\phi}_A \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t(\theta) \\ 0 & 1 & 0 & t(\theta) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{c(\theta)} \end{bmatrix} \begin{bmatrix} p_F \\ p_A \\ q \\ r \end{bmatrix} \quad (0.51)$$

3.6.2 Likningene for translatoriske bevegelser

Hastigheten til {b} sett fra {i} representert i {n} på vektorform.

$$\underline{v}_b^{inn} = \frac{F_b^n}{m} - S(\underline{\omega}_n^{in}) \underline{v}_b^{in} \quad (0.52)$$

Utskrevet og utregnet:

$$\dot{u} = \frac{F_x}{m} - qw + rv \quad (0.53)$$

$$\dot{v} = \frac{F_y}{m} - ru - t(\theta)rw \quad (0.54)$$

$$\dot{w} = \frac{F_z}{m} + t(\theta)rv + qu \quad (0.55)$$

3.6.3 Likningene for rotasjonsbevegelsene

Rotasjonen til {b} sett fra {i} representert i {n} på vektorform:

$$\underline{\dot{\omega}}_b^{inn} = T^{-1} \left(\underline{n}^n - S(\omega_n^{in}) T \underline{\omega}_b^{in} \right) \quad (0.56)$$

Utskrevet, utregnet og med foreksjon:

$$\dot{p}_F = \frac{l_F}{I_{XX}^F} \quad (0.57)$$

$$\dot{p}_A = \frac{l_A}{I_{XX}^A} \quad (0.58)$$

$$\dot{q} = \frac{m}{I_{YY}} - \frac{(I_{XX}^A p_A + I_{XX}^F p_F)}{I_{YY}} r - \frac{I_{ZZ}}{I_{YY}} t(\theta) r^2 \quad (0.59)$$

$$\dot{r} = \frac{n}{I_{ZZ}} + \frac{(I_{XX}^A p_A + I_{XX}^F p_F)}{I_{ZZ}} q + \frac{I_{YY}}{I_{ZZ}} t(\theta) qr \quad (0.60)$$

3.6.4 Posisjonen

Posisjonen til {b} sett fra {i} representert i {i} på vektorform:

$$\underline{\dot{p}}_{ib}^i = R_n^i \underline{y}_b^{in} \quad (0.61)$$

Utskrevet:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi) & c(\psi)s(\theta) \\ s(\psi)c(\theta) & c(\psi) & s(\psi)s(\theta) \\ -s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (0.62)$$

4 Krefter

Utledningene i dette kapitlet er i veldig stor grad basert på arbeidsnotatet "Bevegelseslikninger for artillerigranat med styreseksjon" av Øyvind Grandum og Tareq A. Quassem. [2]

Aerodynamisk kraft avhenger av kvadratet av hastigheten til objektet i forhold til luften, lufttrykket, form og størrelse på objektet og blant annet luftens viskositet. Da et legeme kan ha mange forskjellige former vil den aerodynamiske kraften være svært forskjellig ved forskjellige typer legemer. Det er derfor for et objekt vanlig å lage en aerodynamisk koeffisient som inneholder alle de komplekse størrelsene i en enkel variabel. Det er klart at så langt objektet ikke er likesidet vil ikke koeffisienten for en side gjelde for en annen, og når det er sagt vil også koeffisientene for en side bli mer og mer ugyldig jo mer vi dreier objektet i forhold til vinden. Denne koeffisienten er vanligvis enten bestemt eksperimentelt eller ved et simuleringsprogram. En vanlig standardmåte å beskrive koeffisienten på er:

$$C_F = \frac{F}{\rho v^2 \frac{1}{2} S} \quad (0.63)$$

Hvor C_F er kraftkoeffisienten, F er den totale kraften som påvirker objektet, ρ er lufttrykket, v er hastigheten til objektet relativt til luften og S er et referanseareal. Av formelen kan man se at kraften er en funksjon av kvadratet av hastigheten og en lineær funksjon av lufttrykket og referansearealet. Referansearealet kan være et av flere forskjellige arealer. Hvis kraften på objektet er forårsaket av friksjonen mellom luften og objektets overflate, ville et naturlig valg være objektets overflateareal. (En kan lett tenke seg at et dobbelt så stort areal vil gi dobbel så stor kraft, og man har et areal som kan gjøre enkelte beregninger lette.)

Vi er fri til å velge hvilket areal vi vil. Her er referansearealet:

$$S = \frac{D^2 \pi}{4} \quad (0.64)$$

Her er D referansediameteren på prosjektilet. Det vil si diameteren på prosjektilet der prosjektilet er størst.

Et moment er et produktet av arm \times kraft, og for sammenlikning settes her opp kraft- og momentkoeffisientene snudd med hensyn på kraft og moment.

$$F = C_F \rho v^2 \frac{1}{2} S \quad (0.65)$$

$$M = C_M \rho v^2 \frac{1}{2} S \cdot D \quad (0.66)$$

Den aerodynamiske kraften som påvirker objektet, er forårsaket av luftens hastighet mot kroppen på objektet. Luftens hastighet mot kroppen på objektet er satt sammen av to forskjellige hastigheter, objektets hastighet i forhold til {i} og luftens hastighet i forhold til {i} slik:

$$\underline{v}_{bw}^{in} = \underline{v}_b^{in} - \underline{v}_w^{in} \quad (0.67)$$

Hvor \underline{v}_{bw}^{in} er objektets hastighet i forhold til vinden representert i {n}, \underline{v}_b^{in} er objektets hastighet i forhold til {i} representert i {n} og \underline{v}_w^{in} er vindens hastighet i forhold til {i} representert i {n}.

4.1 Machtallet

Aerodynamikken rundt et objekt forandrer seg en del i forhold til hvilket Mach-tall objektet har gjennom mediet (luften). Dette fører til at koeffisientene blir hentet ut av tabellen i Appendix A og interpolert i forhold til mach-tallet for å få nøyaktigst mulig kraft og moment. Formelen for Mach-tallet er:

$$M = \frac{v_s}{u} \quad (0.68)$$

Her er v_s er hastigheten til objektet og u er hastigheten til lyden i mediet.

Hastigheten til lyd i et medium er gitt ved:

$$u = \sqrt{\frac{C}{\rho}} = \sqrt{\frac{\gamma \cdot p}{\rho}} \quad (0.69)$$

Hvor C er en koeffisient for stivhet og ρ er tettheten og p er trykket.

γ er en ekspansjonsfaktor. Den er et forholdstall mellom spesifikk varmekapasitet til en gass ved konstant trykk til en gass ved et konstant volum. Den oppstår fordi en lydbølge danner en adiabatisk kompresjon, hvor kompresjonsvarmen ikke får tid til å unnsnippe trykkpulsene. Videre har vi den vanlige gasslikningen for en luft:

$$\frac{p}{\rho} = RT \quad (0.70)$$

R er den spesifikke gasskonstanten for luft, og T er temperaturen. Setter vi (4.8) og (4.7) inn i (4.6), blir formelen for mach-tallet:

$$M = \frac{v_s}{\sqrt{\gamma RT}} \quad (0.71)$$

$\sqrt{\gamma R}$ kan slås sammen til en konstant, og utregnet blir den 20.048. Men standard atmosfære (ICAO) forutsetter at T avtar lineært med høyden:

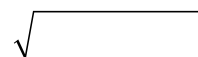
$$T = T_0 + a(h - h_0) \quad (0.72)$$

I "standard atmosfære" er $h_0 = 0$, $T_0 = 288.16K$ og $a = -0.0065 K/m$

Formelen for mach-tallet blir nå:

$$M = \frac{v_s}{20.048 \sqrt{T_0 + a(h - h_0)}} \quad (0.73)$$

Utleddningen av mach-tallet er hentet fra [3] og [9].



4.2 Kraft- og momentkoeffisienter på objektets kropp

Et moment er vanligvis gitt av $arm \times kraft$. I disse likningene får vi ikke oppgitt hvor CP (det aerodynamiske trykksenteret) ligger, men vi får momentet rett ut av koeffisienter i stedet.

4.2.1 Gravitasjonskraften

Gravitasjonskraften er en vektor langs z-aksen i {i} -ramma. I denne modellen skal alle krefter og momenter være representert i {n} -ramma. Formelen for gravitasjonskraften blir derfor:

$$\underline{F}_G^n = m \cdot R_i^n \underline{g}^i \quad (0.74)$$

Hvor m er massen til objektet, R_i^n er rotasjonsmatrisen fra {i} til {n} -ramma, og \underline{g}^i er gravitasjonsvektoren representert i {i} -ramma. Det blir her antatt at det ikke er nødvendig å dele opp gravitasjonskraften i en del for forseksjonen og en for bakseksjonen da gravitasjonskraften gir lik akselerasjon på både for og bakseksjonen.

4.2.2 Normal og aksialkraftkoeffisienter på grunn av vinden

$$C_X^{FA} = -C_{X0} - C_{X2} \frac{v_{bw}^2 + w_{bw}^2}{|\underline{v}_{bw}^{in}|^2} \quad (0.75)$$

$$C_Y^{FA} = -C_{na} \frac{v_{bw}}{|\underline{v}_{bw}^{in}|} \quad (0.76)$$

$$C_Z^{FA} = -C_{Na} \frac{w_{bw}}{|\underline{v}_{bw}^{in}|} \quad (0.77)$$

Normalkraftkoeffisientene C_Y^{FA} og C_Z^{FA} er koeffisienter for krefter langs prosjektillets y og z -retning. C_X^{FA} er aksialkraftkoeffisienten og virker langs objektets x -retning. Her vil en del av aksialkraften virke på forseksjonen. Hvor mye av den kraften som virker på forseksjonen er

usikker, men et anslag kan være på $C_{XF}^{FA} = C_X^{FA} 0.2$. Formelen for fordeling av aksialkraften på for- og bakseksjonen er i alle fall:

$$C_{XF}^{FA} = C_X^{FA} k \quad (0.78)$$

$$C_{XA}^{FA} = C_X^{FA} (1 - k) \quad (0.79)$$

Her er C_{XF}^{FA} en koeffisient for aksialkraftpåvirkning på forseksjonen, C_{XA}^{FA} er en koeffisient for aksialkraftpåvirkning på bakseksjonen og k er en skaleringsfaktor fra [0-1] som sier noe om fordelingen av aksialkraften.

Disse koeffisientene tar for seg kraften som skyldes innbyrdes bevegelse mellom prosjektil og lufta. Her er C_{Na} en normalkraftkoeffisient som er forårsaket av løftet som skapes av vinden normalt på objektets x -retning.

4.2.3 Tippmomentkoeffisienter på grunn av vinden

$$C_Y^{MA} = C_{ma} \frac{W_{bw}}{|\mathbf{v}_{bw}^{in}|} \quad (0.80)$$

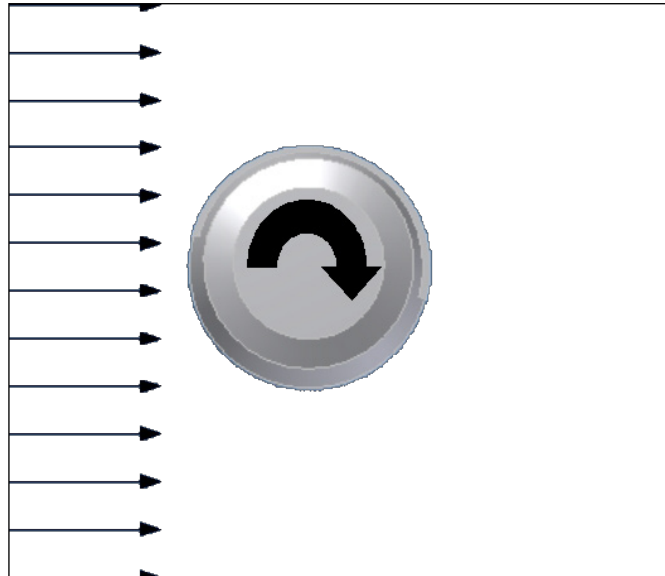
$$C_Z^{MA} = -C_{ma} \frac{v_{bw}}{|\mathbf{v}_{bw}^{in}|} \quad (0.81)$$

C_Y^{MA} og C_Z^{MA} er koeffisienter for momenter i prosjektillets y og z -retning.

C_{ma} er en momentkoeffisient som er forårsaket av at granatens trykksenter (CP) ikke sammenfaller med objektets gravitasjonssenter (CG).

4.2.4 Magnuskraftkoeffisienter

På figuren under er objektet vist bakfra. Pilene fra venstre illustrerer vinden, mens pilen bak på objektet illustrerer rotasjonsretningen til objektet.



Figur 4.1 Illustrasjon for Magnuskraften.

Man kan tenke seg at objektet drar luften med seg i rotasjonen og skaper en roterende vind rundt seg selv. Dette gjør at luften under objektet møter vinden, og luften over objektet har samme retning som vinden. Dette skaper en trykkeffekt på undersiden og en vakuumeffekt på oversiden som igjen skaper en kraft som peker i retning mot vakuumsiden. Her blir det da en kraft som peker oppover, Magnus-kraften. Denne kraftvektoren starter i CP_{magnus} , og så lenge CP_{magnus} og CG ikke er sammenfallende vil dette skape et moment, Magnus-momentet.

Magnuskraften virker bare i objektets y og z-retning. Koeffisientene C_Y^{FM} og C_Z^{FM} tar for seg kraftkomponentene for den.

$$C_Y^{FM} = C_{YPA} \left(\frac{p_A D}{2 |v_{bw}^{in}|} \right) \frac{w_{bw}}{|v_{bw}^{in}|} \quad (0.82)$$

$$C_Z^{FM} = -C_{YPA} \left(\frac{p_A D}{2 |v_{bw}^{in}|} \right) \frac{v_{bw}}{|v_{bw}^{in}|} \quad (0.83)$$

Koeffisienten C_{YPA} er Magnuskraftkoeffisienten fra appendiks A.

4.2.5 Magnusmomentkoeffisienter

Magnusmomentet virker bare om objektets y og z-akse. Koeffisientene C_Y^{MM} og C_Z^{MM} tar for seg momentkomponentene for det.

$$C_Y^{MM} = C_{MPA} \left(\frac{p_A D}{2 |\underline{v}_{bw}^{in}|} \right) \frac{v_{bw}}{|\underline{v}_{bw}^{in}|} \quad (0.84)$$

$$C_Z^{MM} = C_{MPA} \left(\frac{p_A D}{2 |\underline{v}_{bw}^{in}|} \right) \frac{w_{bw}}{|\underline{v}_{bw}^{in}|} \quad (0.85)$$

Koeffisienten C_{MPA} er Magnusmomentkoeffisienten. Ved bruk av høyrehåndsregelen her må man huske på at CP for magnuskraften vanligvis ligger bak CG, slik at $C_{MPA} < 0$.

4.2.6 Pitch -dempemomentkoeffisienter

Pitch-dempemomentkoeffisientene virker bare om objektets y og z-akse. Koeffisientene C_Y^{MQ} og C_Z^{MQ} tar for seg momentkomponentene for dem.

$$C_Y^{MQ} = C_{MQ} \left(\frac{qD}{2 |\underline{v}_{bw}^{in}|} \right) \quad (0.86)$$

$$C_Z^{MQ} = C_{MQ} \left(\frac{rD}{2 |\underline{v}_{bw}^{in}|} \right) \quad (0.87)$$

Koeffisienten C_{MQ} er pitch-dempemomentkoeffisienten. For å forstå pitch-dempemomentet kan man tenke seg at objektet befinner seg svevende i et rom uten gravitasjonsfelt. Rommet er fylt med gass, og objektet har en presesjonsbevegelse. Dette fører til at luften rundt objektet settes i bevegelse. Dette igjen fører til at objektet gir fra seg kinetisk energi, og presesjonen blir langsommere. Til slutt vil presesjonsbevegelsen opphøre. Momentet som luften i dette tilfellet utfører på prosjektilet, kalles pitch-dempemomentet.

4.2.7 Rullmomentet

Rullmomentet virker om objektets x-akse og på bakseksjonen. Koeffisienten C_X^{MR} tar for seg momentkomponenten for den.

$$C_X^{MR} = C_{LP}^A \left(\frac{p_A D}{2 |\underline{v}_{bw}^{in}|} \right) \quad (0.88)$$

Koeffisienten C_{LP}^A er rullmomentkoeffisienten. Forklaringen på rullmomentet blir ganske analogt med forklaringen på pitch-dempemomentet. Et prosjektil som har et spinn, vil dra luften med seg, noe som forårsaker at spinnet minker i hastighet. Det er rullmomentet som er årsaken til dette.

4.3 Kraft- og momentkoeffisienter fra objektets finner, generator og kulelager

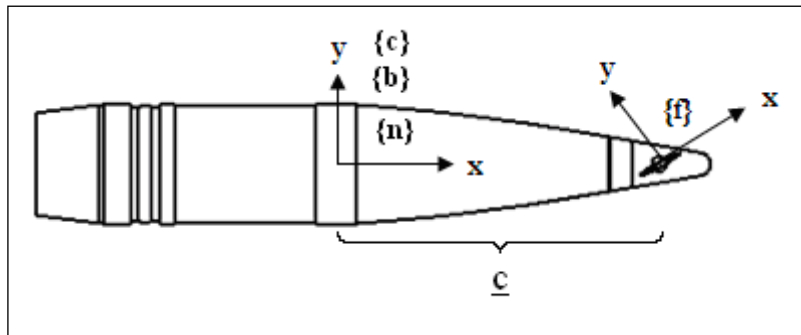
Kroppen til objektet består av to deler. Bakseksjonen og forseksjonen.

Bakseksjonen og forseksjonen har samme θ og ψ eulervinkler men forskjellige ϕ eulervinkler. Eulervinkelen γ betegner rullvinkelen til forseksjonen, og eulervinkelen ϕ betegner rullvinkelen til bakseksjonen. Her blir $\phi = 0$ fordi denne representasjonsrammen ikke skal ha noen rullbevegelse. I tillegg har finnene en angrepsvinkel δ_1 og δ_2 .

4.3.1 Styrefinnene

For utledning av kreftene kan det være greit å tenke seg at vi har to nye rammer til.

Ramma $\{f\}$, som er fast i finnene og $\{c\}$ -ramma, som er fast i forseksjonen. Ramma til forseksjonen $\{c\}$ har origo på samme plass som $\{b\}$ -ramma. $\{b\}$, $\{n\}$ og $\{c\}$ rammene har lik θ og ψ vinkler, men forskjellige ϕ vinkler (ϕ for bakseksjonen, γ for forseksjonen og 0 for representasjonsrammen). $\{f\}$ -ramma har origo på x-aksen til $\{b\}$ -ramma, i trykksenteret til styrefinnene. Se figuren under.



Figur 4.2 Objektet med styrefinner.

Når alle vinkler er 0 radianer, er aksene til $\{i\}$, $\{b\}$, $\{c\}$, $\{n\}$ og $\{f\}$ parallelle. Vi har nå laget oss et tre av rammer. Dette treet kan sees i Figur 2.1: Koordinatsystemhierarkiet.

4.3.1.1 Normalmoment- og normalkraftkoeffisientene til styrefinnene

Normalen til finnene er enhetsvektoren til $\{f\}$ i y-retning, $\underline{n}_{f1}^f = [0 \ 1 \ 0]^T$. Representert i $\{n\}$ -ramma blir det:

$$\underline{n}_{f1}^n = R_f^n \underline{n}_{f1}^f \quad (0.89)$$

R_f^n bør utledes, og det blir gjort her:

$$R_f^n = R_c^n R_f^c = R_\gamma R_\delta \quad (0.90)$$

Da utledningen her blir gjort for styrefinnene, velges det å sette $\delta = \delta_1$. Utreget blir det:

$$R_f^n = R_\gamma R_{\delta_1}$$

$$\begin{bmatrix} c(\delta_1) & -s(\delta_1) & 0 \\ c(\gamma)s(\delta_1) & c(\gamma)c(\delta_1) & -s(\gamma) \\ s(\gamma)s(\delta_1) & s(\gamma)c(\delta_1) & c(\gamma) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\gamma) & -s(\gamma) \\ 0 & s(\gamma) & c(\gamma) \end{bmatrix} \begin{bmatrix} c(\delta_1) & -s(\delta_1) & 0 \\ s(\delta_1) & c(\delta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (0.91)$$

Dette gir \underline{n}_{fl}^n utskrevet:

$$\underline{n}_{fl}^n = \begin{bmatrix} -s(\delta_1) \\ c(\gamma)c(\delta_1) \\ s(\gamma)c(\delta_1) \end{bmatrix} \quad (0.92)$$

Total effektiv vindhastighet \underline{v}_{fw}^{in} ved trykksenteret til finnene er bestemt av objektets hastighet i forhold til lufta \underline{v}_{bw}^{in} , vinkelhastigheten til objektet og avstanden \underline{c} fra CG til CP for de aerodynamiske kreftene som påvirker finnene.

$$\underline{v}_{fw}^{in} = \underline{v}_{bw}^{in} - \underline{\omega}_b^{in} \times \underline{c} \quad (0.93)$$

Dette er den totale effektive vindhastigheten til objektet i punktet CP for finnene. CP for finnene ligger på rotasjonsaksen til forseksjonen. Derfor blir det ikke tatt hensyn til om den roterer. Total effektiv angrepsvinkel blir da:

$$\sin(\varepsilon_1) = \frac{-\langle \underline{v}_{fw}^{in}, \underline{n}_{fl}^n \rangle}{|\underline{v}_{bw}^{in}|} = \frac{-\langle \underline{v}_{bw}^{in}, \underline{n}_{fl}^n \rangle}{|\underline{v}_{bw}^{in}|} + \frac{\langle \underline{\omega}_b^{in} \times \underline{c}, \underline{n}_{fl}^n \rangle}{|\underline{v}_{bw}^{in}|} \quad (0.94)$$

Nå som vi har $\sin(\varepsilon_1)$ kan normalkraftkoeffisienten finnes ved:

$$\underline{C}^{FF1N} = C_{Na}^{fl} \sin(\varepsilon_1) \underline{n}_{fl}^n \quad (0.95)$$

Hvor \underline{C}^{FF1N} består av $\begin{bmatrix} C_X^{FF1N} & C_Y^{FF1N} & C_Z^{FF1N} \end{bmatrix}^T$

4.3.1.2 Aksialkraftkoeffisienten til styrefinnene

Normalkraftvektoren \underline{n}_{fl}^n og hastighetsvektoren \underline{v}_{fw}^{in} danner et plan. Vektoren til aksialkraften ligger langs skjæringslinjen mellom dette planet og planet som selve finnen utgjør. Aksialkraften er veldig liten i forhold til normalkraften. I tillegg blir endringene på aksialkraftvektoren liten hvis vi erstatter \underline{v}_{fw}^{in} med \underline{v}_{bw}^{in} . Denne vektoren er gitt ved:

$$\underline{e}_{xf} = \underline{n}_{fl}^n \times \frac{\underline{n}_{fl}^n \times \underline{v}_{bw}^{in}}{|\underline{n}_{fl}^n \times \underline{v}_{bw}^{in}|} \quad (0.96)$$

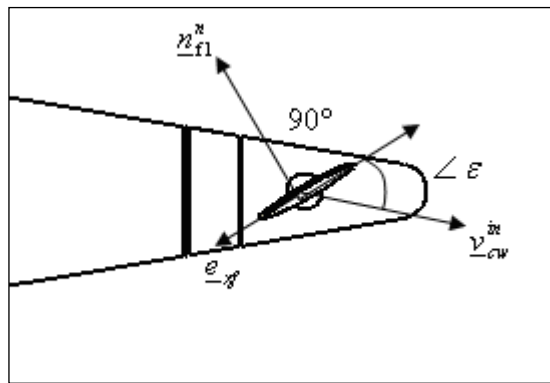
Ved hjelp av regelen for vektortrippelprodukt ($a \times (b \times c) = (a \cdot c)b - (a \cdot b)c$) kan det vises at

$$\underline{n}_{fl}^n \times (\underline{n}_{fl}^n \times \underline{v}_{bw}^{in}) = \langle \underline{n}_{fl}^n, \underline{v}_{bw}^{in} \rangle \underline{n}_{fl}^n - \underline{v}_{bw}^{in} \quad (0.97)$$

Vi har at

$$|\underline{n}_{fl}^n \times \underline{v}_{bw}^{in}| = |\underline{n}_{fl}^n| \cdot |\underline{v}_{bw}^{in}| \cdot \sin(\angle(\underline{n}_{fl}^n, \underline{v}_{bw}^{in})) \quad (0.98)$$

Hvis vi ser ned på planet som normalkraftvektoren \underline{n}_{fl}^n og hastighetsvektoren \underline{v}_{bw}^{in} danner får vi figuren:



Figur 4.3 Vektorforklaring på forseksjonen.

Videre har vi at $\sin(\epsilon_1 + 90^\circ) = \cos(\epsilon_1)$ og siden $|\underline{n}_{fl}^n| = 1$ får vi derfor likningen

$$\underline{e}_{xf} = \underline{n}_{fl}^n \times \frac{\underline{n}_{fl}^n \times \underline{v}_{bw}^{in}}{|\underline{n}_{fl}^n \times \underline{v}_{bw}^{in}|} = \frac{\langle \underline{n}_{fl}^n, \underline{v}_{bw}^{in} \rangle \underline{n}_{fl}^n - \underline{v}_{bw}^{in}}{|\underline{v}_{bw}^{in}| \cos(\epsilon_1)} \quad (0.99)$$

Vi setter $\cos(\epsilon_1) \approx 1$. Ved en angrepsvinkel på 15 grader gir dette en feil på 3.4% på en allerede liten størrelse. Likningen for normalvektoren som av figuren går langs finneplanet, blir da:

$$\underline{e}_{xf} = \frac{\langle \underline{n}_{fl}^n, \underline{v}_{bw}^{in} \rangle \underline{n}_{fl}^n - \underline{v}_{bw}^{in}}{|\underline{v}_{bw}^{in}|} \quad (0.100)$$

Aksialkraftkoeffisienten blir da

$$\underline{C}^{FF1A} = C_{x0}^{f1} \cdot \underline{e}_{xf} = C_{x0}^{f1} \frac{\langle \underline{n}_{fl}^n, \underline{v}_{bw}^{in} \rangle \underline{n}_{fl}^n - \underline{v}_{bw}^{in}}{|\underline{v}_{bw}^{in}|} \quad (0.101)$$

Her kan ikke momentet finnes ved $arm \times kraft$ -prinsippet direkte fordi det er koeffisienter som blir brukt. Likning (4.4) Løst med hensyn på C_M blir:

$$\begin{aligned}
 C_M &= \frac{M}{\rho v^2 \frac{1}{2} S \cdot D} = \frac{F \cdot c}{\rho v^2 \frac{1}{2} S \cdot D} \\
 &= \frac{C_F \rho v^2 \frac{1}{2} S \cdot c}{\rho v^2 \frac{1}{2} S \cdot D} = \frac{C_F \cdot c}{D} \\
 &= \frac{c}{D} C_F
 \end{aligned} \tag{0.102}$$

På vektorform blir det:

$$\underline{C}^{MF1T} = \frac{c}{D} \times (\underline{C}^{FF1N} + \underline{C}^{FF1A}) \tag{0.103}$$

Hvor \underline{C}^{MF1T} består av $\begin{bmatrix} 0 & C_Y^{MF1T} & C_Z^{MF1T} \end{bmatrix}^T$

4.3.1.3 Spinndempingskoeffisienten fra styrefinnene

Spinndempingskoeffisienten virker langs objektets x-akse og på framseksjonen. Koeffisienten C_D^{MF1} tar for seg momentkomponenten for den.

$$C_D^{MF1} = C_{LP}^{F1} \frac{P_F D}{2 |v_{bw}^{in}|} \tag{0.104}$$

Koeffisienten C_{LP}^{F1} er rullmomentkoeffisienten. Når framseksjon med styrefinnene roterer, skaper de bevegelse i luften rundt seg. Dette skaper et moment som vil prøve å dempe spinnnet. Forklaringen til det er helt analogt med forklaringen til rullmomentet

4.3.2 Rullstabiliseringsfinnene

For å bestemme bidraget fra rullstabiliseringsfinnene kan man legge sammen formlene for styrefinnene, med motsatt rettede finnevinkler og halverte koeffisienter. I praksis vil det si at alle ledd som er antisymmetrisk i finnevinkelen fjernes, og alle ledd som er symmetriske i finnevinkelen beholdes. Det vil si at ledd som inneholder $\sin(\delta_2)$ blir borte mens ledd som inneholder $\sin^2(\delta_2)$ beholdes. Disse finnene står 90 grader på styrefinnene og vi må derfor legge til 90 grader på γ -vinkelen. Den nye vinkelen er $\gamma_2 = \gamma + 90^\circ$ Vi bytter derfor ut γ med γ_2 når vi beregner koeffisienter fra rullstabiliseringsfinnene. For å finne

normalkraftkoeffisientene for rullstabiliseringsfinnene tar vi utgangspunkt i likning (4.33) og utvikler denne for rullstabiliseringsfinnene. Da blir utledningen slik:

$$\begin{aligned}\underline{C}^{FF2N} &= C_{Na}^{f2} \sin(\varepsilon_1) \underline{n}_{f2}^n \\ &= \frac{C_{Na}^{f2}}{V_a} \left(\left\langle \underline{\omega}_b^{in} \times \underline{c}, \underline{n}_{f2}^n \right\rangle - \left\langle \underline{v}_{bw}^{in}, \underline{n}_{f2}^n \right\rangle \right) \underline{n}_{f2}^n\end{aligned}\quad (0.105)$$

Likeledes, for å finne aksialkraftkoeffisientene for rullstabiliseringsfinnene tar vi utgangspunkt i likning (4.39) og utvikler denne for rullstabiliseringsfinnene. Da blir utledningen slik:

$$\begin{aligned}\underline{C}^{FF2A} &= C_{x0}^{f2} \cdot \underline{e}_{xf} \\ \underline{C}^{FF2A} &= C_{x0}^{f2} \frac{\left\langle \underline{n}_{f2}^n, \underline{v}_{bw}^{in} \right\rangle \underline{n}_{f2}^n - \underline{v}_{bw}^{in}}{\left| \underline{v}_{bw}^{in} \right|}\end{aligned}\quad (0.106)$$

Med betingelsene over, sammenlagt og utskrevet blir likningene da:

$$\begin{aligned}\underline{C}^{FF2} &= \underline{C}^{FF2N} + \underline{C}^{FF2A} \\ &= \begin{bmatrix} \left(-C_{Na}^{f2} s^2(\delta_2) - C_{x0}^{f2} c^2(\delta_2) \right) \frac{u_{bw}^{in}}{\left| \underline{v}_{bw}^{in} \right|} \\ - \left(C_{x0}^{f2} + C_{La}^{f2} c^2(\delta_2) s^2(\gamma) \right) \frac{v_{bw}^{in}}{\left| \underline{v}_{bw}^{in} \right|} + C_{La}^{f2} \frac{1}{2} c^2(\delta_2) s(2\gamma) \frac{w_{bw}^{in}}{\left| \underline{v}_{bw}^{in} \right|} + \\ C_{Na}^{f2} \frac{1}{2} c^2(\delta_2) s(2\gamma) \frac{qc}{\left| \underline{v}_{bw}^{in} \right|} + C_{Na}^{f2} c^2(\delta_2) s^2(\gamma) \frac{rc}{\left| \underline{v}_{bw}^{in} \right|} \\ \left(C_{La}^{f2} \frac{1}{2} c^2(\delta_2) s(2\gamma) \frac{v_{bw}^{in}}{\left| \underline{v}_{bw}^{in} \right|} - \left(C_{x0}^{f2} + C_{La}^{f2} c^2(\delta_2) c^2(\gamma) \right) \frac{w_{bw}^{in}}{\left| \underline{v}_{bw}^{in} \right|} - \\ C_{Na}^{f2} c^2(\delta_2) c^2(\gamma) \frac{qc}{\left| \underline{v}_{bw}^{in} \right|} - C_{Na}^{f2} \frac{1}{2} c^2(\delta_2) s(2\gamma) \frac{rc}{\left| \underline{v}_{bw}^{in} \right|} \end{bmatrix}\end{aligned}\quad (0.107)$$

Momentet blir da:

$$\underline{C}^{MF2T} = \frac{\underline{c}}{D} \times \underline{C}^{FF2}\quad (0.108)$$

Hvor \underline{C}^{MF2T} består av $\begin{bmatrix} 0 & C_Y^{MF2T} & C_Z^{MF2T} \end{bmatrix}^T$

Spindempingskoeffisienten virker om objektets x-akse og på framseksjonen. Koeffisienten C_D^{MF2} tar for seg momentkomponenten for den.

$$C_D^{MF2} = C_{LP}^{F2} \frac{P_F D}{2 |y_{bw}^{in}|} + C_{LD}^{F2} \delta_2 \quad (0.109)$$

Koeffisienten C_{LP}^{F2} er rullmomentkoeffisienten fra appendiks A. Når framseksjon med styrefinnene roterer, skaper de bevegelse i luften rundt seg. Dette skaper et moment som vil prøve å dempe spinnets. Forklaringen til det er helt analogt med forklaringen til rullmomentet på tilsvarende måte som for styrefinnene.

4.3.3 Moment fra generator

Det er tenkt at man kan benytte seg av forskjellen i vinkelhastigheten til for- og bakseksjonen for å generere elektrisitet som kan benyttes til servoer, sensorer og elektronikk.

Det blir derfor utledet en formel for belastningen dette gir på systemet. Denne energien vil i praksis bli tatt fra spinnets til bakseksjonen og det blir derfor bare tatt hensyn til treghetsensoren og vinkelhastigheten til bakseksjonen i denne utledningen. Som en forenkling antas det med andre ord at forseksjonen ikke roterer. Formelen for elektrisk energi er:

$$E = u \cdot i \cdot t = P \cdot t \quad (0.110)$$

Hvor E er total elektrisk energi, u er elektrisk spenning, i er elektrisk strøm, t er tiden og P er den effekten som i øyeblikket blir generert.

Formelen for energi i en roterende masse er:

$$E = \frac{1}{2} I \omega^2 \quad (0.111)$$

Hvor E er total rotasjonsenergi, I er treghetsmomentet om rotasjonsaksen og ω er vinkelhastigheten.

Formelen for momen er:

$$M = F \cdot r = mar \quad (0.112)$$

Hvor M er dreiemomentet, F er kraften som forårsaker dreiemomentet, r er avstanden fra momentaksen til kraftpåvirkningspunktet, m er massen til objektet som momentet virker på og a er translatorisk akselerasjon.

Akselerasjonen i (4.50) kan skrives som:

$$a = r\dot{\omega} \quad (0.113)$$

Vi får da formelen for moment:

$$M = mar = mr^2\dot{\omega} \quad (0.114)$$

Hvor ω er forskjell i vinkelhastighet på for- og bakseksjonen.

Treghetsmomentet for massen er:

$$I = \int_m r^2 d(m) \quad (0.115)$$

Dette er om x-aksen til {b}-systemet. På et hvert sted er derfor:

$$r^2 = y^2 + z^2 \quad (0.116)$$

Formelen for massetettheten er:

$$\rho = \frac{m}{V} \quad (0.117)$$

Setter (4.54) og (4.55) inn i (4.53) og integrerer volumet i stedet for massen:

$$M_G = \int_V (y^2 + z^2) \rho d(V) \dot{\omega} \quad (0.118)$$

Volumintegralet utskrevet:

$$M_G = \int_{-x}^x \int_{-y}^y \int_{-z}^z (y^2 + z^2) \rho dz dy dx \dot{\omega} = I \dot{\omega} \quad (0.119)$$

Sammenhengen mellom energi og effekt kan skrives som:

$$\dot{E} = P \quad (0.120)$$

Hvis man deriverer (4.49), bruker kjerneregelen og setter inn for effekt og moment, får man formlene:

$$\frac{d(E)}{dt} = \frac{d(E)}{d\omega} \cdot \dot{\omega} = I\omega \cdot \dot{\omega} = P \quad (0.121)$$

$$\begin{aligned} P &= I\dot{\omega}\omega \\ &= M_G\dot{\omega} \end{aligned} \quad (0.122)$$

Løst med hensyn på moment blir det:

$$M_G = \frac{P}{\omega} \quad (0.123)$$

Her er P den effekten som blir tatt ut av systemet. Skal man ha den elektriske effekten man tar ut av en generator, må man ta hensyn til virkningsgraden på generatoren.

4.3.4 Moment fra kulelager

Forseksjonen og bakseksjonen er tenkt adskilt med to kulelager.

Da disse seksjonene roterer med forskjellige hastigheter vil det være en viss gjensidig momentpåvirkning via kulelageret. På nettsidene til "Svenska Kullagerfabriken AB" (skf.com) [4] får vi formelen for momentpåvirkning via kulelager:

$$M_L = \frac{\mu P d}{2} \quad (0.124)$$

Hvor μ er en konstant friksjonskoeffisient (gitt av tabellen under) for kulelageret, P er ekvivalent dynamisk lagerlast (N) og d er diameter på hullet til lageret. Tabellen for μ , hentet fra [4] og ser slik ut:

Table 1: Constant coefficient of friction μ for unsealed bearings	
Bearing type	Coefficient of friction μ
Deep groove ball bearings	0,0015
Angular contact ball bearings	
single row	0,0020
double row	0,0024
four-point contact ball bearings	0,0024
Self-aligning ball bearings	0,0010
Cylindrical roller bearings	
- with cage, when $F_a \approx 0$	0,0011
- full complement, when $F_a \approx 0$	0,0020
Needle roller bearings	0,0025
Taper roller bearings	0,0018
Spherical roller bearings	0,0018
CARB toroidal roller bearings	0,0016
Thrust ball bearings	0,0013
Cylindrical roller thrust bearings	0,0050
Needle roller thrust bearings	0,0050
Spherical roller thrust bearings	0,0018

Tabell 4.1 Koeffisienter for kulelagre.

Hvis $\frac{F_A}{F_R} \leq e$ kan ekvivalent dynamisk lagerlast, P, kan finnes ved formelen:

$$P = F_R + Y_1 F_A \quad (0.125)$$

Ellers kan ekvivalent dynamisk lagerlast finnes ved formelen:

$$P = X F_R + Y_2 F_A \quad (0.126)$$

F_R er radiell last, F_A er aksiallast, Y_1 , X og Y_2 er skaleringsfaktorer for denne lagertypen og e er en koeffisient som blir brukt for valg av likning (4.63) eller (4.64).

Aksialkraften kan finnes med formelen:

$$F_A = \sqrt{(F_{FX} - F_{AX})^2} \quad (0.127)$$

Aksialkraften virker langs x-aksen til prosjektilet, og F_{FX} er kraften som virker på forseksjonen langs x-aksen mens F_{AX} er kraften som virker på bakseksjonen langs x-aksen.

Skaleringsfaktorene og koeffisienten e , blir hentet fra [4]. For det lageret som er tenkt å brukes her, ser databladet slik ut:

Angular contact ball bearings, single row, for paired mounting, back-to-back arrangement									
Principal dimensions			Basic load ratings		Fatigue load limit	Speed ratings	Limiting speed	Mass	Designation
d	D	2B	C	C ₀	P _u	Reference speed			
mm			kN		kN	r/min		kg	-
20	47	28	21,5	15,3	0,65	14000	19000	0,22	2 x 7204 BECBM

Calculation factors
k _r 0,095
e 1,14
X 0,57
Y ₁ 0,55
Y ₂ 0,93
Y ₀ 0,52

Figur 4.4 Produktinformasjon kulelager.

Ved kommunikasjon med [11] er det kommet fram til at man bør prøve å bruke 2 stykk vinkelkontaktlager med enkel rad fordi disse lagrene tåler en god del last i både aksial og radiell retning, og tåler også høye hastigheter.

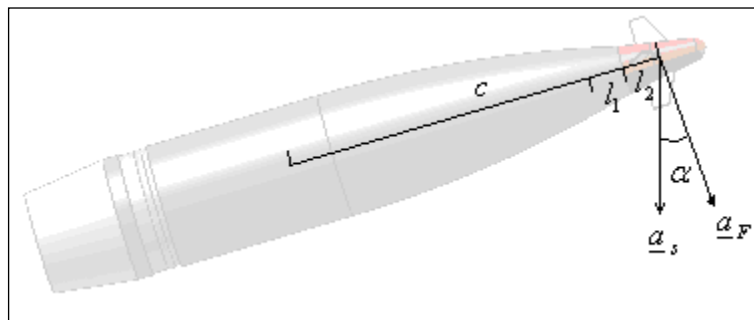
Dette gir friksjonskoeffisienten $\mu = 0.002$. Kraftene som påvirker lagrene virker i både aksial og radial retning. Vi antar at det bakerste lageret er montert opp slik at det vil ta opp all aksialkraft. Det fremste lageret vil da bare være et støttelager mot radielle krefter mens det bakre lagret vil ha både radielle og aksielle kraftpåvirkninger. Det blir under utledningen av de radielle kreftene antatt at $CP_F = CG_F$. Det vil si at massemiddepunktet til forseksjonen er på samme plass som trykksenteret til finnene. Dette gjøres bare fordi at brannrøret ikke fins ennå, og man vet da ikke hvor CG_F blir. Det er i alle fall lett å justere senere da dette bare har innvirkning på noen lengder i formlene. Ser man bort fra at objektet er i flukt og konsentrerer seg kun om sentripetalhastigheten til CG_F gitt av vinkelhastigheten, får man formelen:

$$\underline{v}_F = \underline{\omega}_b^{in} \times \underline{c} \quad (0.128)$$

Akselerasjonen som er gitt av denne, er:

$$\underline{a}_s = \frac{1}{\Delta t} (\underline{v}_{F2} - \underline{v}_{F1}) \quad (0.129)$$

Figuren viser et øyeblikksbilde av et spinnende prosjektil. På grunn av nutasjonsbevegelsen vil denne akselerasjonen virke i retning som vist på figuren.



Figur 4.5 Akselerasjonsretningen på framseksjonen når objektet nuterer.

Fra dette øyeblikksbildet ser man at:

$$\underline{a}_F = c(\alpha) \underline{a}_s \quad (0.130)$$

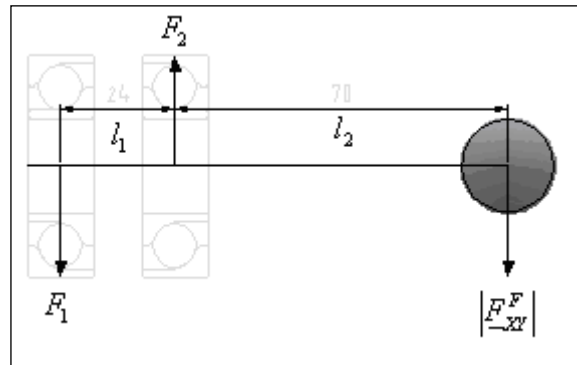
Da det er relative små vinkelutslag, vil det her bli forenklet til:

$$\underline{a}_F \approx \underline{a}_s \quad (0.131)$$

Hvis man i $\{n\}$ -ramma ser på \underline{a}_F som en vektor i 2 dimensjoner (y og z -retning) og tar høyde for kraften som finnes påvirker objektet i de retningene, får vi formelen for total kraft i massemidtpunktet til forseksjonen:

$$\underline{F}_{YZ}^F = m_F \underline{a}_F + \begin{bmatrix} F_{FY} \\ F_{FZ} \end{bmatrix} \quad (0.132)$$

Vi får nå denne figuren for lageroppheget:



Figur 4.6 Kraftpåvirkning på lageroppheget.

Som nevnt tidligere kan det hende det blir justeringer på målene. Formlene vil allikevel være like. Her er $l_1 = 24 \text{ mm}$ og $l_2 = 70 \text{ mm}$. Målene er tatt med antakelse om at man bruker lager med betegnelse "7204 BECBP" fra skf.com og har en avstand mellom dem på 10 mm.

Tar man hensyn til at finnene kan ha et utslag og presesjonsbevegelsens påvirkning, vil kreftene som påvirker lagrene radially bli:

$$F_1 = \frac{l_2}{l_1} |F_{YZ}^F| \quad (0.133)$$

$$F_2 = F_1 + |F_{YZ}^F| \quad (0.134)$$

Hvis $F_A/F_R \leq e$ for hvert av lagrene, blir den totale kraften på hvert av lagrene:

$$F_{L1} = F_1 + 0,55F_A \quad (0.135)$$

$$F_{L2} = F_2 \quad (0.136)$$

Og hvis $F_A/F_R > e$ for hvert av lagrene, blir den totale kraften på hvert av lagrene:

$$F_{L1} = 0,57F_1 + 0,93F_A \quad (0.137)$$

$$F_{L2} = 0,57F_2 \quad (0.138)$$

Momentet fra kulelagrene blir da:

$$\begin{aligned} M_{LS} &= M_{L1} + M_{L2} \\ &= \frac{\mu d}{2}(F_{L1} + F_{L2}) \end{aligned} \quad (0.139)$$

Dette momentet har benevnningen Nmm (Newton millimeter). Benevnningen som blir brukt er Nm. Derfor blir det totale momentet:

$$M_L = \frac{\mu d}{2 \cdot 1000}(F_{L1} + F_{L2}) \quad (0.140)$$

4.4 Oppsummering

De totale krefter som påvirker forseksjonen, er:

$$\underline{F}_F = \frac{\rho |\underline{v}_{bw}^{in}|^2 S}{2} \left(\begin{bmatrix} C_{XF}^{FA} \\ 0 \\ 0 \end{bmatrix} + \underline{C}^{FF1N} + \underline{C}^{FF1A} + \underline{C}^{FF2} \right) \quad (0.141)$$

De totale krefter som påvirker bakseksjonen, er:

$$\underline{F}_A = \underline{F}_G + \frac{\rho |\underline{v}_{bw}^{in}|^2 S}{2} \begin{bmatrix} C_{XA}^{FA} \\ C_Y^{FA} + C_Y^{FM} \\ C_Z^{FA} + C_Z^{FM} \end{bmatrix} \quad (0.142)$$

De totale krefter som påvirker objektet, er:

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} = \begin{bmatrix} F_{FX} + F_{AX} \\ F_{FY} + F_{AY} \\ F_{FZ} + F_{AZ} \end{bmatrix} \quad (0.143)$$

De totale momenter som påvirker objektet, er:

$$\begin{bmatrix} l_F \\ l_A \\ m \\ n \end{bmatrix} = \frac{\rho |\underline{v}_{bw}^{in}|^2 S \cdot D}{2} \begin{bmatrix} C_D^{MF1} + C_D^{MF2} \\ C_X^{MR} \\ C_Y^{MA} + C_Y^{MM} + C_Y^{MQ} + C_Y^{MF1T} + C_Y^{MF2T} \\ C_Z^{MA} + C_Z^{MM} + C_Z^{MQ} + C_Z^{MF1T} + C_Z^{MF2T} \end{bmatrix} + \begin{bmatrix} M_G + M_L \\ -M_G - M_L \\ 0 \\ 0 \end{bmatrix} \quad (0.144)$$

5 Forklaring på simuleringsprogrammet og filer

I appendiks B ligger filene til simuleringsprogrammet og til diverse skript som behandler de forskjellige dataene fra simuleringsresultatene. Dette er m-filer som kjører under matlab. Her er en gjennomgang av disse filene.

5.1 Simuleringsprogrammet

Simuleringsprogrammet beregner prosjektilets posisjon, stilling, vinkelhastighet og hastighet i diskrete tidssteg gjennom hele banen. Avhengig av hva man skal gjøre med programmet så er det mye man kan forandre på i det. Slik som programmet fungerer nå, har det 5 inngangsparametre og 4 utgangsparametre. Figuren under viser funksjonsskallet til simuleringsprogrammet.

```
[MO xpos ypos uk2b] = SimProsjekttilMFiner( ...
    Styrefinneutslag, ...
    størrelsesfaktor1 , ...
    størrelsesfaktor2, ...
    Påslagstid, ...
    Rullvinkel, ...
    MomentVirkning);
```

Figur 5.1 Funksjonsskallet til simuleringsprogrammet.

I punktene under blir det en gjennomgang av parametrene fra venstre mot høyre. Ved gjennomgang av disse punktene kommer det egentlig fram hvordan man bruker dette programmet hvis man ikke er avhengig av å gjøre noe inni det. Hvis man ønsker å legge på vind er det mulig å legge på en konstant vind i en retning. Da må man inn i filen ”SimProsjekttilMFiner” på linje 31 som vist i figuren under.

```
30 % Vindhastighet i [x y z] retning
31 - Vind = [0; 0; 0]; % m/s
32 % Stille 0,0-0,2 Røyken stiger rett opp.
33 % Flau vind 0,3-1,5 En kan se vindretningen av røykens drift.
34 % Svak vind 1,6-3,3 Bladene på trærne rører seg.
35 % Lett bris 3,4-5,4 Vinden strekker lette flagg og vimpler.
36 % Løber bris 5,5-7,9 Vinden løfter støv og løse papirer.
37 % Frisk bris 8,0-10,7 Småtrær med løv begynner å svaie.
38 % Liten kuling 10,8-13,8 Store greiner og mindre stammer rører seg.
39 % Stiv kuling 13,9-17,1 Hele trær rører på seg.
40 % Sterk kuling 17,2-20,7 Vinden brykker kvister av trærne.
41 % Liten storm 20,8-24,4 Hele store trær svaier og hiver.
42 % Full storm 24,5-28,4 Sjelden inne i landet. Trær rykkes opp med
43 % Sterk storm 28,5-32,6 Forekommer sjelden og følges av store
44 % ødeleggelser.
45 % Orkan 32,6- Forekommer meget sjelden. Uvanlig store
46 % ødeleggelser. (32,6 m/s tilsvarer 117km/h)
47
```

Figur 5.2 Vindvektor i simuleringsprogrammet.

Tallene til venstre viser hvilken linje dette gjelder. Som man ser av figuren, er det i kommentarfeltet tatt med hva vinden kalles når styrken varieres. Dette er forøvrig tatt fra [5]. Vinden er i dette programmet en vektor i $\{i\}$ systemet.

Ønsker man å forandre andre startparametre som utgangshastighet, skytevinkel og antall mellomsteg mellom lagrede data, gjøres dette i filen "SimProsjektIMFinner" på linje 19, 21 og 25 som vist i figuren under.

18	-	Power	= 100;	% Drar ut 100 watt fra spinnet
19	-	VO	= 829;	% Starthastighet [m/s] 829
20	-	Asimut	= 0;	% Dreining om z akse [rad]
21	-	Elevasjon	= 45*pi/180;	% Dreining om y akse [rad]
22	-	%Startsteg	= 0.0001;	% Steglengde pr iterasjon.
23	-	A.N	= 100000;	% antall iterasjoner og lagrede
24	-			% variabler 5000
25	-	A.mellomsteg	= 5;	% Antall mellomsteg for hver
26	-			% lagrede variabler.
27	-	Spinn	= 2*pi*VO/ (TW*Dref);	

Figur 5.3 Oppstartsparmetre.

Vinkelen til rullstabiliseringsfinnene er programmert slik at det skal sammenlagt bli 0 moment på x-aksen til forseksjonen. Dette sørger for at det ikke er noen rullbevegelse på forseksjonen, og det blir da lettere å analysere data fra forsøkene.

5.1.1 Returparameter nummer 1

På figur 5.1 heter returparameter nummer 1 M0. Dette er en variabel av typen struct og har følgelig flere undervariabler. De undervariablene som finnes i M0 kan sees på tabellen under. Som en forkortelse vil styrefinnene bli kalt finne 1 og rullstabiliseringsfinnene bli kalt finne 2 i denne tabellen.

Navn:	Størrelse:	Forklaring:
L	1x1	Viser hvor mange steg med data som er returnert i M0
X0f2	100000x1	Aksialkraftkoeffisient for finne 2 gjennom hele turen
X0f1	100000x1	Aksialkraftkoeffisient for finne 1 gjennom hele turen
Naf2	100000x1	Normalkraftkoeffisient for finne 2 gjennom hele turen
Naf1	100000x1	Normalkraftkoeffisient for finne 1 gjennom hele turen
lpf2	100000x1	Spinndempingskoeffisient fra finne 2 gjennom hele turen
lpf1	100000x1	Spinndempingskoeffisient fra finne 1 gjennom hele turen
ldf2	100000x1	Spinndempingskoeffisient fra finne 2 gjennom hele turen
ldf1	100000x1	Spinndempingskoeffisient fra finne 1 gjennom hele turen
Laf2	100000x1	Det samme som Naf2-X0f2
Laf1	100000x1	Det samme som Naf1-X0f1
ma	100000x1	Tippmomentkoeffisient gjennom hele turen

X2	100000×1	Aksialkraftkoeffisient på objektet gjennom hele turen
X0	100000×1	Aksialkraftkoeffisient på objektet gjennom hele turen
Na	100000×1	Normalkraftkoeffisient på objektet gjennom hele turen
lp	100000×1	Rullmomentet på objektet gjennom hele turen
m _q	100000×1	Pitch dempemoment på objektet gjennom hele turen
m _{pa}	100000×1	Magnusmomentkoeffisient på objektet gjennom hele turen
y _{pa}	100000×1	Magnuskraftkoeffisient på objektet gjennom hele turen
M	100000×1	Machnummer som ble brukt gjennom hele turen
t	100000×1	Tidspunkt for hvert returnerte steg gjennom hele turen
FO	100000×3	Kraft på objektet gjennom hele turen
BS	100000×3	Hastigheten til objektet gjennom hele turen
IP	100000×3	Posisjonen til objektet gjennom hele turen
MO	100000×4	Momentet på objektet gjennom hele turen
EA	100000×4	Stillingen til objektet gjennom hele turen
AS	100000×4	Vinkelhastigheten til objektet gjennom hele turen
OBS	1×1	Et varsel om at den totale hastigheten er utenfor mactabellen
mellomsteg	1×1	Forteller hvor mange steg det ble brukt men ikke tatt med mellom to returnerte steg

Tabell 5.1 Variabler i returparameter 1.

Som man ser i tabellen er bufferene satt til en dybde på 100000 plasser. Dette kan man endre i ”SimProsjekttilMFinner” på linje 23. Se figur Figur 5.3: Oppstartsparemetre. De plassene som er blitt fylt med simuleringsdata, befinner seg innenfor de L første verdiene. de andre vil typisk være lik 0. Hvis man for eksempel skal plote x, -y banen til prosjektilet og har M0, vil kommandoen i matlab kunne se slik ut:

```
>> plot(MO.IP(1:MO.L,1), MO.IP(1:MO.L,2))
```

Figur 5.4 Plotkommando i matlab.

5.1.2 Returparameter nummer 2 og 3

Returparameter nummer 2 er x-posisjonen til treffpunktet, og returparameter nummer 3 er y-posisjonen til treffpunktet.

5.1.3 Returparameter nummer 4

Returparameter nummer 4 gir vanligvis tilbake størrelseskoeffisienten til rullstabiliseringsfinnene uforandret. Men, hvis av en eller annen årsak rullstabiliseringsfinnene under simuleringen skulle ha hatt behov for å vri seg mer enn 10 grader (som er utenfor det definerte arbeidsområdet til finnene), vil dette kompenseres med å utvide finnestørrelsen. Forslaget til den nye størrelseskoeffisienten vil da bli returnert i stedet for den gamle. For å fange opp dette kan denne sammenliknes med størrelsesforholdet man hadde da simuleringen startet.

5.1.4 Inngangsparameter nummer 1

Inngangsparameter nummer 1 er utslaget til styrefinnene i radianer. I startøyeblikket er utslaget lik 0 radianer, men etter at tiden i inngangsparameter nummer 4 er utgått byttes dette med inngangsparameter nummer 1.

5.1.5 Inngangsparameter nummer 2

Inngangsparameter nummer 2 er størrelsesfaktoren til styrefinnene. Hvis denne verdien er lik 0, vil finnene være som borte; er de lik 1 vil finnene ha sin opprinnelige størrelse; er de 2 vil finnene være dobbelt så stor osv.

5.1.6 Inngangsparameter nummer 3

Inngangsparameter nummer 3 er størrelsesfaktoren til rullstabiliseringsfinnene. Hvis denne verdien er lik 0, vil finnene være som borte; er de lik 1 vil finnene ha sin opprinnelige størrelse; er de 2 vil finnene være dobbelt så stor osv.

5.1.7 Inngangsparameter nummer 4

Inngangsparameter nummer 4 bestemmer når styrefinnene skal få utslaget som ligger i inngangsparameter nummer 1. Før denne tiden er utslaget lik 0 radianer.

5.1.8 Inngangsparameter nummer 5

Inngangsparameter nummer 5 gir startvinkelen til forseksjonen. Programmet vil sørge for å vri rullstabiliseringsfinnene på en slik måte at det blir null moment om x-aksen til forseksjonen.

5.1.9 Inngangsparameter nummer 6

Inngangsparameter nummer 6 gir et testmoment mellom for- og bakseksjonen. Dette momentet er en erstatter for kulelager- og generatormomentet.

5.2 Plotteprogrammet

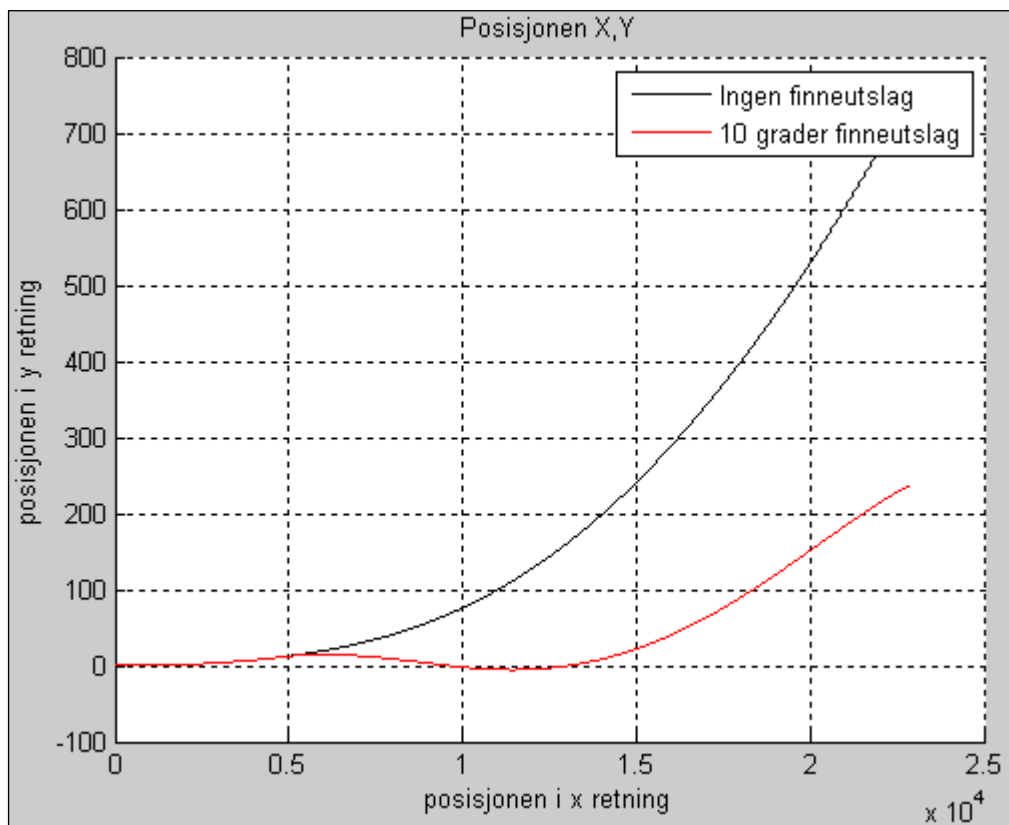
Plotteprogrammet heter ”PlotResults” og tar et variabelt antall parametre. Parametrene må være på formen som returparameter nummer 1 i simuleringsprogrammet. I structen må det da også da føyes til et variabelnavn som heter ”MedFinner”. Denne variabelen skal ha verdien 1 hvis man bruker det simuleringsprogrammet som denne rapporten beskriver. Det ble laget et simuleringsprogram for et prosjektil uten finner først, og plotteprogrammet er kompatibelt med dette programmet også. For dette programmet (som er gjort i forbindelse med denne rapporten, men ikke følger med denne rapporten) skal variabelen ”MedFinner” ha verdien 0. Simuleringsprogrammet som simulerer et prosjektil med finner er bygget på simuleringsprogrammet som simulerer et prosjektil uten finner, og de er derfor veldig kompatible. I structen skal man også ha med en variabel som heter ”Color”. Denne variabelen er en tekststreng som brukes på samme måte som man setter farge på plottene i plot-kommandoen til matlab. Den siste variabelen man skal ha med, heter ”Navn”. Den er en beskrivende tekst som vil dukke opp i forklaringsteksten til plottet, (Gitt at man har gjort to simuleringer kalt M0 og M1 og lagt til de 3 ekstra parametrene som vist på figuren under).

```
MO.MedFinner    = 1;
MO.Color        = 'k';
MO.Navn         = 'Ingen finneutslag';

M1.MedFinner    = 1;
M1.Color        = 'r';
M1.Navn         = '10 grader finneutslag';
```

Figur 5.5 Skripteksempel.

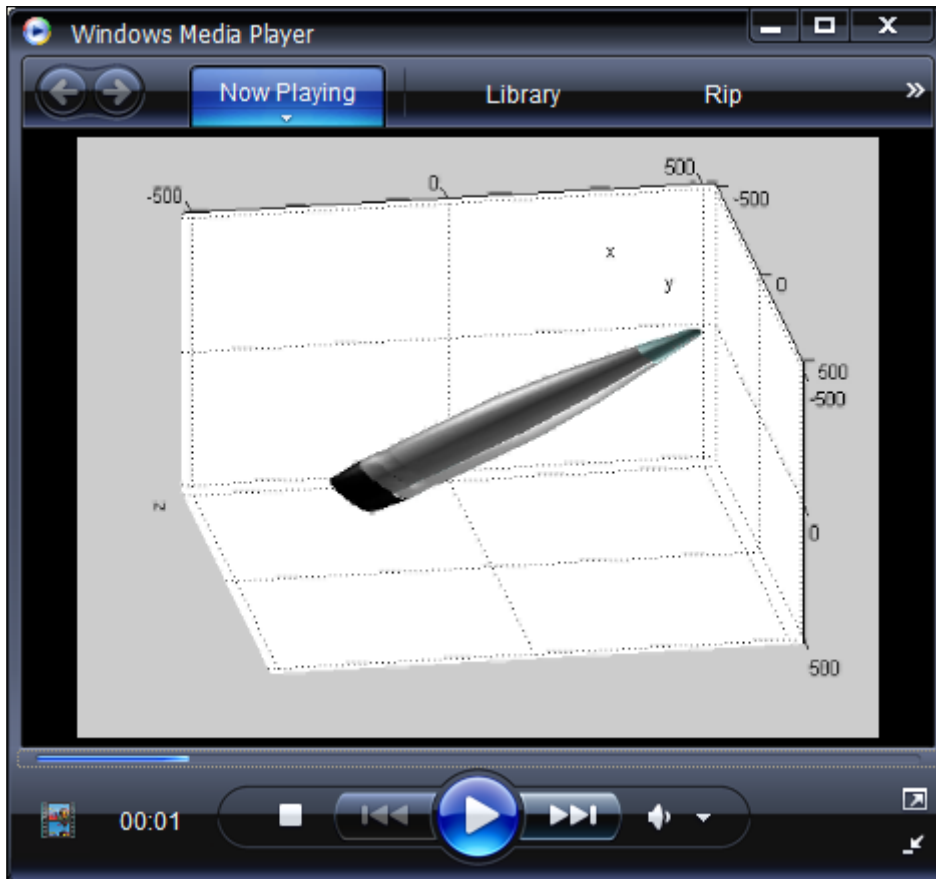
En kjøring av plotteprogrammet med kommandoen PlotResults(M0, M1) vil for eksempel da kunne gi et posisjonsplott som vist på figuren under.



Figur 5.6 Eksempel på et posisjonsplott.

5.3 Filmprogrammet

Det er mulig å lage en film av stillingen på prosjektilet under flukten. Filmen tar da ikke for seg banen til prosjektilet. Et eksempel på hvordan filmen vil se ut i Windows mediaplayer er vist i figuren under. Dette programmet ble laget i forbindelse med testing av simuleringsprogrammet.



Figur 5.7 Et stillbilde laget med filmprogrammet.

Hvis man i matlab skriver "help Film", kommer denne hjelpeteksten opp:

```
>> help Film
Film(Navn, Vinkler, Steg)
Navn = navnet på filmen
Vinkler = [Nx3] eulervinkler
Steg = steglengden
```

Figur 5.8 Hjelpetekst til filmprogrammet.

Skulle man laget en film som het "prosjektilet" og brukt M0 fra dataene over, kunne man ha skrevet kommandoen `Film('prosjektilet', M0.EA(1:M0.L, 2:4), 100)`. Her ville man da bare ha brukt hver hundrede eulervinkel i matrisen som `M0.EA(1:M0.L, 2:4)` utgjør.

6 Tester og resultater

Hvis disse dataene hadde vært representert i {b}- i stedet for i {n}-systemet, ville mange av plottene vært fulle av oscillasjoner. Det er nå som resultatene skal vises, at fordelene med å representere dette i en ikke-rullende ramme kommer til syne. Z-aksen har positiv retning nedover, men for å få et visuelt bedre plott er det skiftet fortegn på alle posisjonsverdiene til z-aksen. Felles for alle testene under er at de er gjort uten vind, med en starthastighet på 825m/s, en startvinkel på 45 grader i forhold til flat jord og alle skudd er gjort i retning nordover med mindre det er beskrevet annerledes i den aktuelle testen. Startspinnets finnes av formelen:

$$p_{A0} = \frac{2\pi}{T_w D} v_0 \quad (0.145)$$

Hvor $T_w = 20 \frac{\text{Kalibre}}{\text{Omdreining}}$ er riflestigningen i løpet, og v_0 er starthastigheten.

Styrefinneutslaget er satt til 0 grader i starten, og settes til 10 grader etter at det har gått 10 sekunder.

Det blir under disse testene gjort en forenkling av modellen. Vi sier at momentet fra kulelagrene M_L og momentet fra generatoren M_G er lik null. Også tilføyer vi et nytt testmoment M_T i stedet for M_L og M_G . Dette gjøres fordi det nå er lettere å beregne et utslag δ_2 som matcher dette momentet på rullstabiliseringsfinnene, fordi M_T lett kan holdes konstant under hele flukten. Momentlikningen blir da seende slik ut.

$$\begin{bmatrix} 0 \\ l_A \\ m \\ n \end{bmatrix} = \frac{\rho |y_{bw}^{in}|^2 S \cdot D}{2} \begin{bmatrix} C_D^{MF1} + C_D^{MF2} \\ C_X^{MR} \\ C_Y^{MA} + C_Y^{MM} + C_Y^{MP} + C_Y^{MF1T} + C_Y^{MF2T} \\ C_Z^{MA} + C_Z^{MM} + C_Z^{MP} + C_Z^{MF1T} + C_Z^{MF2T} \end{bmatrix} + \begin{bmatrix} M_T \\ -M_T \\ 0 \\ 0 \end{bmatrix} \quad (0.146)$$

For å få et best mulig måleresultat ønsker man at framseksjonen ikke skal ha noe særlig rullbevegelse under flukten. Dette gjøres ved å justere finneutslaget δ_2 slik at man nuller ut momentpåvirkningen til framseksjonen. Da forseksjonen ikke ruller, blir $C_D^{MF1} = 0$ og $C_D^{MF2} = C_{LD}^{F2} \delta_2$. Setter man dette inn i (6.2) og løser med hensyn til δ_2 , får man likningen:

$$\delta_2 = -\frac{2M_T}{\rho |y_{bw}^{in}|^2 S D C_{LD}^{F2}} \quad (0.147)$$

Man kan av likning (6.3) tenke seg at hvis δ_2 er stor vil også M_T være stor. Dette er en ulineær modell og det går ikke an å bare øke δ_2 for å kompensere for stort moment. Som et tankeeksperiment kan man tenke seg at hvis δ_2 blir økt til 360 grader, eller 2π radianer, hvor stort blir momentet da? Finnen er jo gått en hel runde og kommet tilbake til utgangspunktet. Dette er noe som ikke er synlig i disse likningene. Som en tommelfingerregel er det blitt bestemt at

finneutslagene ikke skal være over 10 grader, eller $\frac{\pi}{18}$ radianer. Hvis ikke annet er sagt er $M_T = 0$ under testingene.

6.1 Sammenlikning av treffpunkt med skytetabell

Simuleringsmodellen tar utgangspunkt i et OEF3BB prosjektil hvor baseburn-elementet ikke er tent. Det prosjektilet som da vil være best å sammenlikne med, er et OEF3HB som er et prosjektil uten baseburn. Denne sammenlikningen tar derfor for seg simuleringsmodellen mot skytedata [8] på et OEF3HB prosjektil. I denne testen er utgangshastigheten satt til 804,7 m/s og elevasjon til 45,73 grader. Dette for å få en best mulig sammenlikning med skytedata tatt fra [8]. Med en elevasjon på 812.9 mils (På side 139 i [8]) har et OEF3HB-prosjektil en skyteavstand på 23 700 meter og en drift på 37,7 mils. ($360^\circ = 2\pi rad = 6400 mils$).

Hvis man står i origo på {i}-systemet, tilsvarer det et treffpunkt med koordinatene:

$$\begin{aligned}x_i &= 23684m \\y_i &= 877m\end{aligned}\tag{0.148}$$

Treffpunktet med dette simuleringsprogrammet er:

$$\begin{aligned}x_i &= 22378m \\y_i &= 753m\end{aligned}\tag{0.149}$$

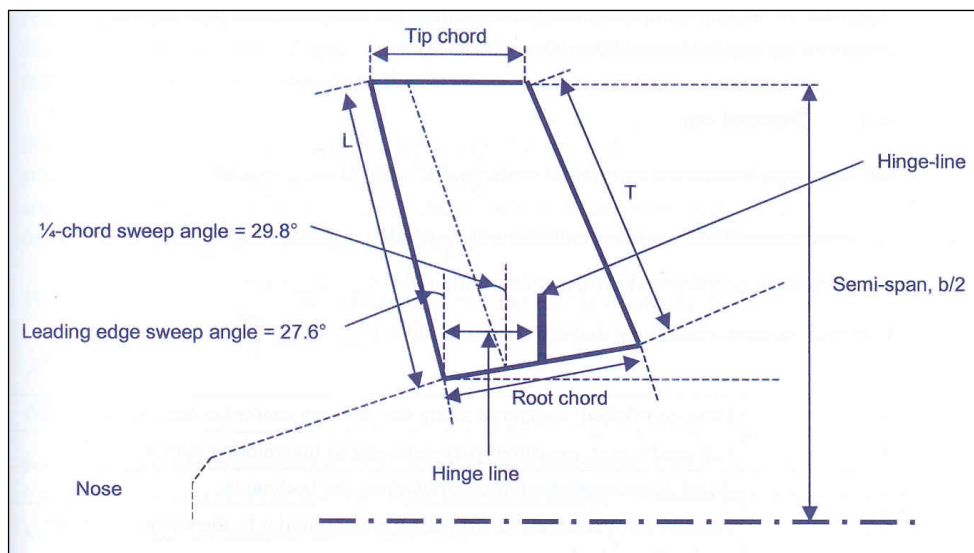
Dette gir en differanse på:

$$\begin{aligned}\Delta x_i &= 1306m \\ \Delta y_i &= 124m\end{aligned}\tag{0.150}$$

Forskjellene på disse prosjektilene er at simuleringsprogrammet bruker en OEF3BB som er et prosjektil med baseburn, bare at i disse simuleringene er baseburn ikke tent. Konsekvensene er at man har et tyngre prosjektil i simuleringsprogrammet enn i tabellen. I simuleringsprogrammet går man ut fra at jorden er flat. På en avstand som dette vil det si at når prosjektilet i simuleringsprogrammet treffer bakken er enda prosjektilet i tabellen 40 meter over bakken. Sist men ikke minst så har prosjektilet i simuleringsprogrammet noen finner som sørger for større drag.

6.2 Test av finnestørrelser

Det ønskes å teste ut hvor store finner man kan bruke før objektet blir ustabil. Finnestørrelsene man kan bruke, er veldig avhengig av momentpåvirkningen mellom for- og bakseksjonen. Det er tre parametre som er avhengig av hverandre. Størrelsen på styrefinnene, størrelsen på rullstabiliseringsfinnerne og størrelsen på momentet mellom for- og bakseksjonen. Et designforslag for finnene er hentet fra [1] og er vist i figuren og tabellen under.



Figur 6.1 Finnedesignet.

Fin config	Root chord	Tip chord	Length LE	Length TE	Semi span	Fin area	Equivalent rectangle	
	cr (mm)	ct (mm)	L (mm)	T (mm)	b/2 (mm)	A (mm ²)	chord (mm)	width (mm)
Large	32.5	25.0	55.4	55.5	75	1406	28.6	49.1
Medium	26.0	20.0	44.1	44.1	65	895	22.9	39.1
Small	19.5	15.0	32.8	32.8	55	498	17.1	29.1
Mini	12.9	10.0	21.5	21.5	45	217	11.4	19.1

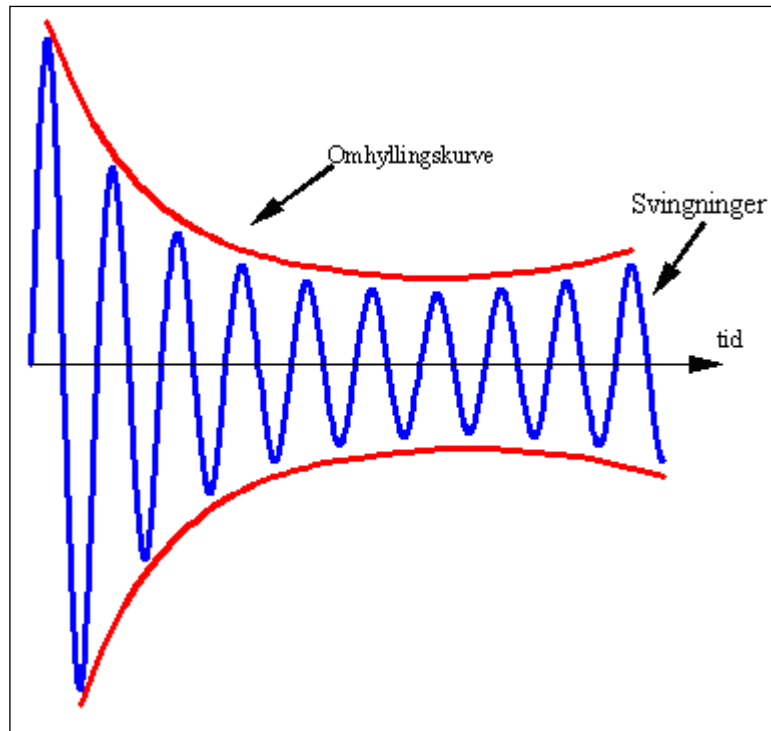
Tabell 6.1 Data på finnedesignet.

Det er tatt utgangspunkt i finner type medium til styrefinner og type mini til rullstabiliseringsfinner. Finnene er designet som vanlige plater uten noen form for krumming.

6.2.1 Testoppsett

Under beregningen av koeffisientene til finnene er det ikke tatt hensyn til hvordan prosjekttilkroppen påvirker aerodynamikken til finnene og motsatt. Med de antakelser som er gjort, så er koeffisientene lineært avhengig av arealet til finnene. Det vil si at blir koeffisienten dobbelt så stor, vil det tilsvare en finne som er dobbelt så stor. Under dette oppsettet vil derfor koeffisientene for finnene bli multiplisert med en størrelsesfaktor for lettere å kunne justere størrelsene opp og ned.

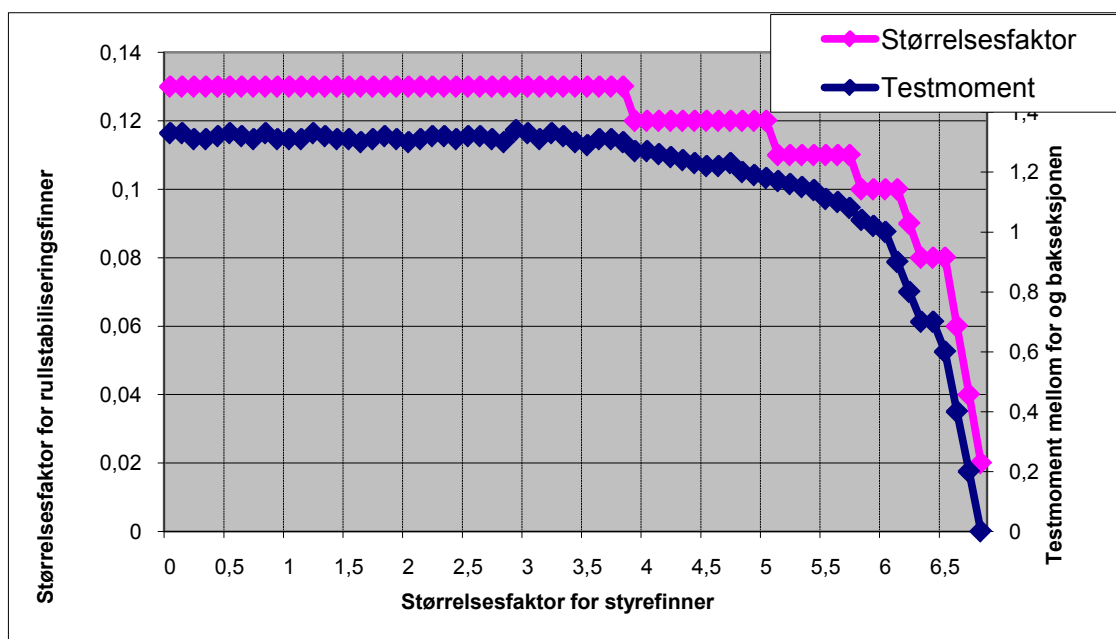
Når prosjektilet er stabilt, vil omhyllingskurven til yaw-svingningene gå mot en fast verdi med tiden. Vi kan tenke oss at hvis prosjektilet aldri lander, så vil det til slutt stoppe å spinne og derfor vil det bli ustabil. Absoluttverdien til omhyllingskurvene vil da begynne å øke i verdi.



Figur 6.2 Omhyllingskurve for yaw -svingningene.

På denne testen er størrelsesfaktoren til styrefinnene variert fra 0 til 7 for så å se på hvor stort testmoment M_T man kan bruke mellom for- og bakseksjonen gjennom hele flyturen før prosjektilet blir ustabil. Så er det sett på hvilken størrelsesfaktor til rullstabiliseringsfennene som gir dette momentet ved et finneutslag på maks 10 grader i løpet av turen.

6.2.2 Resultater

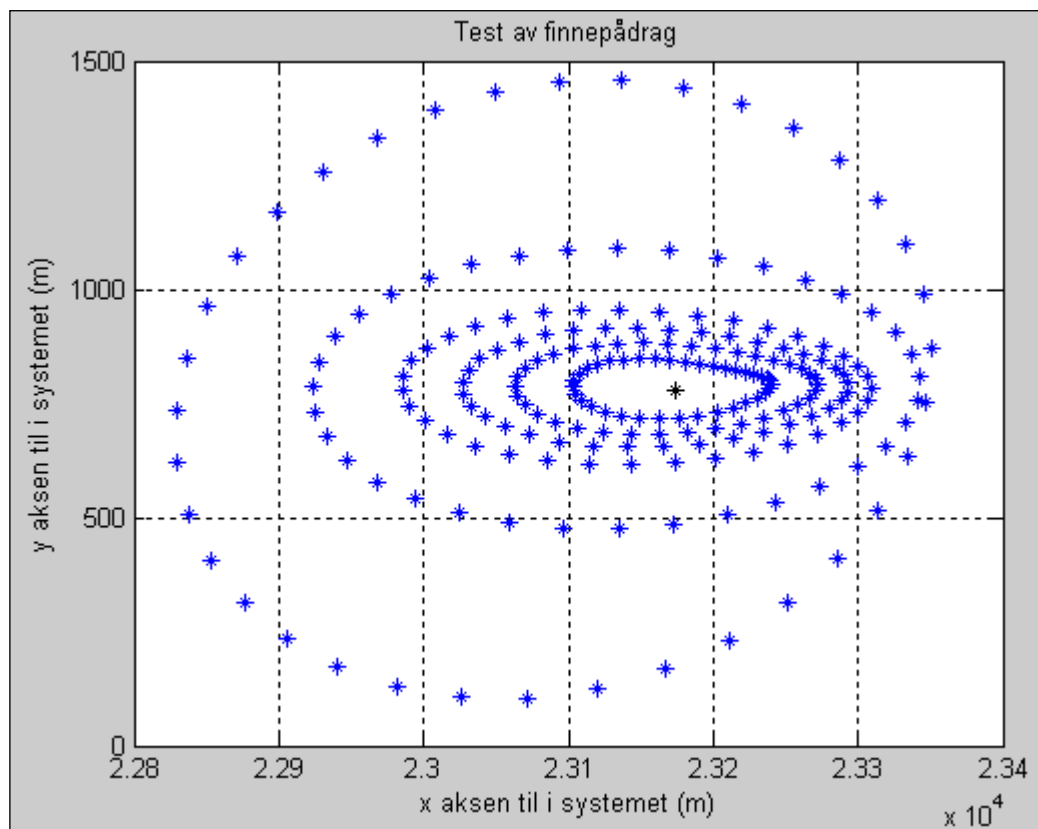


Figur 6.3 Moment i forhold til størrelsesfaktor på finner.

Man ser av figur 6.3 at hvis størrelsesfaktoren til styrefinnene er lik 1 vil systemet tåle et moment mellom for- og bakseksjonen på litt over 1,3 Nm. For å kunne gi dette momentet gjennom hele turen med en maks utslag på 10 grader på rullstabiliseringsfinnerne, må størrelsen til disse finnerne være minst 0,13 ganger så stor som de finnerne som er foreslått. Man ser også av figuren at hvis størrelsesfaktoren til styrefinnene er lik 7 eller over, vil systemet bli ustabilisert uansett hvilket testmoment man påtrykker.

6.3 Test av finnepådrag

Denne testen viser hvor langt ut av kurs et finnepådrag kan klare å få systemet gitt at størrelsesfaktorene på finnerne er lik 1. Det er ikke hensiktsmessig å legge opp til å korrigere banen til prosjektilet umiddelbart etter avfiring. Man antar at skytset er rettet inn best mulig, og at man regulerer inn eventuell avdrift underveis. På denne testen er rullvinkelen fra 0 til 350 grader med 10 graders stepp og et styrefinneutslag på 10 grader etter 10 til 60 sekunders flytid med 10 sekunders trinn. Dette tilsvarer 216 simuleringer med 216 forskjellige treffpunkter som vist på figuren under.

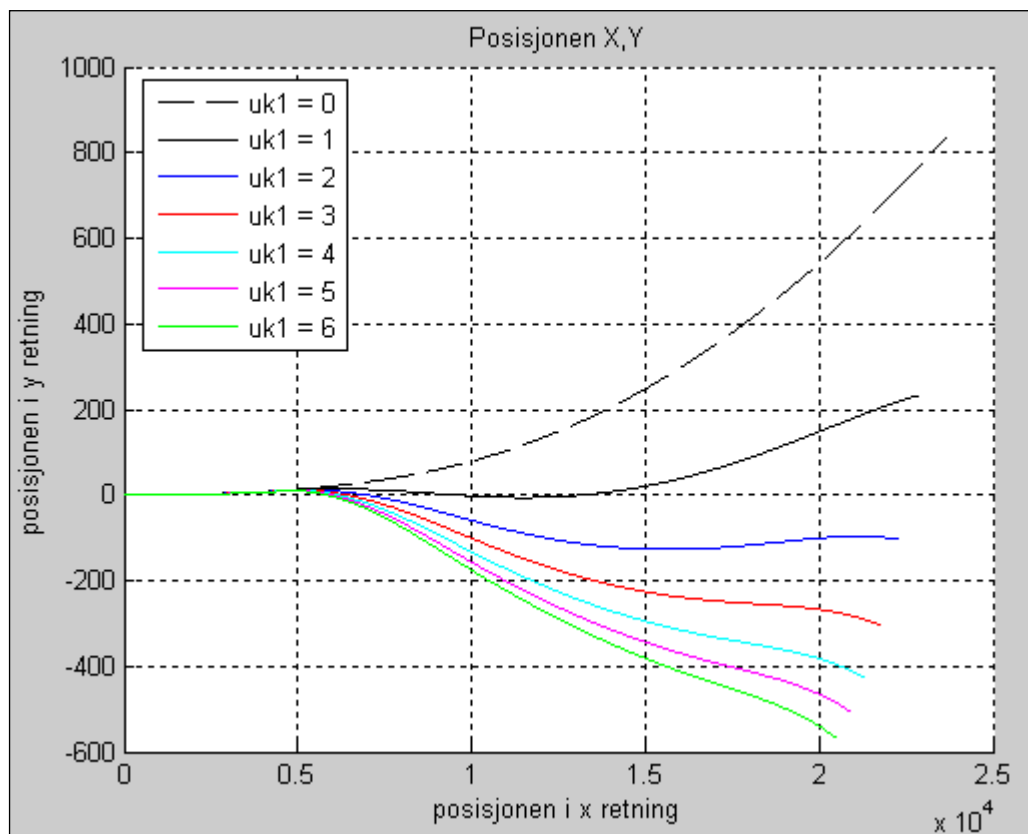


Figur 6.4 Korreksjonsevne.

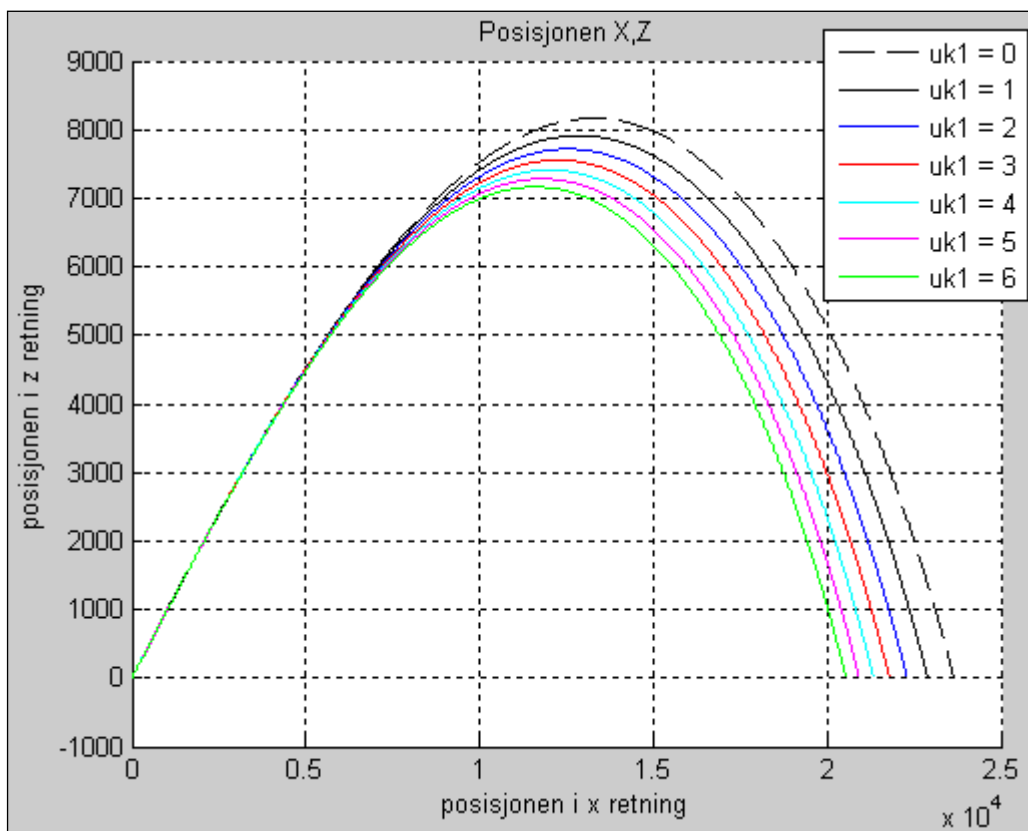
Den innerste sirkelen markerer treffpunktene hvor man senest satte på et styrefinneutslag. Det vil si etter 60 sekunder flytid. Treffpunktet i midten er uten finnepådrag og er bare tatt med for sammenlikning.

6.4 Styrefinnestørrelsens innflytelse på banen.

Størrelsen på styrefinnene har en god del å si på styreevnen til prosjektilet. Men, jo større finner man bruker, jo større blir draget. Figurene under viser banen til prosjektilet med forskjellige størrelser på styrefinnene. Her er ukl størrelsesfaktoren til styrefinnene. Det vil si at på de forskjellige plotter tilsvarer styrefinnestørrelsen $895 \cdot ukl$



Figur 6.5 Skalering av styrefinnenes påvirkning på banen i x- og y-retning.



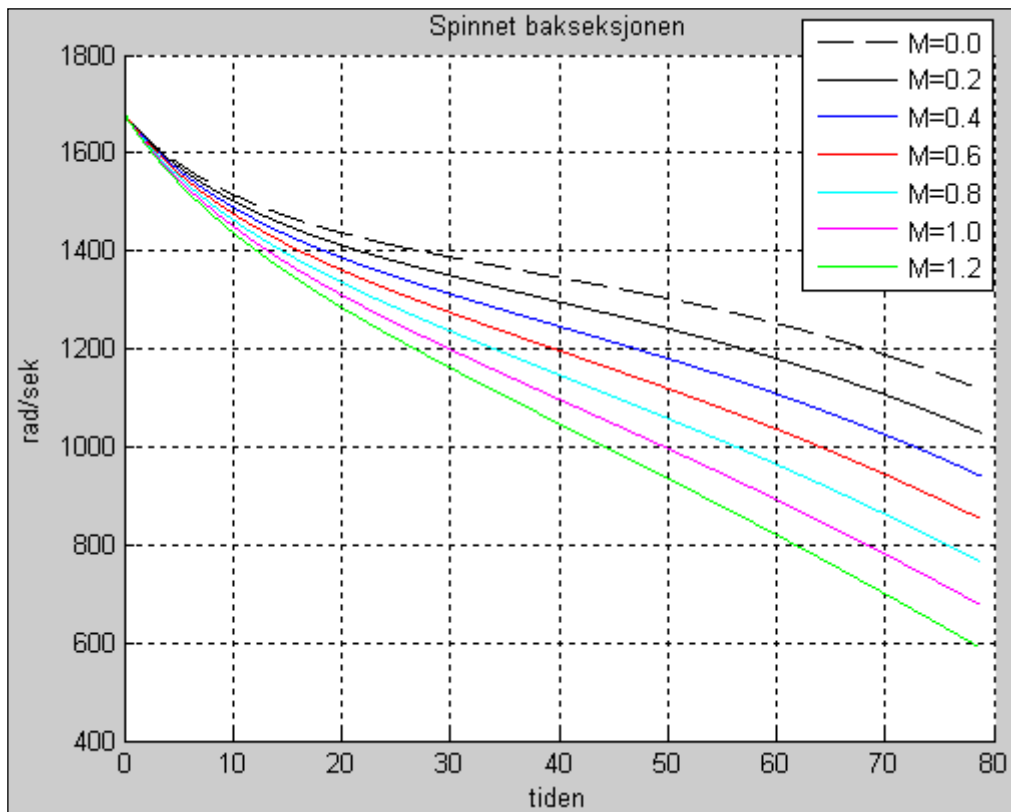
Figur 6.6 Skalering av styrefinnenes påvirkning på banen i x- og z-retning.

Som beskrevet før er positiv z-akse nedover. På figur 6.6 er z-aksen invertert.

Etter 10 sekunder flygetid slås styrefinnene på med et utslag på 10 grader. Finnene ligger langs z-aksen til {c} (forseksjonen). Som sagt i begynnelsen av dette kapitlet, blir δ_2 på rullstabiliseringsfynnene ordnet slik at det ikke blir noen moment langs x-aksen til {c}. Det vil si at forseksjonen ikke har noe særlig rullbevegelse. Det betyr at et positivt utslag på styrefinnene gir en kraft som dytter prosjektilet i positiv y -retning. Legg merke til at prosjektilet svinger mer i retning av negativ y -retning i forhold til hvis det ikke hadde vært styring.

6.5 Momentpåvirkningen mellom for- og bakseksjonens innvirkning på spinnnet.

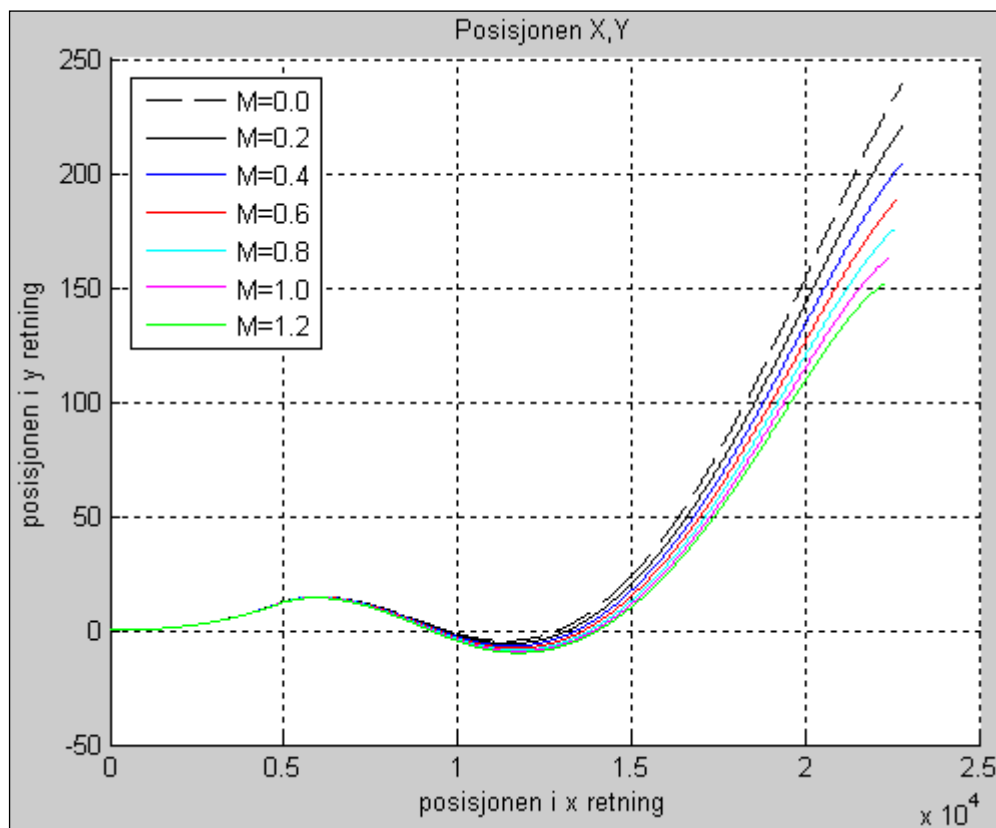
Det som momentpåvirkningen mellom for- og bakseksjonen gjør, er å bremse spinnnet til bakseksjonen samtidig som det prøver å øke spinnnet til forseksjonen. Her er rullstabiliseringsfynnene ordnet slik at forseksjonen står i ro. Figuren under viser hvordan spinnnet på bakseksjonen blir bremsset opp med forskjellige testmomenter M_T fra 0 til 1,2 Nm.



Figur 6.7 Momentpåvirkning mellom for- og bakseksjonens innvirkning på spinnnet bak.

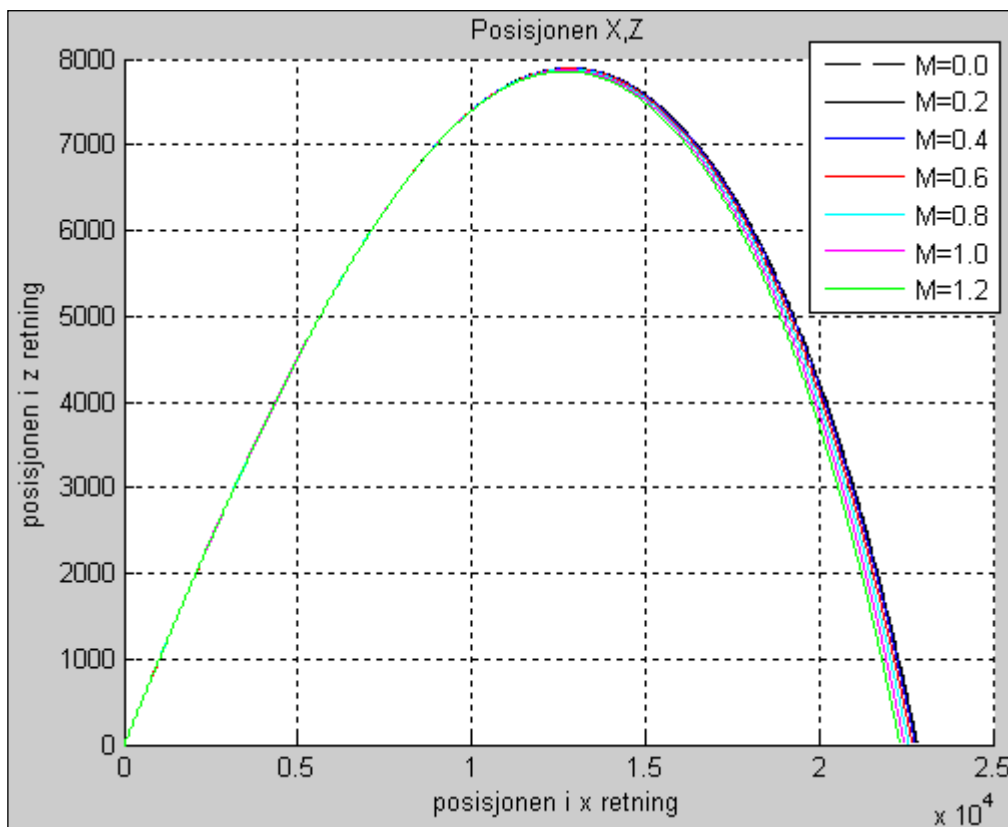
6.6 Momentpåvirkningen mellom for- og bakseksjonens innvirkning på banen

Det er spinnnet til bakseksjonen som sørger for avdriften i positiv y-retning. Større momentpåvirkning mellom for- og bakseksjonen vil sørge for redusert spinn på bakseksjonen og dermed også redusert avdrift. Dette viser figuren under.



Figur 6.8 Momentpåvirkning mellom for- og bakseksjonens innvirkning på banen i x og y retning.

Vi ser av figuren under at redusert spinn også gir noe redusert lengde i x-retning.



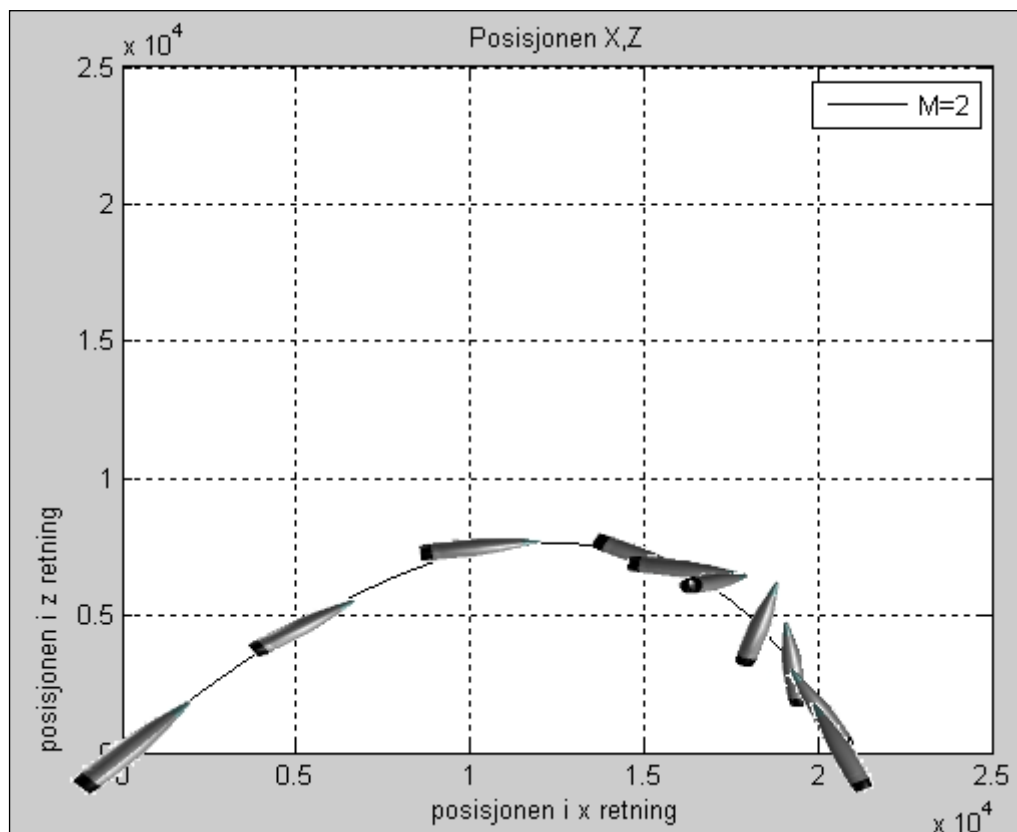
Figur 6.9 Momentpåvirkning mellom for- og bakseksjonens innvirkning på banen i x og z retning.

Som beskrevet før er positiv z-akse nedover. På figur 6.9 er z-aksen invertert.

6.7 For stor momentpåvirkningen mellom for- og bakseksjonen

Blir momentet mellom for- og bakseksjonen for stort, bremses spinnets på bakseksjonen til slutt så mye at prosjektilet blir ustabil. Under denne testen er momentet mellom for- og bakseksjonen satt til 3.5 Nm. Dette fører egentlig til at spinnets blir lik 0 rad/sek på bakseksjonen i løpet av flygetiden. Fortsetter dette momentet vil bakseksjonen begynne å spinne andre veien igjen. Dette er under denne testen tatt hånd om ved å nulle ut momentet når spinnets stopper opp.

På figuren under er det vist x, z -posisjonen til prosjektilet. Oppå denne figuren er det satt på bilder av prosjektilets stilling i forhold til hvor den er i banen.



Figur 6.10 Ustabilt prosjektil.

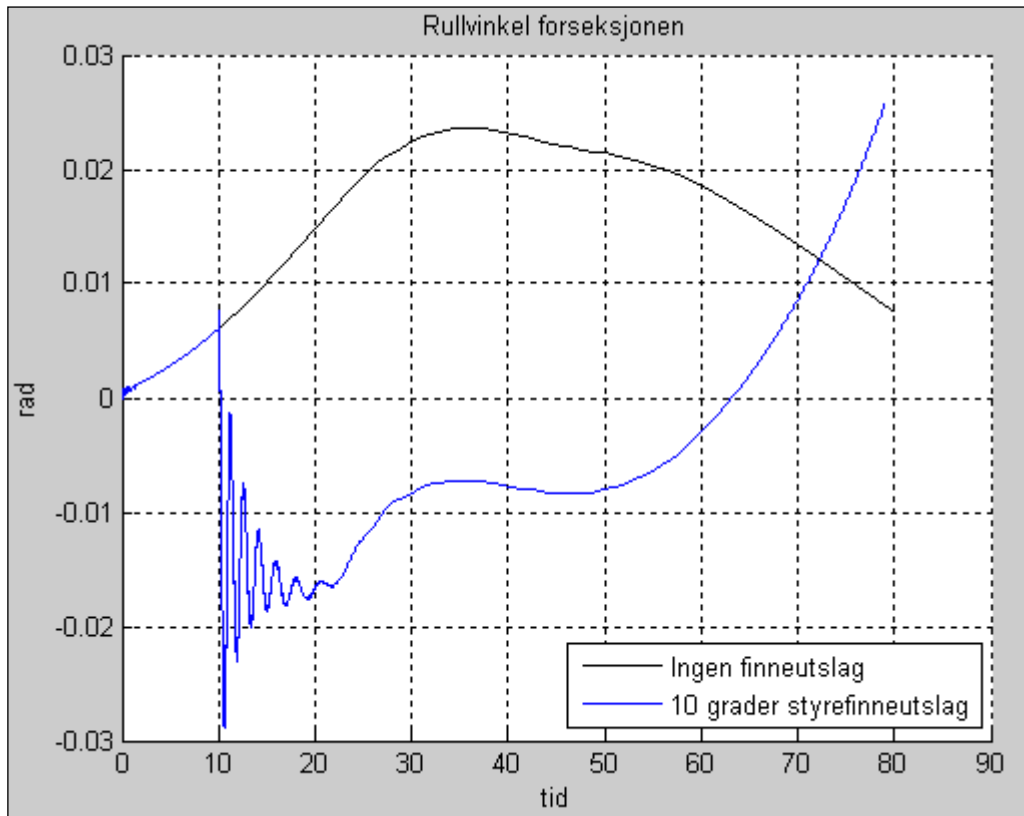
Som beskrevet før, er positiv z-akse nedover. På figur 6.10 er z-aksen invertert.

På denne figuren sees det ikke så godt nøyaktig hvilken bevegelse prosjektilet har når det blir ustabil. Det er ikke hensiktsmessig å ta med så tette bilder på denne figuren. Men, det man ser er at prosjektilet stabiliserer seg med å fly bakover. Dette viser alle animasjoner som her er gjort på ustabile prosjektiler med styrefinner. Men, de aerodynamiske koeffisientene som er beregnet for dette prosjektilet gjelder bare når det flyr framover, og de blir mer og mer ugyldige jo større angrepsvinkel man har. Hvis man skyter med pil og bue vil massemidelpunktet være foran trykksenteret på pilen, og den er følgelig stabil. Det ser ut til å være akkurat det samme som skjer her. Når prosjektilet spinner, er den stabil selv om trykksenteret er foran massemidelpunktet, men når spinnet opphører, snur den seg slik at trykksenteret blir bak massemidelpunktet, og den er stabil igjen, bare at den flyr baklengs.

6.8 Styrefinneutslags påvirkning på eulervinklene

Da bakseksjonen spinner ganske fort, er det ikke så veldig innteressant å se på rullvinkelen til bakseksjonen. Her vil det derfor bare bli vist plott av de andre eulervinklene. Finnene står under disse testene langs z-aksen til forseksjonen og gir etter 10 sekunders flygetid et finneutslag på 10 grader. Dette fører til at prosjektilet starter å presesere og nutere. Nutasjonsbevegelsen er raskere enn presesjonsbevegelsen.

6.8.1 Rullvinkelen til forseksjonen



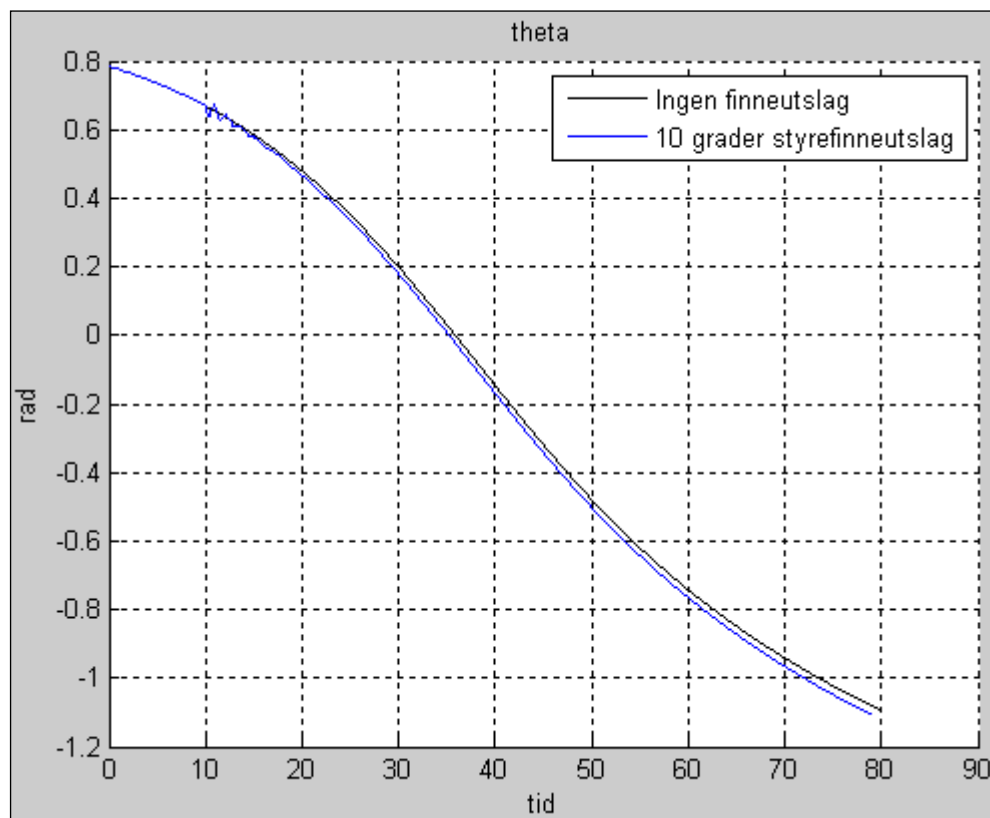
Figur 6.11 Rullvinkel forseksjonen.

Fra likning (3.47) kan vi se at likningen for rullvinkelen er

$$\dot{\phi}_F = p_F + t(\theta)r \quad (0.151)$$

Da momentet langs x-aksen til forseksjonen er lik 0, vil også p_F være lik 0. Det vil si at det er bare leddet $t(\theta)r$ som gir utslag her. Rett etter at man har vridd finnene 10 grader starter prosjektilet med en presesjons- og nutasjonsbevegelse. Disse bevegelsene dør etter hvert ut. På dette plottet er det virkningen av presesjonsbevegelsen man ser best.

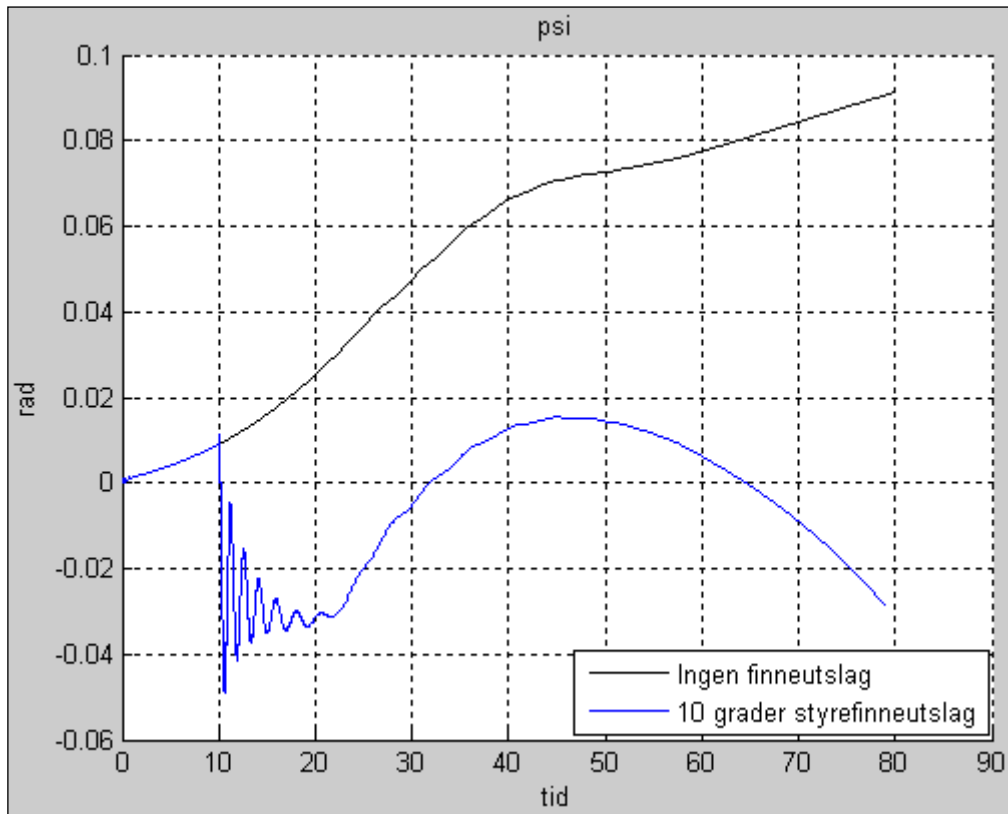
6.8.2 Theta vinkelen



Figur 6.12 Theta-vinkelen.

Som man ser av figuren, har ikke dette finneutslaget så stor innvirkning på θ -vinkelen. Det er fordi sånn som finnene står i denne testen, prøver de å skape et moment om z-aksen til forseksjonen noe som for det meste vil føre til økt ψ -vinkel og ikke så stor økning i θ -vinkelen.

6.8.3 Psi-vinkelen



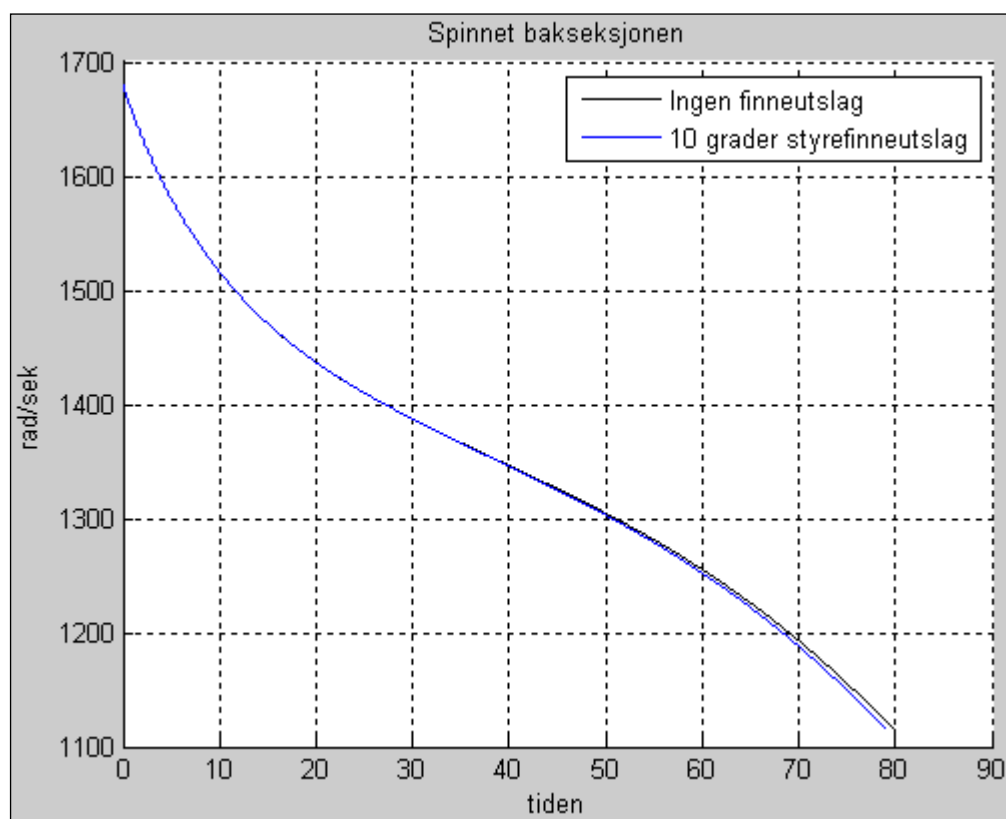
Figur 6.13 Psi-vinkelen.

Da finneutslaget prøver å skape et moment rundt z-aksen til forsystemet (z-aksen til forsystemet er ikke så ulikt z-aksen til {n} systemet) vil det naturligvis være størst virkning på denne av de tre eulervinklene som er vist her.

6.9 Styrefinneutslags påvirkning på vinkelhastighetene

Siden forseksjonen i disse simuleringene starter med 0 vinkelhastighet langs sin x-akse, og på grunn av at forseksjonen er ordnet slik at den ikke har noen moment på seg i x-retning vil denne vinkelhastigheten forbli 0. Derfor er ikke den vist her.

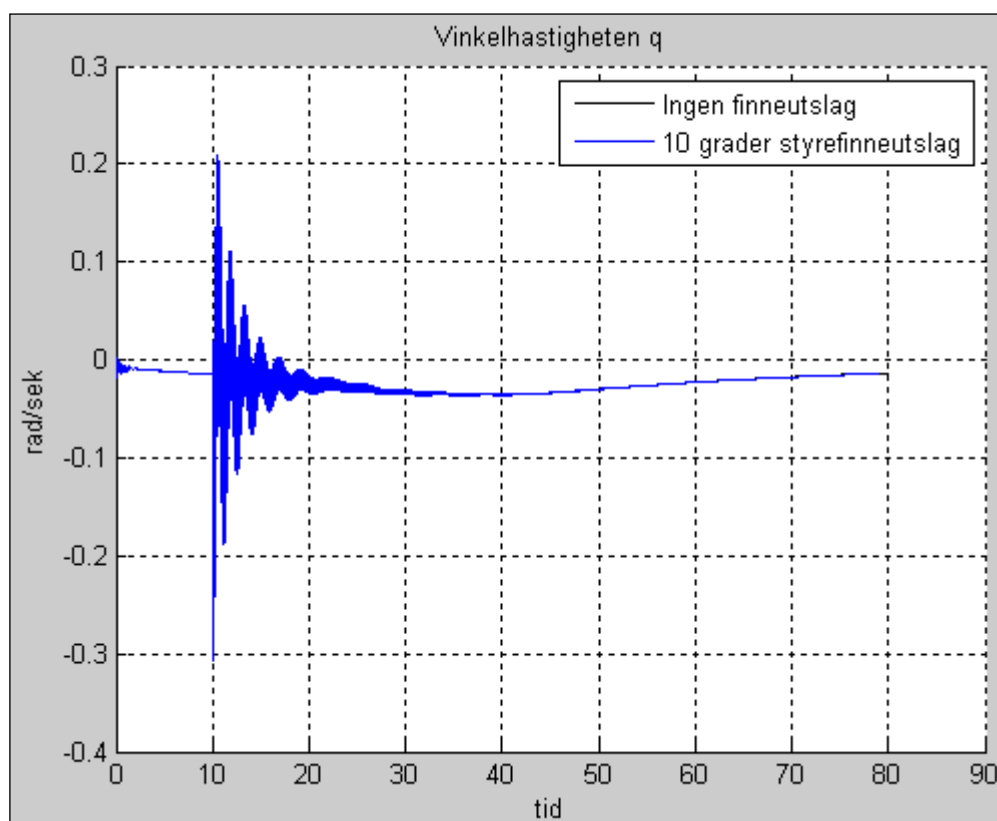
6.9.1 Spinnet bak



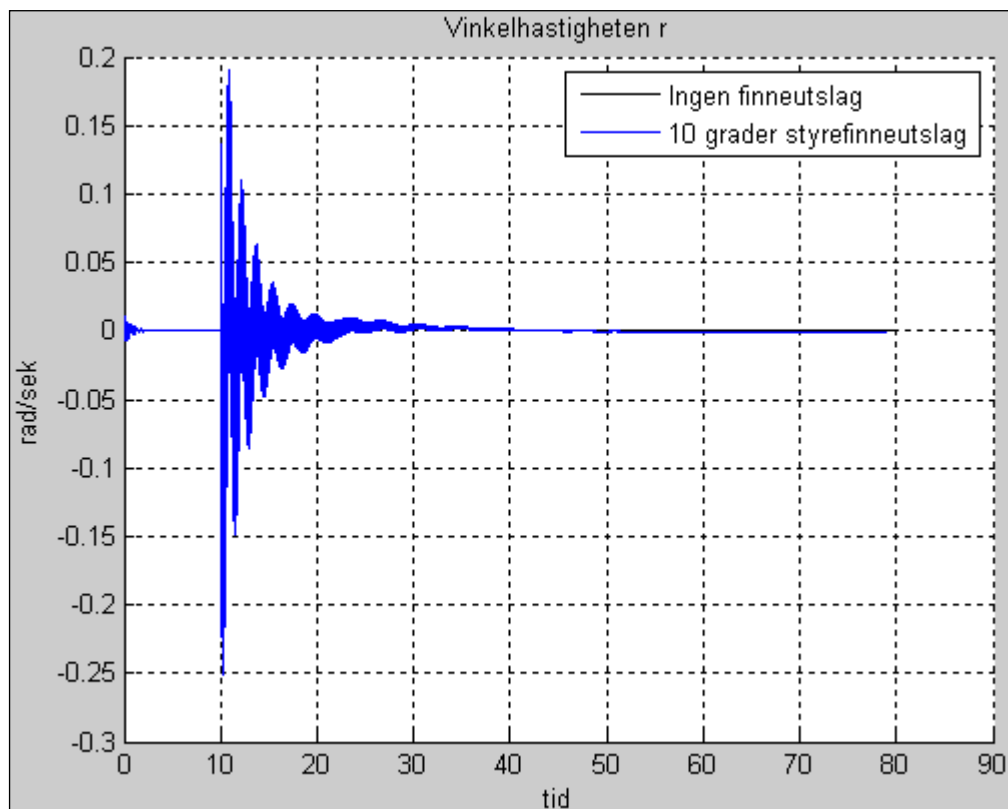
Figur 6.14 Spinnet til bakseksjonen.

Som man ser har ikke finneutslaget noe særlig påvirkning på spinnet til bakseksjonen. Det er fordi finneutslaget prøver å skape et moment som er vinkelrett på dette spinnet.

6.9.2 Vinkelhastighetene q og r



Figur 6.15 Vinkelhastigheten q .



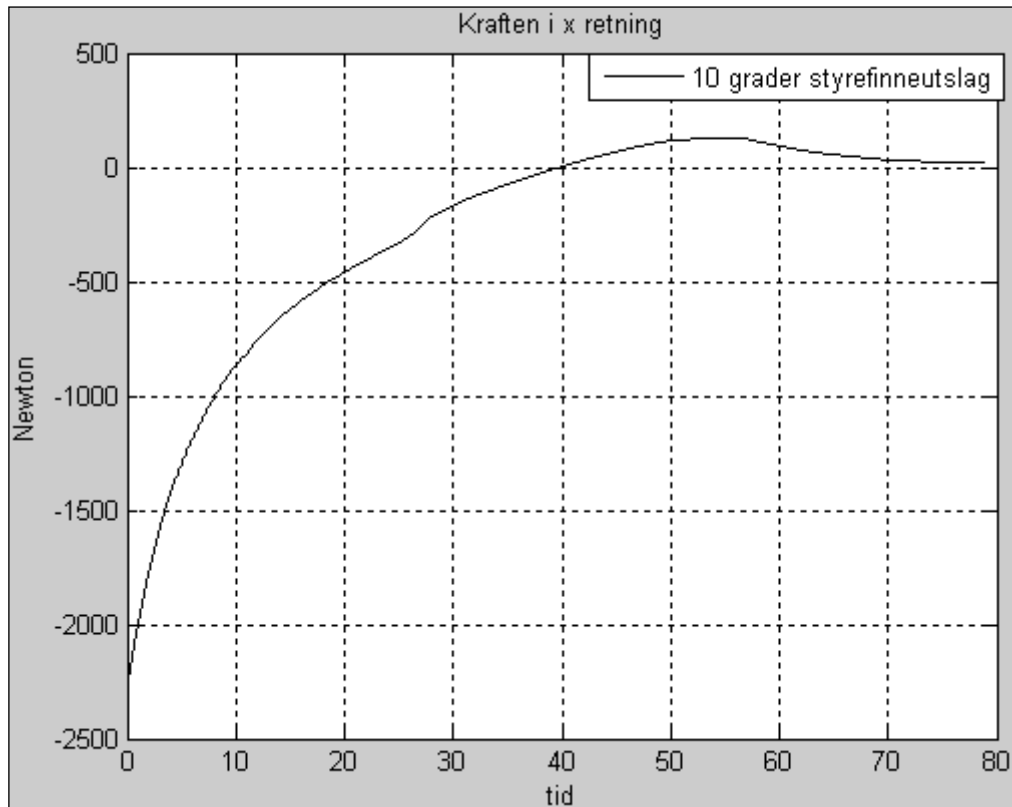
Figur 6.16 Vinkelhastigheten r .

Da prosjektilet begynner å presesere og nutere vises dette godt på disse to vinkelhastighetene. De kraftige svingningene som vises på plottene, er skapt av nutasjonsbevegelsen. Nutasjonsbevegelsen er den raskeste av presesjons- og nutasjonsbevegelsene. Hvis man ønsker et bilde av hvordan presesjonsbevegelsen påvirker disse to vinkelhastighetene, vil omhyllingskurvene på plottene gi en indikasjon på det.

6.10 Krefter på kroppen

Det vi ser på, er krefter på kroppen til prosjektilet. Kraftene langs prosjektilets y -akse og z -akse varierer med spinnnet til kroppen. Men, siden de blir representert i en ikke-rullende ramme, unngår vi å se på denne variasjonen. Dette blir som å se på kreftene til $\{n\}$ -systemet og det blir betraktelig mer lettleset plot.

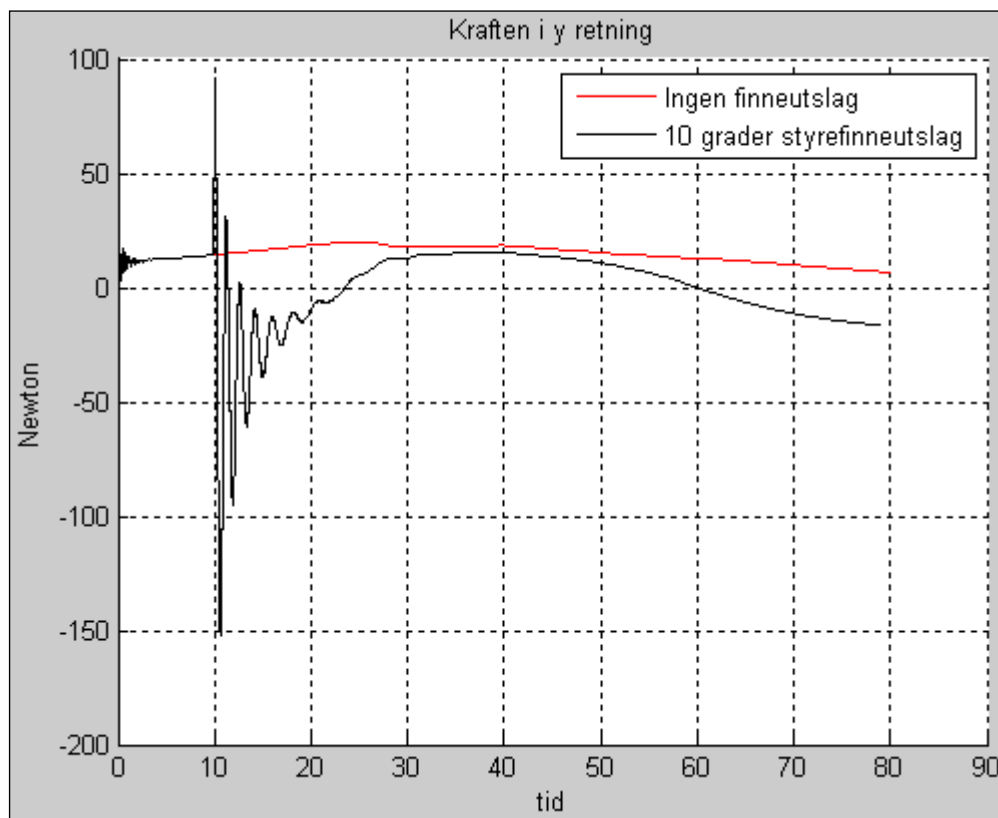
6.10.1 Krefter langs x-aksen representert i ikke-rullsystemet



Figur 6.17 Kraft i x-retning.

Under denne testen nådde prosjektilet toppunktet på turen etter ca 35 sekunder. Av figuren ser vi at kreftene er lik null etter 40 sekunder. Det er fordi da peker nesen til prosjektilet (x-aksen til {b}) litt nedover, og gravitasjonskraften og de aerodynamiske kreftene langs x-aksen utlikner hverandre.

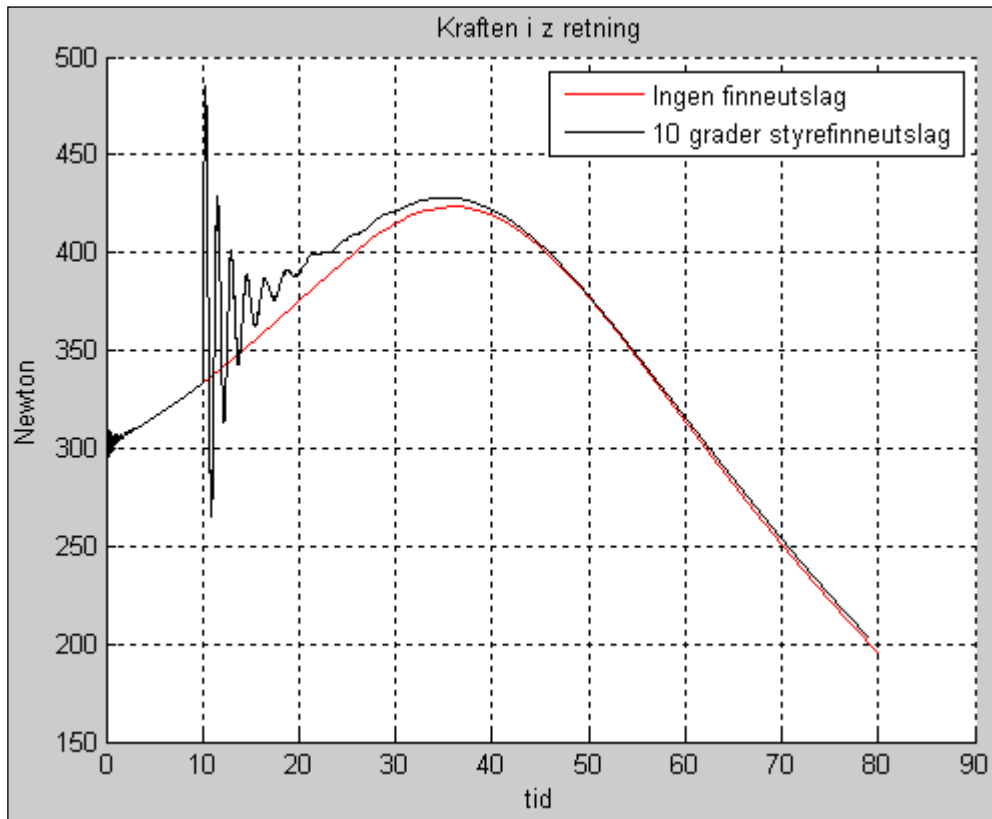
6.10.2 Krefter langs y-aksen representert i ikke-rullsystemet



Figur 6.18 Kraft i y retning.

Når det her blir et finneutslag, vil dette føre til at finnene gir en kraft i positiv y-retning allikevel ser man på plottet at den totale kraften blir negativ i forhold til den som er uten finneutslag. Dette er fordi at prosjektilet prøver å nulle ut momentet som finnene skaper ved å legge seg i motsatt retning av den retningen finnene prøver å dra den.

6.10.3 Krefter langs z-aksen representert i ikke-rullsystemet



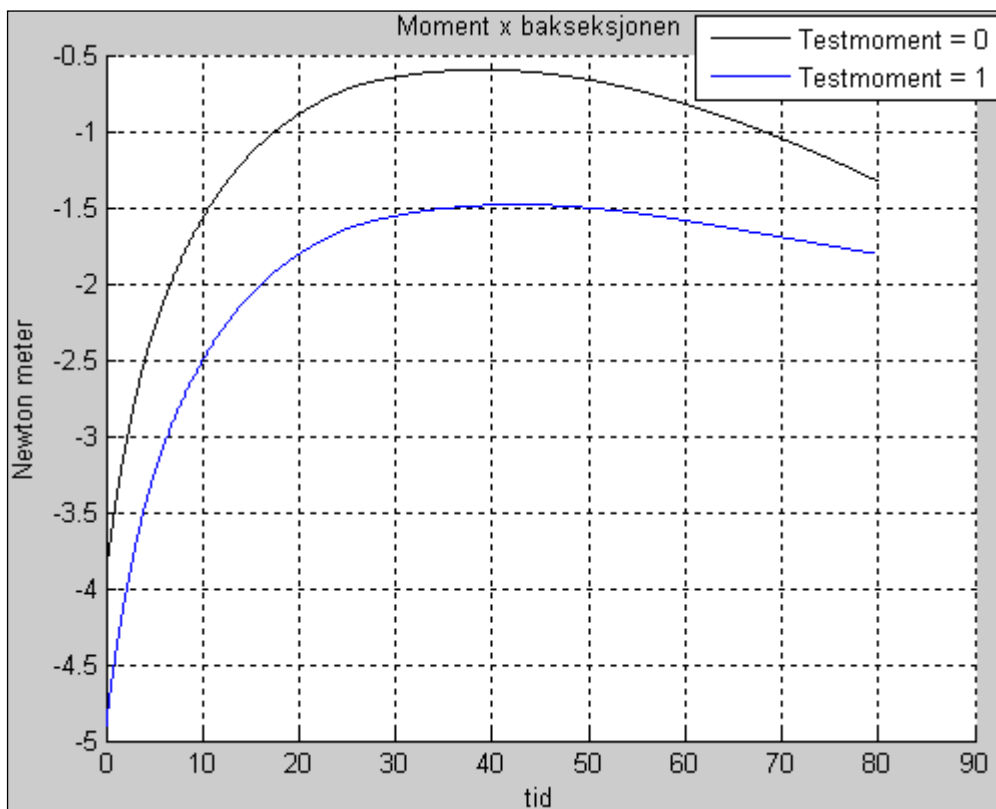
Figur 6.19 Kraft i z-retning.

Som man ser av figuren er kraften i z-retningen på topp når det har gått ca 35 sekunder. Da er prosjektilet på toppen av banen og ligger så å si parallelt med bakken. Da virker hele gravitasjonskraften langs denne akse.

6.11 Momentene på kroppen

Det blir her bare vist momentet om aksene til bakseksjonen. Momentet om x-aksen til forseksjonen er forøvrig lik 0.

6.11.1 Momentet om x-aksen



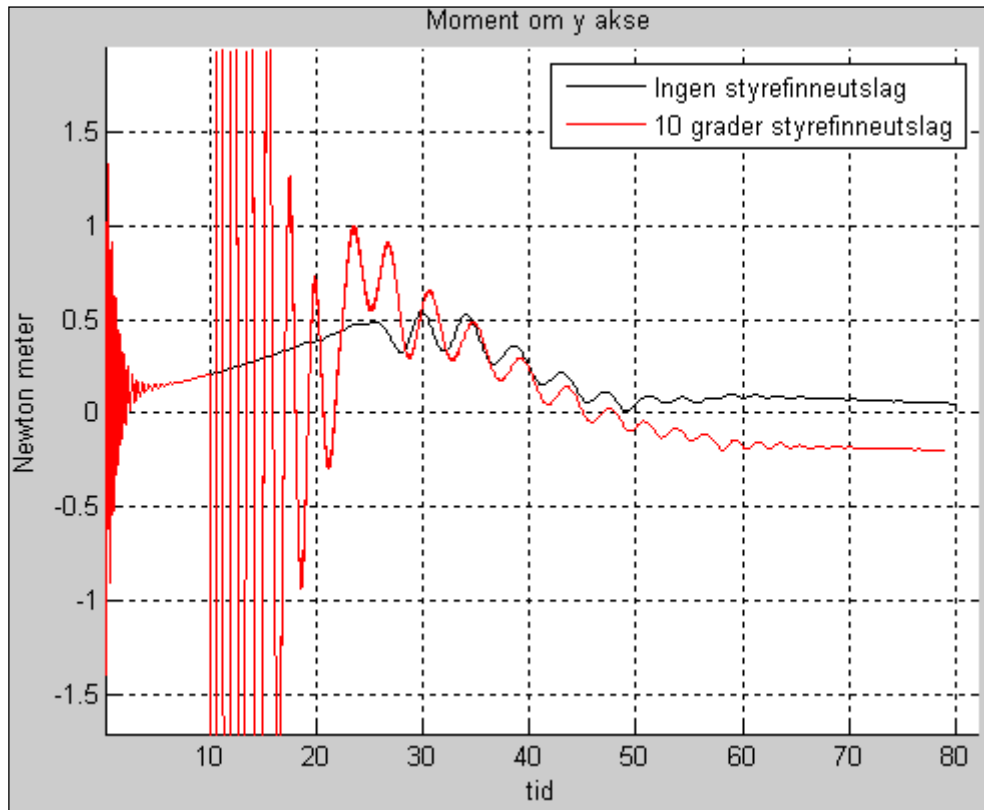
Figur 6.20 Moment om x-aksen til bakseksjonen.

Her ser man forskjellen på momentpåvirkningen til bakseksjonen når man har et testmoment mellom for- og bakseksjonen og når man ikke har det. Hvis man snur på likning (3.54), ser man at momentet er den deriverte av spinnnet.

$$l_A = I_{XX}^A \dot{p}_A \quad (0.152)$$

Momentet l_A består av testmomentet M_T og rullmomentet C_X^{MR} . Prosjektilet når toppen av banen etter cirka 35 sekunder. Da er farten på det laveste. Når prosjektilet skytes ut, er farten på det høyeste. Man kan se av figuren at momentet er en funksjon av hastigheten.

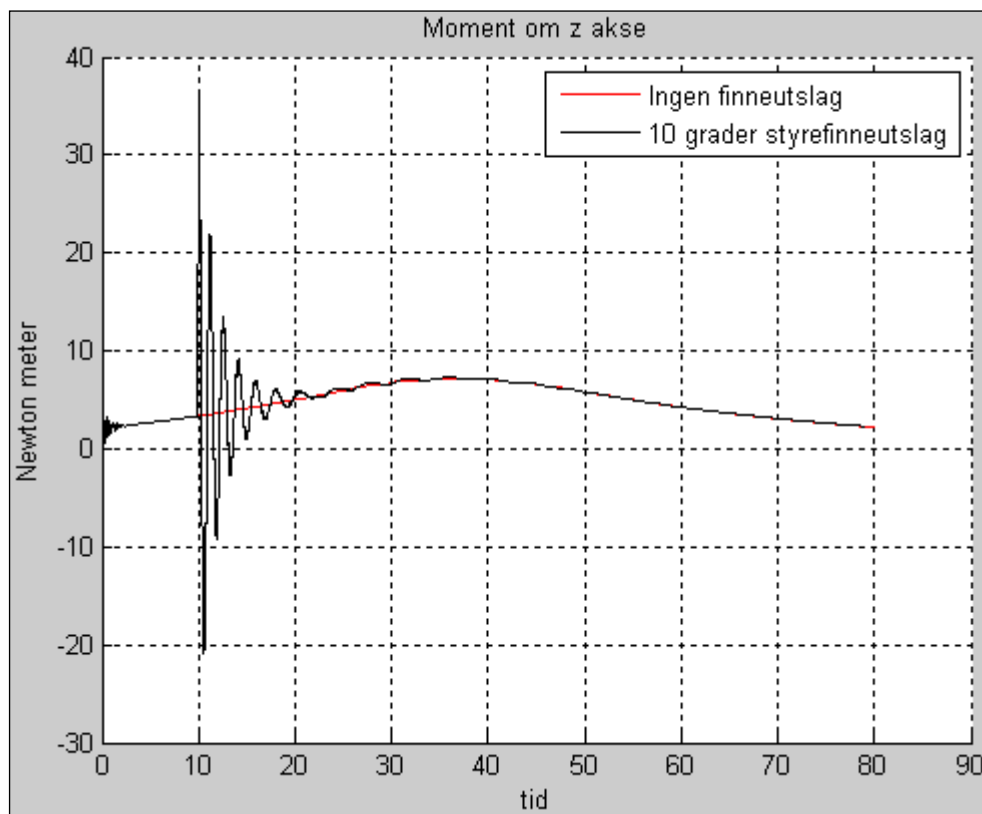
6.11.2 Moment om y-aksen



Figur 6.21 Moment om y aksen.

Et finneutslag som dette gir et ekstra positivt moment om z-aksen. Dette fører til at prosjektilet stiller seg opp mot vinden på en sãnn måte at dette ekstra momentet blir nullstilt. Måten dette påvirker momentet om y-aksen kan man se på denne figuren.

6.11.3 Moment om z-aksen



Figur 6.22 Moment om z akse.

Styrefinnene får et finneutslag etter 10 sekunder. Så blir det noen oscillasjoner som etter hvert stabiliserer seg. Når oscillasjonene har stabilisert seg, er momentet om z-aksen på samme kurve som om det ikke skulle vært noen finneutslag der. Styrefinnene skaper et ekstra moment om z-aksen, men siden momentet blir det samme som uten finner, betyr det at prosjektilet snur seg i forhold til vinden slik at det ekstra momentet fra styrefinnene opphører.

7 Konklusjon

I [2] hadde FFI de ferdige bevegelseslikningene for et spinnende prosjektil. For å få en mest mulig helhetlig og god dokumentasjon ble de utledet her.

Koeffisientene for rullagrene i [2] ga så mye moment mellom for- og bakseksjonen at prosjektilet ble ustabil. Tester som er gjort her, viser at prosjektilet bare tåler et moment på maks 1.33 Nm mellom for- og bakseksjonen før det blir ustabil, mens koeffisientene for rullagrene [2] kunne gi et moment på 50 Nm. I denne rapporten er disse koeffisientene byttet ut med en likning for moment som er hentet fra [4], men under test ble dette momentet for enkelhetsskyld erstattet med et testmoment. I tillegg er det utledet en likning for hvilken momentpåvirkning mellom for- og bakseksjonen et eventuelt energiuttak ville gi. Det ble gjort tester på toleransen til finnestørrelsene. Simuleringer viser at finnene er godt innenfor toleransegrensen, og det anbefales derfor å bruke de opprinnelige størrelsene.

Rapporten inneholder et forslag for å beregne kulelagermomentet og eventuelt generatormoment, og det anbefales at disse likningene blir verifisert, testet og eventuelt implementert når man har et reguleringsystem. Langtflygende prosjektiler bruker baseburn. Bruk av baseburn sørger for mindre drag. Effekten av "baseburn" er ikke tatt med og det anbefales å implementere dette også. Videre anbefales det å utvide modellen slik at man får med Coreolis-effekten og jordkrummingen også. På grunn av jordkrummingen kan et skudd på 24 km utgjøre en høydeforskjell på 45 meter. Prosjektilet flyr i en annen retning enn den veien finnene drar det. Dette er forklart noe i kapittel 6.4. Det anbefales å prøve å finne fram til en formel som kan gi størrelsen på hvilevinkelen (yaw of repose) til prosjektilet og samtidig hvilken retning den drar som en funksjon av blant annet finneutslaget. Dette for lettere å kunne forutsi hvilken retning finnene bør dra for å få prosjektilet til å fly i ønsket retning.

Referanser

- [1] Grandum Øyvind, "2D correction fuze for 155mm artillery ammunition", FFI rapport 2005/00048 Restricted, 2005.
- [2] Quassem A. Tareq. Øyvind Grandum, "Bevegelseslikninger for artillerigranat med styreseksjon", FFI rapport upublisert 2009.
- [3] Grandum Øyvind, "Utvikling av en 6-dof fluktsimulator for pilprosjektiler", FFI rapport 98/04903, 1998.
- [4] Svensk kulelagerprodusent: www.skf.com
- [5] Meteorologisk institutt og NRK's værside www.yr.no
- [6] Ajer Halvor & Andresen Nils Ulrik, "Effects of Indirect-Fire Systems on Tactical Targets", FFI presentasjon.
- [7] Hallingstad Oddvar, "Matematisk modellering av dynamiske systemer", UNIK notat, 2008
- [8] "Firing Tables OEF3HB" 2008
- [9] http://en.wikipedia.org/wiki/Speed_of_sound
- [10] Longdon, L.W. "Textbook of ballistics and gunnery (vol 1)" 1987 Her majesty's Stationary Office, London
- [11] Personlig kommunikasjon med Dag Hansen på www.skf.com

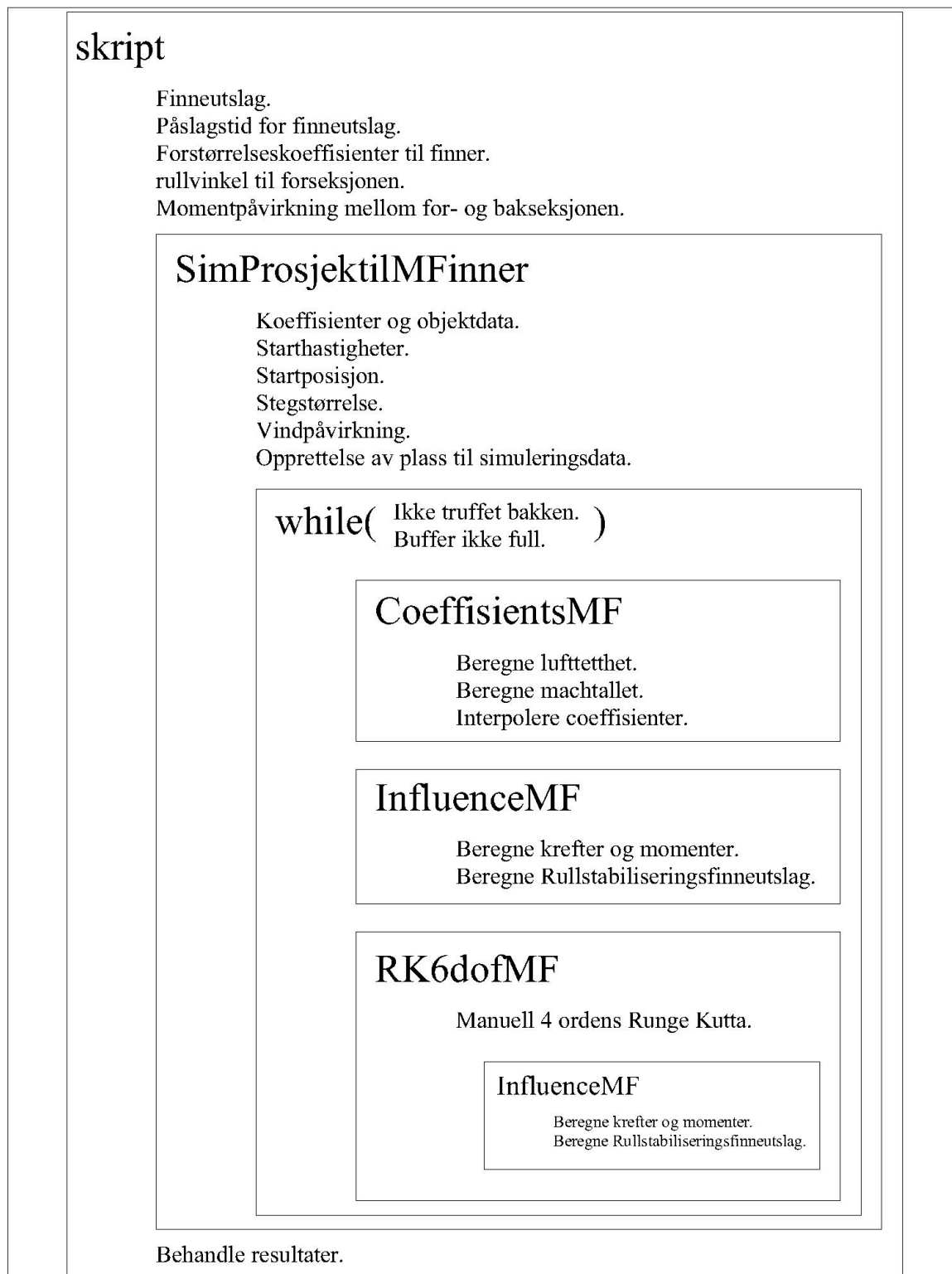
Appendix A Tabell over aerodynamiske koeffisienter

Mach	CX0	CX2	CNa	CNa3	Cma	Cmq	CYpa	Clp	Cmpa
0,01	0,1616	2,32	1,762	21,2	3,923	-13,48	-1,2	-0,03867	-0,39244
0,4	0,1616	2,32	1,762	21,2	3,923	-16,6	-1,2	-0,0335	-0,39244
0,6	0,1616	2,32	1,762	21,2	3,923	-18,2	-1,2	-0,03084	-0,39244
0,7	0,1616	2,32	1,762	21,2	3,923	-19	-1,2	-0,02952	-0,29454
0,75	0,1616	2,32	1,762	21,2	3,923	-19,4	-1,2	-0,02885	-0,24559
0,8	0,1616	2,32	1,762	21,2	3,923	-19,8	-1,2	-0,02819	-0,19664
0,85	0,1616	2,32	1,762	21,2	4,0965	-20,2	-1,2	-0,02753	-0,06164
0,875	0,1616	2,32	1,762	21,2	4,18325	-20,4	-1,2	-0,0272	0,033508
0,9	0,1645	2,32	1,764	21,2	4,27	-20,6	-1,2	-0,02687	0,114576
0,925	0,1694	2,23	1,858	21,2	4,405	-20,8	-1,2	-0,02653	0,240096
0,975	0,2247	2,08	2,036	21,1	4,315	-21,2	-1,2	-0,02587	0,46944
1	0,3090	1,99	2,163	21,1	4,077	-21,4	-1,2	-0,02554	0,519276
1,025	0,3288	4,54	2,225	23,7	3,90355	-21,6	-1,2	-0,02521	0,573904
1,1	0,3314	4,43	2,340	23,6	3,69	-22,2	-1,2	-0,02486	0,666476
1,2	0,3239	4,37	2,396	23,6	3,6	-23	-1,2	-0,0244	0,670778
1,35	0,3106	4,28	2,477	23,5	3,5625	-24,2	-1,2	-0,0237	0,688978
1,75	0,2777	4,07	2,664	23,4	3,5	-25,4	-1,2	-0,02185	0,7045
2	0,2553	3,91	2,822	23,3	3,5	-25,4	-1,2	-0,02089	0,714748
2,25	0,2419	3,77	2,948	23,2	3,5	-25,4	-1,2	-0,02039	0,72202
2,5	0,2287	3,69	3,029	23,2	3,5	-25,4	-1,2	-0,02007	0,725662
2,9	0,2078	3,70	3,008	23,2	3,5	-25,4	-1,2	-0,01956	0,731476

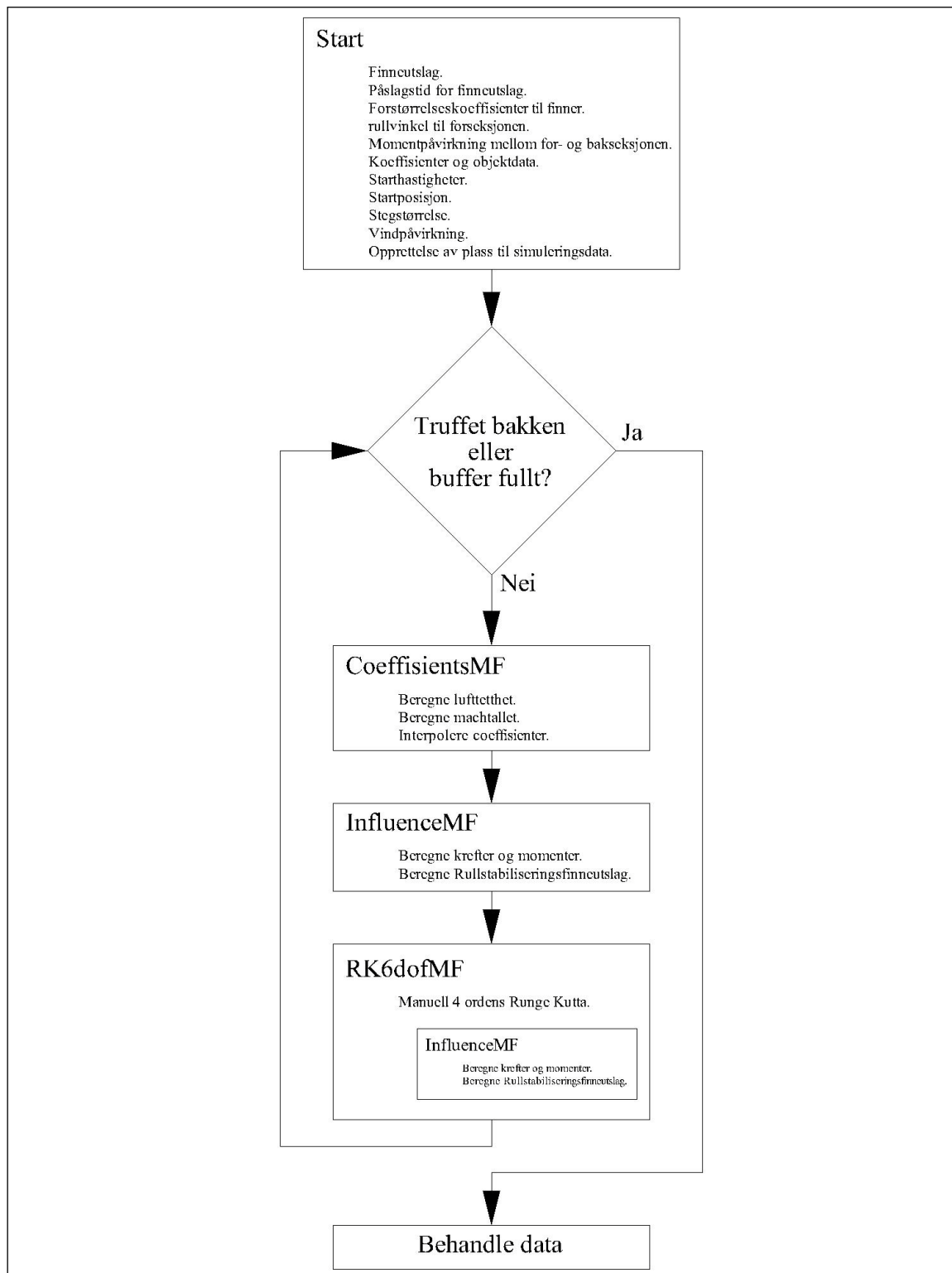
CNaf1	CLaf1	CX0f1	Clpf1	Cldf1	Cnaf2	CLaf2	CX0f2	Clpf2	Cldf2
0,22069	0,21579	0,0049	-0,03955	0	0,073563	0,070463	0,0031	-0,01318	0,0172
0,22069	0,21559	0,0051	-0,03955	0	0,073563	0,070263	0,0033	-0,01318	0,0172
0,22069	0,21549	0,0052	-0,03955	0	0,073563	0,070163	0,0034	-0,01318	0,0172
0,22069	0,21479	0,0059	-0,03955	0	0,073563	0,069863	0,0037	-0,01318	0,0172
0,22069	0,21449	0,0062	-0,03955	0	0,073563	0,069663	0,0039	-0,01318	0,0172
0,22069	0,21409	0,0066	-0,03955	0	0,073563	0,069563	0,004	-0,01318	0,0172
0,227586	0,220486	0,0071	-0,04079	0	0,075862	0,071662	0,0042	-0,0136	0,0178
0,231034	0,223634	0,0074	-0,04141	0	0,077011	0,072711	0,0043	-0,0138	0,0180
0,234483	0,226783	0,0077	-0,04203	0	0,078161	0,073761	0,0044	-0,01401	0,0183
0,241379	0,233479	0,0079	-0,04326	0	0,08046	0,07596	0,0045	-0,01442	0,0188
0,255172	0,246872	0,0083	-0,04573	0	0,085057	0,080457	0,0046	-0,01524	0,0199
0,262069	0,253669	0,0084	-0,04697	0	0,087356	0,082656	0,0047	-0,01566	0,0205
0,275862	0,267262	0,0086	-0,04944	0	0,091954	0,087154	0,0048	-0,01648	0,0215
0,317241	0,308641	0,0086	-0,05686	0	0,105747	0,100647	0,0051	-0,01895	0,0248
0,289655	0,281155	0,0085	-0,05191	0	0,096552	0,091152	0,0054	-0,0173	0,0226
0,248276	0,239476	0,0088	-0,0445	0	0,082759	0,076859	0,0059	-0,01483	0,0194
0,173793	0,165133	0,00866	-0,03115	0	0,057931	0,051831	0,0061	-0,01038	0,0136
0,124138	0,114738	0,0094	-0,02225	0	0,041379	0,034379	0,007	-0,00742	0,0097
0,110345	0,100945	0,0094	-0,01978	0	0,036782	0,029582	0,0072	-0,00659	0,0086
0,096552	0,087052	0,0095	-0,0173	0	0,032184	0,024784	0,0074	-0,00577	0,0075
0,080041	0,070381	0,00966	-0,01435	0	0,02668	0,01888	0,0078	-0,00478	0,0062

Appendix B Dataprogram og filer

For å lettere få en oversikt over strukturen i programmet er figuren under tatt med.



Figuren viser hvordan simuleringsprogrammet er oppbygget og hvor filene som programmet trenger for å kjøre hører hjemme. Programflyten kan man se fra figuren under:



Figur B.1: Programflyt.

Filene er lagt ved slik som de var da dette ble skrevet. Skriptfila blir som regel forandret på etter behov for å få de resultater man var ute etter. I ”InfluenceMF” beregnes for eksempel finneutslaget δ_2 . Det vil si at hvis man ønsker en annen beregning av δ_2 må man inn i denne filen for å forandre på det. Ønsker man for eksempel å legge på vind må man i ”SimProsjekttilMFinner” for å gjøre det.

For behandling av data ble det laget en del andre filer rundt selve dataprogrammet. De viktigste av disse er lagt ved her for helhetens skyld.

B.1 Skript

```
% skript
% addpath('Y:\dp3_2Dkorr\Raymond\Prosjekter\Indirekte
ild\prosjekttil\155mm\matlab\data');
% addpath('Z:\tekst\Prosjekter\Indirekte
ild\prosjekttil\155mm\matlab\data');

% Parametre:
Finneutslag      = 10*pi/180;      % 10 grader.
uk1              = 1;              % Utvidelseskoeffisient til finne 1.
uk2              = 1;              % Utvidelseskoeffisient til finne 2.
TurnOnTime       = [10 20 30 40 50 60]; % Påslagstidspunkt for
finneutslag.
Gamma            = [0:10:350]*pi/180; % Rullvinkelutslag.
MomentVirkning   = 0;              % Momentpåvirkning mellom for og
bakseksjonen.

K = length(TurnOnTime);
L = length(Gamma);

Tabell = zeros(K*L, 4);
n = 1;
tic
for k = 1:1:K
    for l = 1:1:L
        [M0 x y uk2b] = SimProsjekttilMFinner( ...
            Finneutslag, ...
            uk1, ...
            uk2, ...
            TurnOnTime(k), ...
            Gamma(l), ...
            MomentVirkning);

        Tabell(n,1) = x;
        Tabell(n,2) = y;
        Tabell(n,3) = k;
        Tabell(n,4) = l;
        n = n+1;
    end
end

M0.MedFinner     = 1;
M0.Color         = 'k';
M0.Navn          = 'M=0';
```

```

CPUtid = toc;
%
% PlotResults(M0); %, M1, M2, M3, M4, M5, M6);

if(uk2 ~= uk2b)
    disp('Gammel utvidelseskoeffisient for finne 2 er for liten. ');
    disp(['Ny utvidelseskoeffisient bør være større en: ' num2str(uk2b)
        ...
        '. ']);
end

[ping, fs] = wavread('ping.wav');
soundsc(ping, fs);

```

B.2 SimProsjekttilMFinner

```

function [A, xpos, ypos, uk2] = SimProsjekttilMFinner(dlx, uk1, uk2, ...
    Utslagstid, Gamma, M)
% SimProsjekttilMFinner(dl, uk1, , uk2, Utslagstid, Gamma)
% dlx      = styrefinneutslag.
% uk1      = utvidelseskoeffisient til styrefinner.
% uk2      = utvidelseskoeffisient til rullstabiliseringsfinner.
% Utslagstid = det tidspunktet hvor styrefinneutslaget starter.
% Gamma    = vinkelen til forseksjonen.
% M        = momentpåvirkning mellom for og bakseksjonen.

load Konfig_OEF3HB_UF
% load Konfig_data
load EXCELData
% load EXCELDataOld

% Startbetingelser
% A.uk      = uk;
Power      = 100;           % Drar ut 100 watt fra spinnet
V0         = 829;          % Starthastighet [m/s] 829
Asimut     = 0;           % Dreining om z aksen [rad]
Elevasjon  = 45*pi/180;    % Dreining om y aksen [rad]
%Startsteg = 0.0001;      % Steglengde pr iterasjon.
A.N        = 100000;      % antall iterasjoner og
lagrede                                         % variabler 5000
A.mellomsteg = 5;        % Antall mellomsteg for hver
                                           % lagrede variabler.

Spinn      = 2*pi*V0/(TW*Dref);
% Nutasjon = 0;
Steg       = 2*pi/(Spinn*6);   % 8 Tar .. oppdateringer hver
runde

% Vindhastighet i [x y z] retning
Vind = [0; 0; 0];           % m/s
% Stille   0,0-0,2      Røyken stiger rett opp.
% Flau vind 0,3-1,5     En kan se vindretningen av røykens drift.
% Svak vind 1,6-3,3     Bladene på trærne rører seg.
% Lett bris 3,4-5,4     Vinden strekker lette flagg og vimpler.
% Laber bris 5,5-7,9    Vinden løfter støv og løse papirer.
% Frisk bris 8,0-10,7    Småtrær med løv begynner å svaie.
% Liten kuling 10,8-13,8 Store greiner og mindre stammer rører seg.
% Stiv kuling 13,9-17,1 Hele trær rører på seg.
% Sterk kuling 17,2-20,7 Vinden brykker kvister av trærne.
% Liten storm 20,8-24,4 Hele store trær svaier og hiver.

```

```

% Full storm    24,5-28,4    Sjelden inne i landet. Trær rykkes opp med
rot.
% Sterk storm  28,5-32,6    Forekommer sjelden og følges av store
% ødeleggelser.
% Orkan        32,6-        Forekommer meget sjelden. Uvanlig store
% ødeleggelser. (32,6 m/s tilsvarer 117km/h)

A.OBS          = 0;        % forandring til 1 sier at hastigheten er lavere
en
                                     % laveste mach og 2 sier at hastigheten er
høyere
                                     % en høyeste mach

% Lager vektorer som skal bli fylt opp med data fra kjøringen
A.AS          = zeros(A.N,4);
A.EA          = zeros(A.N,4);
A.MO          = zeros(A.N,4);

A.IP          = zeros(A.N,3);
A.BS          = zeros(A.N,3);
A.FO          = zeros(A.N,3);
A.t           = zeros(A.N,1);
A.M           = zeros(A.N,1); % Macknummer

% Setter startbetingelsene inn
A.AS(1,:) = [0, Spinn, 0, 0];          % AngularSpeed
A.EA(1,:) = [Gamma, 0, Elevasjon, Asimut]; % EulerAngles

A.IP(1,:) = zeros(1,3);                % InertialPosition
A.BS(1,:) = [V0, 0, 0];
IP       = A.IP(1,:).';
EA       = A.EA(1,:).';
BS       = A.BS(1,:).';
AS       = A.AS(1,:).';
WS       = Rot(A.EA(1,2), A.EA(1,3), A.EA(1,4)).'*Vind; % vindhastighet i
body systemet
TS = norm(BS-WS); % beregner den totale hastigheten

k=1;
l=1;
mellomfaktor = A.mellomsteg+1;
d1 = 0;
d2 = 0;
tid = 0;

% Debug
A.ypa    = zeros(A.N,1);
A.mpa    = zeros(A.N,1);
A.mq     = zeros(A.N,1);
A.lp     = zeros(A.N,1);
A.Na     = zeros(A.N,1);
A.X0     = zeros(A.N,1);
A.X2     = zeros(A.N,1);
A.ma     = zeros(A.N,1);
A.Laf1   = zeros(A.N,1);
A.Laf2   = zeros(A.N,1);
A.ldf1   = zeros(A.N,1);
A.ldf2   = zeros(A.N,1);
A.lpf1   = zeros(A.N,1);
A.lpf2   = zeros(A.N,1);

```

```

A.Naf1 = zeros(A.N,1);
A.Naf2 = zeros(A.N,1);
A.X0f1 = zeros(A.N,1);
A.X0f2 = zeros(A.N,1);

while((k<A.N*mellomfaktor) && IP(3)<=0)

    % Beregner koeffisienter
    [C Mack] = CoefficientsMF(TS, -IP(3), adat);

    % finneutslag
    if(tid >= Utslagstid) % sekunder
        dl = dlx; % Styrefinner:                radianer
    end

    % Beregner krefter og momenter
    [FO MO uk2] = InfluenceMF(C, TS, BS-WS, EA, AS, m, Dref, Sref, ...
        dl, d2, adat.Rf_fb, Power, uk1, uk2, M);

    % Ittererer seg fram et steg
    [Ret uk2] = RK6dofMF(BS, Vind, TS, AS, IP, EA, Steg, m, I_matr, ...
        C, Dref, Sref, FO, MO, dl, d2, adat.Rf_fb, Ixx_F, Power, uk1,
uk2, M);

    k=k+1;

    tid = tid + Steg;
    % Oppdaterer gamle variabler
    IP = Ret.IP;
    EA = Ret.EA;
    BS = Ret.BS;
    AS = Ret.AS;
    WS = Rot(EA(2), EA(3), EA(4)).'*Vind;
    TS = norm(BS-WS);

    if(rem(k+A.mellomsteg,mellomfaktor) == 0)
        % Lagrer data
        A.FO(1,:) = FO;
        A.MO(1,:) = MO;
        A.M(1,1) = Mack;

        % debug
        A.ypa(1,1) = C.ypa;
        A.mpa(1,1) = C.mpa;
        A.mq(1,1) = C.mq;
        A.lp(1,1) = C.lp;
        A.Na(1,1) = C.Na;
        A.X0(1,1) = C.X0;
        A.X2(1,1) = C.X2;
        A.ma(1,1) = C.ma;
        A.Laf1(1,1) = C.Laf1;
        A.Laf2(1,1) = C.Laf2;
        A.ldf1(1,1) = C.ldf1;
        A.ldf2(1,1) = C.ldf2;
        A.lpf1(1,1) = C.lpf1;
        A.lpf2(1,1) = C.lpf2;
        A.Naf1(1,1) = C.Naf1;
        A.Naf2(1,1) = C.Naf2;
    end
end

```

```

A.X0f1(1,1) = C.X0f1;
A.X0f2(1,1) = C.X0f2;

l          = l+1;

A.IP(1,:) = IP.';
A.EA(1,:) = EA.';
A.BS(1,:) = BS.';
A.AS(1,:) = AS.';

A.t(1,1) = tid;

% Forandrer steglengden
Steg = 2*pi/(AS(2)*6);
% Steg = 0.0002;
end

```

end

```

A.N = k;
A.L = l;
A.mellomsteg = A.mellomsteg-1;
xpos = IP(1);
ypos = IP(2);

```

end

B.3 CoeffisientsMF

```
function [C M] = CoefficientsMF(V, h, adat)
```

```

% % Machtallet
% T = 288.15-0.0065*h;
% M = V/(20.048*sqrt(T));
% l = find(adat.Mach>M, 1, 'first');
% k = l-1;
%
% % Lufttettheten
% C.rho = 1.225*(T/288.16)^4.2561;
T = 288.15-6.5e-3*h;
M = V/sqrt(1.4*287.1*T);
l = find(adat.Mach>M, 1, 'first');
k = l-1;

```

```
C.rho = 1.225*(T/288.15)^4.2595;
```

```

% Interpolerer
if(isempty(l))
    % Hastigheten er høyere en det høyeste machtall og i mangelen
    % på noe bedre å gjøre velges det å bruke det høyeste machtall.
    Lengde = length(adat.Cypa);
    C.ypa = adat.Cypa(Lengde);
    C.mpa = adat.Cmpa(Lengde);
    C.mq = adat.Cmq(Lengde);
    C.lp = adat.Clp(Lengde);
    C.Na = adat.CNa(Lengde);
    C.X0 = adat.CX0(Lengde);

```

```

C.X2 = adat.CX2(Lengde);
C.ma = adat.Cma(Lengde);
C.OBS = 2;
% Ekstra på grunn av finner
C.Laf1 = adat.CLaf1(Lengde);
C.Laf2 = adat.CLaf2(Lengde);
C.ldf1 = adat.Cldf1(Lengde);
C.ldf2 = adat.Cldf2(Lengde);
C.lpf1 = adat.Clpf1(Lengde);
C.lpf2 = adat.Clpf2(Lengde);
C.Naf1 = adat.CNaf1(Lengde);
C.Naf2 = adat.CNaf2(Lengde);
C.X0f1 = adat.CX0f1(Lengde);
C.X0f2 = adat.CX0f2(Lengde);
elseif(l~=1)
% Hastigheten ligger innenfor machtallet og det interpoleres derfor
% mellom de to nærmeste mach tallene.
x = (M-adat.Mach(k))/(adat.Mach(1)-adat.Mach(k));

C.ypa = adat.Cypa(k) + (adat.Cypa(1)-adat.Cypa(k))*x;
C.mpa = adat.Cmpa(k) + (adat.Cmpa(1)-adat.Cmpa(k))*x;
C.mq = adat.Cmq(k) + (adat.Cmq(1)-adat.Cmq(k))*x;
C.lp = adat.Clp(k) + (adat.Clp(1)-adat.Clp(k))*x;
C.Na = adat.CNa(k) + (adat.CNa(1)-adat.CNa(k))*x;
C.X0 = adat.CX0(k) + (adat.CX0(1)-adat.CX0(k))*x;
C.X2 = adat.CX2(k) + (adat.CX2(1)-adat.CX2(k))*x;
C.ma = adat.Cma(k) + (adat.Cma(1)-adat.Cma(k))*x;
C.OBS = 0;
% Ekstra på grunn av finner
C.Laf1 = adat.CLaf1(k) + (adat.CLaf1(1)-adat.CLaf1(k))*x;
C.Laf2 = adat.CLaf2(k) + (adat.CLaf2(1)-adat.CLaf2(k))*x;
C.ldf1 = adat.Cldf1(k) + (adat.Cldf1(1)-adat.Cldf1(k))*x;
C.ldf2 = adat.Cldf2(k) + (adat.Cldf2(1)-adat.Cldf2(k))*x;
C.lpf1 = adat.Clpf1(k) + (adat.Clpf1(1)-adat.Clpf1(k))*x;
C.lpf2 = adat.Clpf2(k) + (adat.Clpf2(1)-adat.Clpf2(k))*x;
C.Naf1 = adat.CNaf1(k) + (adat.CNaf1(1)-adat.CNaf1(k))*x;
C.Naf2 = adat.CNaf2(k) + (adat.CNaf2(1)-adat.CNaf2(k))*x;
C.X0f1 = adat.CX0f1(k) + (adat.CX0f1(1)-adat.CX0f1(k))*x;
C.X0f2 = adat.CX0f2(k) + (adat.CX0f2(1)-adat.CX0f2(k))*x;
else
% Hastigheten er lavere en det laveste machtallet og i mangelen
% på noe bedre å gjøre velges det å bruke det laveste mach tall.
C.ypa = adat.Cypa(1);
C.mpa = adat.Cmpa(1);
C.mq = adat.Cmq(1);
C.lp = adat.Clp(1);
C.Na = adat.CNa(1);
C.X0 = adat.CX0(1);
C.X2 = adat.CX2(1);
C.ma = adat.Cma(1);
C.OBS = 1;
% Ekstra på grunn av finner
C.Laf1 = adat.CLaf1(1);
C.Laf2 = adat.CLaf2(1);
C.ldf1 = adat.Cldf1(1);
C.ldf2 = adat.Cldf2(1);
C.lpf1 = adat.Clpf1(1);
C.lpf2 = adat.Clpf2(1);
C.Naf1 = adat.CNaf1(1);
C.Naf2 = adat.CNaf2(1);
C.X0f1 = adat.CX0f1(1);

```

```

    C.X0f2 = adat.CX0f2(1);
end

end

```

B.4 InfluenceMF

```

function [FO MO uk2] = InfluenceMF(C, V, BS, EA, AS, m, D, S, d1, d2,
...
    CGCP, Power, uk1, uk2, M)
% InfluenceMF(C, TS, BS+BSK1/2-WS, EA+EAK1/2, AS+ASK1/2, m, D, S);

% cpsi = cos(EA(4));
% spsi = sin(EA(4));
% cteh = cos(EA(3)) ;
% steh = sin(EA(3));

%% Bakseksjonen
% Magnus koeffisienter
MAG = AS(2)*D/(2*V); % brukes under alle magnus koeffisienter
CFmagY = C.ypa*MAG*BS(3)/V;
% CFmagYT = -C.ypa*MAG*sie*sifi;
CFmagZ = -C.ypa*MAG*BS(2)/V;
% CFmagZT = C.ypa*MAG*sie*cofi;
% DONE

CMmagM = C.mpa*MAG*BS(2)/V;
% CMmagMT = -C.mpa*MAG*sie*cofi;
CMmagN = C.mpa*MAG*BS(3)/V;
% CMmagNT = -C.mpa*MAG*sie*sifi;

% Pitch rull- og dempemoment koeffisienter
CMrullL = C.lp*AS(2)*D/(2*V);
CMpitM = C.mq*AS(3)*D/(2*V);
CMpitN = C.mq*AS(4)*D/(2*V);

% Normal og aksialkraft koeffisienter
% CFaksA = -(C.X0+C.X2*(BS(2)^2+BS(3)^2)/V^2)*0.8;
% CFaksF = -(C.X0+C.X2*(BS(2)^2+BS(3)^2)/V^2)*0.2;
CFaksA = -C.X0*0.8;
CFaksF = -C.X0*0.2;

% CFaksXT = -CX0A-C.X2*V*sie^2;
CFnorY = -C.Na*BS(2)/V;
% CFnorYT = C.Na*sie*cofi;
CFnorZ = -C.Na*BS(3)/V;
% CFnorZT = C.Na*sie*sifi;

% Tippmomentet
CMtipM = C.ma*BS(3)/V;
% CMtipMT = -C.ma*sie*sifi;
CMtipN = -C.ma*BS(2)/V;
% CMtipNT = C.ma*sie*cofi;

%% Forseksjonen styrefinner

```



```

% Normalvektoren til styrefinnene
x1 = -sin(d1);
y1 = cos(d1)*cos(EA(1));
z1 = cos(d1)*sin(EA(1));
nflb = [x1; y1; z1];

% Normalkraft/moment finne 1
sieps = (Kryss(AS(2:4), [CGCP; 0; 0]).'*nflb - BS.*nflb)/V;
CFF1N = C.Naf1*uk1*sieps*nflb;

% Aksialkraft. finne 1
CFF1A = C.X0f1*uk1*((nflb.*BS)*nflb-BS)/V;

% Spindndempingskoeffisient finne 1
CMF1D = C.lpf1*uk1*AS(1)*D/(2*V);
CMF1N = Kryss([CGCP; 0; 0], CFF1N+CFF1A)/D;

%% Forseksjonen rullstabiliseringsfinner CFF2 = CFF2A + CFF2N

% Test av finner: Må finne et finneutslag som gir totale momenter i x
% retning på forseksjonen lik 0.
d2 = -2*M/(C.rho*V^2*S*D*C.ldf2*uk2);
if(d2>10)
    uk2 = -2*M/(C.rho*V^2*S*D*C.ldf2*10);
    d2=10;
elseif(d2<-10)
    uk2 = -2*M/(C.rho*V^2*S*D*C.ldf2*-10);
    d2=-10;
end

CFF2 = [(-C.Naf2*uk2*sin(d2)^2-C.X0f2*uk2*cos(d2)^2)*BS(1); ...
-
(C.X0f2*uk2+C.Laf2*uk2*cos(d2)^2*sin(EA(1))^2)*BS(2)+C.Laf2*uk2*cos(d2)^
2*sin(2*EA(1))*BS(3)/2 + ...
C.Naf2*uk2*CGCP*cos(d2)^2*(sin(2*EA(1))*AS(3)/2+sin(EA(1))^2*AS(4));
...
C.Laf2*uk2*cos(d2)^2*sin(2*EA(1))*BS(2)/2-
(C.X0f2*uk2+C.Laf2*uk2*cos(d2)^2*cos(EA(1))^2)*BS(3) - ...
C.Naf2*uk2*CGCP*cos(d2)^2*(cos(EA(1))^2*AS(3)-
sin(2*EA(1))*AS(4)/2)]/V;

% Spindndempingskoeffisient finne 2
CMF2D = C.lpf2*uk2*AS(1)*D/(2*V)+C.ldf2*uk2*d2;

% moment finne 2
CMF2T = Kryss([CGCP; 0; 0], CFF2)/D;

%% Totale krefter, kulelagermomentet og generatormomentet
% Aerodynamiske krefter fra kroppen
FA = C.rho*V^2*S/2*[CFaksA; (CFmagY+CFnorY); (CFmagZ+CFnorZ)];

% Aerodynamiske krefter fra finner og framseksjonen [3x1]
FF = C.rho*V^2*D^2*pi/8*([CFaksF; 0; 0]+CFF1N+CFF1A+CFF2);
% Gravitasjonskraften
% R = [cpsi*cteh,      -spsi,  cpsi*steh; ...
%       spsi*cteh,  cpsi,    spsi*steh; ...
%       -steh,      0,      cteh];

```

```

% FG = m*R.'*[0; 0; 9.8185037];
FG = m*9.8185037*[-sin(EA(3)); 0; cos(EA(3))];

% % Kulelagermomentet (virker langs x akse)
% ML = 0.002*((FF-FA).'(FF-FA))^(1/2)*0.02/2; <- ordne senere.
% % Generatormomentet
% MG = Power/((AS(2)-AS(1))^2)^(1/2);

% oppsummert
FO = FA+FG+FF;

%% Totale momentene
% Aerodynamiske momenter på bakseksjonen
CMKropp = [0; CMrull; (CMmagM+CMpitM+CMtipM); (CMmagN+CMpitN+CMtipN)];
% Aerodynamiske momenter på framseksjonen [4x1]
CMStyr = [CMF1D+CMF2D; 0; CMF1N(2:3)+CMF2T(2:3)];
% oppsummert
if(AS(2)>0)
    MO = C.rho*V^2*(D)^3*pi/8*(CMKropp+CMStyr)+[M;-M;0;0];
else
    MO = C.rho*V^2*(D)^3*pi/8*(CMKropp+CMStyr);
end

%%
end
%%

% function R = Rot(ea)
% cpsi = cos(ea(4));
% spsi = sin(ea(4));
% cteh = cos(ea(3));
% steh = sin(ea(3));
% R = [cpsi*cteh,    -spsi,  cpsi*steh; ...
%      spsi*cteh,   cpsi,   spsi*steh; ...
%      -steh,      0,      cteh];
% end

%%
function X = Kryss(V1, V2)
X = [V1(2)*V2(3)-V1(3)*V2(2); ...
     V1(3)*V2(1)-V1(1)*V2(3); ...
     V1(1)*V2(2)-V1(2)*V2(1)];
end

%%

B.5 RK6dofMF

function [Ret uk2] = RK6dofMF(BS, Vind, TS, AS, IP, EA, Steg, m, I_matr,
...
    C, D, S, FO, MO, d1, d2, CGCP, Ixx_F, Power, uk1, uk2, M)

%% Runge Kutta steg 1

% [FO MO uk2] = InfluenceMF(C, V, BS, EA, AS, m, D, S, d1, d2, ...
% CGCP, Power, uk1, uk2, M);

```

```

IPK1 = Steg*IPF(EA, BS);
EAK1 = Steg*EAF(EA, AS);
BSK1 = Steg*BSF(FO, AS, BS, EA, m);
ASK1 = Steg*ASF(MO, I_matr, Ixx_F, AS, EA);

%% Runge Kutta steg 2
% Ny retning for vinden er
WS = Rot(EA(2:4)+EAK1(2:4)/2).'*Vind;
[FO MO uk2] = InfluenceMF(C, TS, BS+BSK1/2-WS, EA+EAK1/2, AS+ASK1/2, ...
    m, D, S, dl, d2, CGCP, Power, uk1, uk2, M);

IPK2 = Steg*IPF(EA+EAK1/2, BS+BSK1/2);
EAK2 = Steg*EAF(EA+EAK1/2, AS+ASK1/2);
BSK2 = Steg*BSF(FO, AS+ASK1/2, BS+BSK1/2, EA+EAK1/2, m);
ASK2 = Steg*ASF(MO, I_matr, Ixx_F, AS+ASK1/2, EA+EAK1/2);

%% Runge Kutta steg 3
% Ny retning for vinden er
WS = Rot(EA(2:4)+EAK2(2:4)/2).'*Vind;
[FO MO uk2] = InfluenceMF(C, TS, BS+BSK2/2-WS, EA+EAK2/2, AS+ASK2/2, ...
    m, D, S, dl, d2, CGCP, Power, uk1, uk2, M);

IPK3 = Steg*IPF(EA+EAK2/2, BS+BSK2/2);
EAK3 = Steg*EAF(EA+EAK2/2, AS+ASK2/2);
BSK3 = Steg*BSF(FO, AS+ASK2/2, BS+BSK2/2, EA+EAK2/2, m);
ASK3 = Steg*ASF(MO, I_matr, Ixx_F, AS+ASK2/2, EA+EAK2/2);

%% Runge Kutta steg 4
% Ny retning for vinden er
WS = Rot(EA(2:4)+EAK3(2:4)).'*Vind;
[FO MO uk2] = InfluenceMF(C, TS, BS+BSK3-WS, EA+EAK3, AS+ASK3, m, D, S,
    ...
    dl, d2, CGCP, Power, uk1, uk2, M);

IPK4 = Steg*IPF(EA+EAK3, BS+BSK3);
EAK4 = Steg*EAF(EA+EAK3, AS+ASK3);
BSK4 = Steg*BSF(FO, AS+ASK3, BS+BSK3, EA+EAK3, m);
ASK4 = Steg*ASF(MO, I_matr, Ixx_F, AS+ASK3, EA+EAK3);

%% Runge Kutta oppsummering

Ret.BS = BS+(BSK1+2*BSK2+2*BSK3+BSK4)/6; % [3x1]
Ret.IP = IP+(IPK1+2*IPK2+2*IPK3+IPK4)/6; % [3x1]

Ret.AS = AS+(ASK1+2*ASK2+2*ASK3+ASK4)/6; % [4x1]
Ret.EA = EA+(EAK1+2*EAK2+2*EAK3+EAK4)/6; % [4x1]
end
%%
function R = IPF(ea, bs)
R(1,1) = cos(ea(4))*cos(ea(3))*bs(1)-
sin(ea(4))*bs(2)+cos(ea(4))*sin(ea(3))*bs(3);
R(2,1) =
sin(ea(4))*cos(ea(3))*bs(1)+cos(ea(4))*bs(2)+sin(ea(4))*sin(ea(3))*bs(3)
;
R(3,1) = -sin(ea(3))*bs(1)+cos(ea(3))*bs(3);
end
%%
function R = EAF(ea, as)

```

```

R(1,1) = as(1)+as(4)*tan(ea(3));
R(2,1) = as(2)+as(4)*tan(ea(3));
R(3,1) = as(3);
R(4,1) = as(4)/cos(ea(3));
end

%%
function R = BSF(fo, as, bs, ea, m)
R(1,1) = fo(1)/m-as(3)*bs(3)+as(4)*bs(2);
R(2,1) = fo(2)/m-bs(1)*as(4)-bs(3)*as(4)*tan(ea(3));
R(3,1) = fo(3)/m+bs(2)*as(4)*tan(ea(3))+bs(1)*as(3);
end

%%
function R = ASF(mo, I, If, as, ea)
R(1,1) = mo(1)/If;
R(2,1) = mo(2)/I(1,1);
R(3,1) = (mo(3)-(I(1,1)*as(2)+If*as(1))*as(4)-
I(3,3)*as(4)^2*tan(ea(3)))/I(2,2);
R(4,1) =
(mo(4)+(I(1,1)*as(2)+If*as(1))*as(3)+I(2,2)*as(4)*as(3)*tan(ea(3)))/I(3,
3);
end

%%
function R = Rot(ea)
cpsi = cos(ea(3));
spsi = sin(ea(3));
cteh = cos(ea(2)) ;
steh = sin(ea(2));
R = [cpsi*cteh,    -spsi, cpsi*steh; ...
      spsi*cteh, cpsi,   spsi*steh; ...
      -steh,     0,     cteh];
end

```

B.6 PlotResults

```

function PlotResults(varargin)

    nvar = nargin;

    %% Plotter spinnet bak i forhold til tiden
    SpinnBakFig = figure;
    if(nvar > 1)
        hold on
    end

    LegTekst = '';
    for k=1:nvar
        x = varargin.t(1:varargin{k}.L);
        if(varargin{k}.MedFinner == 1)
            y = varargin{k}.AS(1:varargin{k}.L, 2);
        else
            y = varargin{k}.AS(1:varargin{k}.L, 1);
        end
        plot(x, y, varargin{k}.Color)
        if(k<nvar)
            LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ' ' , ' '];
        else

```

```

        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Spinnet bakseksjonen');
xlabel('tiden');
ylabel('rad/sek');
set(gcf, 'Name', 'Spinnet bakseksjonen');
grid

%% Plotter spinnet forran i forhold til tiden
SpinnForFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    if(varargin{k}.MedFinner == 1)
        x = varargin{k}.t(1:varargin{k}.L);
        y = varargin{k}.AS(1:varargin{k}.L, 1);
        plot(x, y, varargin{k}.Color)
        if(k<nvar)
            LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
        else
            LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
        end
    end
end
eval(['legend(' LegTekst ')']);

title('Spinnet forseksjonen');
xlabel('tid');
ylabel('rad/sek');
set(gcf, 'Name', 'Spinnet forseksjonen');
grid
%%

%% Plotter posisjonen y i forhold til x
PosXYFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    x = varargin{k}.IP(1:varargin{k}.L, 1);
    y = varargin{k}.IP(1:varargin{k}.L, 2);
    plot(x, y, varargin{k}.Color)

    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

```

```

title('Posisjonen X,Y');
xlabel('posisjonen i x retning');
ylabel('posisjonen i y retning');
set(gcf, 'Name', 'Posisjonen X,Y');
grid

%% Plotter posisjonene x i forhold til z
PosXZFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    x = varargin{k}.IP(1:varargin{k}.L, 1);
    y = varargin{k}.IP(1:varargin{k}.L, 3)*-1;
    plot(x, y, varargin{k}.Color)

    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Posisjonen X,Z');
xlabel('posisjonen i x retning');
ylabel('posisjonen i z retning');
set(gcf, 'Name', 'Posisjonen X,Z');
grid

%% Plotter fi for forseksjonen
FiForFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    if(varargin{k}.MedFinner == 1)
        % x = varargin{k}.IP(1:varargin{k}.L, 1);
        x = varargin{k}.t(1:varargin{k}.L);
        y = varargin{k}.EA(1:varargin{k}.L, 1);
        plot(x, y, varargin{k}.Color)
        if(k<nvar)
            LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
        else
            LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
        end
    end
end

end
eval(['legend(' LegTekst ')']);

title('fi forseksjonen');
xlabel('tid');
ylabel('rad');

```

```

set(gcf, 'Name', 'fi forseksjonen');
grid

%% Plotter theta
ThetaFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.EA(1:varargin{k}.L, 3);
    else
        y = varargin{k}.EA(1:varargin{k}.L, 2);
    end
    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('theta');
xlabel('tid');
ylabel('rad');
set(gcf, 'Name', 'theta');
grid

%% Plotter psi
PsiFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.EA(1:varargin{k}.L, 4);
    else
        y = varargin{k}.EA(1:varargin{k}.L, 3);
    end
    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('psi');
xlabel('tid');
ylabel('rad');

```

```

set(gcf, 'Name', 'psi');
grid

%% Plotter vinkelhastigheten q forhold til tiden
QFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.AS(1:varargin{k}.L, 3);
    else
        y = varargin{k}.AS(1:varargin{k}.L, 2);
    end
    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ''];
    end
end
eval(['legend(' LegTekst ')']);

title('Vinkelhastigheten q');
xlabel('tid');
ylabel('rad/sek');
set(gcf, 'Name', 'Vinkelhastigheten q');
grid

%% Plotter vinkelhastigheten r forhold til tiden
RFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.AS(1:varargin{k}.L, 4);
    else
        y = varargin{k}.AS(1:varargin{k}.L, 3);
    end
    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ''];
    end
end
eval(['legend(' LegTekst ')']);

title('Vinkelhastigheten r');
xlabel('tid');
ylabel('rad/sek');

```



```

set(gcf, 'Name', 'Vinkelhastigheten r');
grid

%% Plotter Moment i xf retning forhold til tiden
MXFFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.MO(1:varargin{k}.L-1, 1);
        plot(x, y, varargin{k}.Color)
    end

    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Moment x forseksjon');
xlabel('tid');
ylabel('Newton meter');
set(gcf, 'Name', 'Moment x forseksjon');
grid

%% Plotter Moment i xa retning forhold til tiden
MXAFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.MO(1:varargin{k}.L-1, 2);
    else
        y = varargin{k}.MO(1:varargin{k}.L-1, 1);
    end
    plot(x, y, varargin{k}.Color)

    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Moment x bakseksjonen');
xlabel('tid');

```

```

ylabel('Newton meter');
set(gcf, 'Name', 'Moment x bakseksjonen');
grid

%% Plotter Moment i y retning forhold til tiden
MYFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.MO(1:varargin{k}.L-1, 3);
    else
        y = varargin{k}.MO(1:varargin{k}.L-1, 2);
    end
    plot(x, y, varargin{k}.Color)

    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Moment om y akse');
xlabel('tid');
ylabel('Newton meter');
set(gcf, 'Name', 'Moment om y akse');
grid

%% Plotter Moment i z retning forhold til tiden
MZFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    if(varargin{k}.MedFinner == 1)
        y = varargin{k}.MO(1:varargin{k}.L-1, 4);
    else
        y = varargin{k}.MO(1:varargin{k}.L-1, 3);
    end
    plot(x, y, varargin{k}.Color)

    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);
title('Moment om z akse');

```

```

xlabel('tid');
ylabel('Newton meter');
set(gcf, 'Name', 'Moment om z akse');
grid

%% Plotter alpha i forhold til tiden
AlphaFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    x = varargin{k}.t(1:varargin{k}.L-1);
    u = varargin{k}.BS(1:varargin{k}.L-1, 1);
    w = varargin{k}.BS(1:varargin{k}.L-1, 3);

    alpha = atan(w./u);
    plot(x, alpha, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Alpha');
xlabel('tid');
ylabel('rad');
set(gcf, 'Name', 'Alpha');
grid

%% Plotter beta i forhold til tiden
BetaFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    x = varargin{k}.t(1:varargin{k}.L-1);
    u = varargin{k}.BS(1:varargin{k}.L-1, 1);
    v = varargin{k}.BS(1:varargin{k}.L-1, 2);

    alpha = atan(v./u);
    plot(x, alpha, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Beta');
xlabel('tid');
ylabel('rad');
set(gcf, 'Name', 'Beta');

```

```

grid

%% Plotter Kraft i x retning i forhold til tiden
RFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    y = varargin{k}.FO(1:varargin{k}.L-1, 1);

    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ''];
    end
end
eval(['legend(' LegTekst ')']);

title('Kraften i x retning');
xlabel('tid');
ylabel('Newton');
set(gcf, 'Name', 'Kraften i x retning');
grid

%% Plotter Kraft i y retning i forhold til tiden
RFig = figure;
if(nvar > 1)
    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    y = varargin{k}.FO(1:varargin{k}.L-1, 2);

    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ''];
    end
end
eval(['legend(' LegTekst ')']);

title('Kraften i y retning');
xlabel('tid');
ylabel('Newton');
set(gcf, 'Name', 'Kraften i y retning');
grid

%% Plotter Kraft i z retning i forhold til tiden
RFig = figure;
if(nvar > 1)

```

```

    hold on
end

LegTekst = '';
for k=1:nvar
    % x = varargin{k}.IP(1:varargin{k}.L, 1);
    x = varargin{k}.t(1:varargin{k}.L-1);
    y = varargin{k}.FO(1:varargin{k}.L-1, 3);

    plot(x, y, varargin{k}.Color)
    if(k<nvar)
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' ' ', '];
    else
        LegTekst = [LegTekst ' ' varargin{k}.Navn ' '];
    end
end
eval(['legend(' LegTekst ')']);

title('Kraften i z retning');
xlabel('tid');
ylabel('Newton');
set(gcf, 'Name', 'Kraften i z retning');
grid

end

```

B.7 Film

```

function Film(Navn, Vinkler, Steg)
% Film(Navn, Vinkler, Steg)
% Navn = navnet på filmen
% Vinkler = [Nx3] eulervinkler
% Steg = steglengden

fig = figure('visible','on');
set(fig, 'render', 'OpenGL');

mov=avifile(Navn,'compression','CinePak');

for l=1:Steg:length(Vinkler)

    fi = Vinkler(l,1);
    tetha = Vinkler(l,2);
    psi = Vinkler(l,3);
    Rl = Rot(fi, tetha, psi);

    Prosjektilkropp(Rl);
    hold on
    Prosjektilbak(Rl);
    BranrorUtenFinner(Rl);
    hold off
    grid off

    light('Position',[700 700 700],'Style','local');
    light('Position',[-700 -700 -700],'Style','local');
    xlabel('x')

```

```

ylabel('y')
xlabel('z')
axis([-1520 1520 -1520 1520 -1520 1520])

set(gca, 'CameraPositionMode', 'manual');
set(gca, 'CameraPosition', [-1000*0, 6000, -4000*0]);
set(gca, 'CameraUpVector', [0, 0, -1]);

F = getframe(fig);
mov=addframe(mov,F);

end

close(fig);
mov = close(mov);

```

B.8 HentEXCELData

```

function HentEXCELData
Filnavn = 'aero_OEF3BB_2008.xls';

adat.Mach      = xlsread(Filnavn, 1, 'A2:A22');
adat.CX0       = xlsread(Filnavn, 1, 'B2:B22');
adat.CX2       = xlsread(Filnavn, 1, 'C2:C22');
adat.CNa       = xlsread(Filnavn, 1, 'D2:D22');
adat.CNa3      = xlsread(Filnavn, 1, 'E2:E22');
adat.Cma       = xlsread(Filnavn, 1, 'F2:F22');
adat.Cmq       = xlsread(Filnavn, 1, 'G2:G22');
adat.Cypa      = xlsread(Filnavn, 1, 'H2:H22');
adat.Clp       = xlsread(Filnavn, 1, 'I2:I22');
adat.Cmpa      = xlsread(Filnavn, 1, 'J2:J22');
adat.CNaf1     = xlsread(Filnavn, 1, 'K2:K22');
adat.CLaf1     = xlsread(Filnavn, 1, 'L2:L22');
adat.CX0f1     = xlsread(Filnavn, 1, 'M2:M22');
adat.Clpf1     = xlsread(Filnavn, 1, 'N2:N22');
adat.Cldf1     = xlsread(Filnavn, 1, 'O2:O22');
adat.CNaf2     = xlsread(Filnavn, 1, 'P2:P22');
adat.CLaf2     = xlsread(Filnavn, 1, 'Q2:Q22');
adat.CX0f2     = xlsread(Filnavn, 1, 'R2:R22');
adat.Clpf2     = xlsread(Filnavn, 1, 'S2:S22');
adat.Cldf2     = xlsread(Filnavn, 1, 'T2:T22');

adat.BBMach    = xlsread(Filnavn, 1, 'A25:A37');
adat.BB        = xlsread(Filnavn, 1, 'B25:B37');

% Før baseburn (_fb)
adat.m_fb      = xlsread(Filnavn, 1, 'B41:B41'); % masse
adat.Ixx_fb    = xlsread(Filnavn, 1, 'B42:B42'); % treghet
adat.Iyy_fb    = xlsread(Filnavn, 1, 'B43:B43'); % treghet
adat.Rf_fb     = xlsread(Filnavn, 1, 'B44:B44'); % Avstand CG finner.

% Etter baseburn (_eb)
adat.m_eb      = xlsread(Filnavn, 1, 'C41:C41'); % masse
adat.Ixx_eb    = xlsread(Filnavn, 1, 'C42:C42'); % treghet
adat.Iyy_eb    = xlsread(Filnavn, 1, 'C43:C43'); % treghet
adat.Rf_eb     = xlsread(Filnavn, 1, 'C44:C44'); % Avstand CG finner.

adat.D         = xlsread(Filnavn, 1, 'B40:B40'); % referanse diameter

```

```

adat.rf1      = xlsread(Filnavn, 1, 'B45:B45'); % Avstand x {b} CP store
finner
adat.rf2      = xlsread(Filnavn, 1, 'B46:B46'); % Avstand x {b} CP småe
finner

%% ToDo oppdater kommentarer

adat.CRB      = xlsread(Filnavn, 1, 'B46:B46');
adat.CV       = xlsread(Filnavn, 1, 'B46:B46');
adat.mF       = xlsread(Filnavn, 1, 'B46:B46');
adat.mA       = xlsread(Filnavn, 1, 'B46:B46');

clear Filnavn;

save EXCELData

end

```

B.9 Konfig_OEF3HB_UF

```

function Konfig_OEF3HB_UF

% bestemmes senere
% q_ini = 0; % Initiell pitch-rate [rad/s]. Verdien 2
er hentet fra FFI/RAPPORT-2005/00048 side 31, kap. 8
% r_ini = 0; % Initiell yaw-rate [rad/s]. Verdien 2 er
hentet fra FFI/RAPPORT-2005/00048 side 31, kap. 8
% p_ini = 2*pi*V0/(TW*Dref); % Granatens initielle spinn [rad]

% Konstanter
TW = 20; % Riflestigning; målt i kalibre per omdreining

% OEF3HB-data
Dref = 0.155; % Referanselengde; diameter for
155mm granat
Sref = pi*(Dref/2)^2; % Referanseareal

m = 43.67; % Granatens masse

Ixx = 0.149; % Tregghetsmoment om x-aksen [kgm^2]
Iyy = 1.62; % Tregghetsmoment om y-aksen [kgm^2]
Izz = 1.62; % Tregghetsmoment om z-aksen [kgm^2]
I_matr = [Ixx,0,0;0,Iyy,0;0,0,Izz]; % Tregghetsmatrisa
inv_I_matr = I_matr^-1; % Inverse av tregghetsmatrisa

%Tregghetsmoment for "For-body"
Ixx_F = 8.1*10^(-4); % Tregghetsmoment om x-aksen
[kgm^2]"For-body"

save Konfig_OEF3HB_UF

end

```

B.10 Prosjektilbak

```
function Bak = Prosjektilbak(R)
% Prosjektilbak(R)
% Denne funksjonen plotter en 155mm Prosjektils bakre del i xyz planet
% der x er framover, y er opp og z langs finnen.
% Funksjonen returnerer håndtaket til "overflaten".
% R er rotasjonsmatrisen

Nturn=16;

% definerer grunnvektoren.
GV = [-222.93, 61.25, 0
      -272.43, 76.6, 0
      -353.25, 66.25, 0
      -348.73, 61.25, 0
      -222.93, 61.25, 0];

Nvec=length(GV);

% Lager resten av vektorene
V = zeros([Nturn size(GV)]);

for k=1:Nturn
    Rint = Rot(k/Nturn*2*pi,0,0);
    for l=1:Nvec
        V(k,l,:)=(Rint*GV(l,:)).';
    end
end

% Roterer vektorene i rett stilling
vt = zeros(1,3);
for k=1:Nturn
    for l=1:Nvec
        vt(1,1)=V(k,l,1);
        vt(1,2)=V(k,l,2);
        vt(1,3)=V(k,l,3);
        vt=(R*vt.').';
        V(k,l,:)=vt;
    end
end

% Lager overflaten
x=zeros(Nturn+1, Nvec);
y=zeros(Nturn+1, Nvec);
z=zeros(Nturn+1, Nvec);

for k=1:Nturn
    if k==1
        for l=1:Nvec
            x(k,l) = V(k,l,1);
            x(Nturn+1,l) = V(k,l,1);
            y(k,l) = V(k,l,2);
            y(Nturn+1,l) = V(k,l,2);
            z(k,l) = V(k,l,3);
            z(Nturn+1,l) = V(k,l,3);
        end
    else
        for l=1:Nvec
```



```

        x(k,1) = V(k,1,1);
        y(k,1) = V(k,1,2);
        z(k,1) = V(k,1,3);
    end
end
end

% lager flaten II
Bak = surf(x,y,z);
set(Bak, 'FaceColor', [0,0,0]);
set(Bak, 'LineStyle', 'none');
set(Bak, 'FaceLighting', 'phong');

end

```

B.11 Prosjektilkropp

```

function Kropp = Prosjektilkropp(R)
% Prosjektilkropp(R)
% Denne funksjonen plotter en 155mm Prosjektilkropp i xyz planet
% der x er framover, y er opp og z langs finnen.
% Funksjonen returnerer håndtaket til "overflaten".
% R er rotasjonsmatrisen

Nturn=16;

% definerer grunnvektoren.
GV = [387.57, 0, 0
      387.57, 35.35, 0
      225.04, 60.77, 0
      61.34, 77, 0
      -194.33, 76.93, 0
      -193.87, 74.33, 0
      -231.79, 74.33, 0
      -231.33, 76.93, 0
      -269.29, 76.76, 0
      -272.43, 76.6, 0
      -222.93, 61.25, 0
      -222.93, 0, 0];

Nvec=length(GV);

% Lager resten av vektorene
V = zeros([Nturn size(GV)]);

for k=1:Nturn
    Rint = Rot(k/Nturn*2*pi,0,0);
    for l=1:Nvec
        V(k,l,:)=(Rint*GV(l,:)).';
    end
end

% Roterer vektorene i rett stilling
vt = zeros(1,3);
for k=1:Nturn
    for l=1:Nvec
        vt(1,1)=V(k,l,1);
        vt(1,2)=V(k,l,2);
        vt(1,3)=V(k,l,3);
    end
end

```

```

        vt=(R*vt.').';
        V(k,l,:)=vt;
    end
end

% Lager overflaten
x=zeros(Nturn+1, Nvec);
y=zeros(Nturn+1, Nvec);
z=zeros(Nturn+1, Nvec);

for k=1:Nturn
    if k==1
        for l=1:Nvec
            x(k,l) = V(k,l,1);
            x(Nturn+1,l) = V(k,l,1);
            y(k,l) = V(k,l,2);
            y(Nturn+1,l) = V(k,l,2);
            z(k,l) = V(k,l,3);
            z(Nturn+1,l) = V(k,l,3);
        end
    else
        for l=1:Nvec
            x(k,l) = V(k,l,1);
            y(k,l) = V(k,l,2);
            z(k,l) = V(k,l,3);
        end
    end
end

% lager flaten II
Kropp = surf(x,y,z);
set(Kropp, 'FaceColor', [0.6,0.6,0.6]);
set(Kropp, 'LineStyle', 'none');
set(Kropp, 'FaceLighting', 'phong');

end

```

B.12 BrannrorUtenFinner

```

function Ror = BranrorUtenFinner(R)
% BranrorUtenFinner(R)
% Denne funksjonen plotter et brannrør til en 155mm Prosjektilkropp
% i xyz planet.
% der x er framover, y er opp og z langs finnen.
% Funksjonen returnerer håndtaket til "overflaten".
% R er rotasjonsmatrisen

Nturn=16;

% definerer grunnvektoren.
GV = [516.9 0 0
      516.05 4.98 0
      513.37 9.67 0
      509.82 12.74 0
      504.44 14.78 0
      387.57 35.35 0
      387.57 0 0];
Nvec=length(GV);

```

```

% Lager resten av vektorene
V = zeros([Nturn size(GV)]);

for k=1:Nturn
    Rint = Rot(k/Nturn*2*pi,0,0);
    for l=1:Nvec
        V(k,l,:)=(Rint*GV(l,:)).'.';
    end
end

% Roterer vektorene i rett stilling
vt = zeros(1,3);
for k=1:Nturn
    for l=1:Nvec
        vt(1,1)=V(k,l,1);
        vt(1,2)=V(k,l,2);
        vt(1,3)=V(k,l,3);
        vt=(R*vt.').'.';
        V(k,l,:)=vt;
    end
end

% Lager overflaten
x=zeros(Nturn+1, Nvec);
y=zeros(Nturn+1, Nvec);
z=zeros(Nturn+1, Nvec);

for k=1:Nturn
    if k==1
        for l=1:Nvec
            x(k,l) = V(k,l,1);
            x(Nturn+1,l) = V(k,l,1);
            y(k,l) = V(k,l,2);
            y(Nturn+1,l) = V(k,l,2);
            z(k,l) = V(k,l,3);
            z(Nturn+1,l) = V(k,l,3);
        end
    else
        for l=1:Nvec
            x(k,l) = V(k,l,1);
            y(k,l) = V(k,l,2);
            z(k,l) = V(k,l,3);
        end
    end
end

% lager flaten II
Ror = surf(x,y,z);
set(Ror, 'FaceColor', [0.4,0.6,0.6]);
set(Ror, 'LineStyle', 'none');
set(Ror, 'FaceLighting', 'phong');

end

```

B.13 BrannrorMedFinner

```
function [Ror Sma1 Sma2 Stor] = BranrorMedFinner(R1, R2)
% [Ror Sma1 Sma2 Stor] = BranrorMedFinner(R1, R2)
% Denne funksjonen plotter et brannrør med finner til en 155mm
% prosjektilkropp i xyz planet.
% der x er framover, y er opp og z langs finnen.
% Funksjonen returnerer håndtaket til "overflaten".
% Ror = brannrør
% Sma = Småe finner
% Stor = Store finner
% R1 er rotasjonsmatrisen til brannrøret og
% R2 er rotasjonsmatrisen til styrefinnene

Ror = BranrorUtenFinner(R1);

%% Småe finner
% definerer grunnvektorene.
GV1 = [-0.76,1.31,0
       7.85,5.72,0
       11.82,7.4,0
       12.33,7.31,0
       12.44,7.18,0
       12.5,7.02,0
       12.32,6.54,0
       8.88,3.94,0
       0.76,-1.31,0
       -7.85,-5.72,0
       -11.82,-7.4,0
       -12.33,-7.31,0
       -12.44,-7.18,0
       -12.5,-7.02,0
       -12.32,-6.54,0
       -8.88,-3.94,0];
bg1 = [0 0 0];
ed1 = [0 0 40];
trans = [419.9 0 0];

GV2 = GV1;
GV2(:,3) = GV2(:,3)+40;

Rx = Rot(pi,0,0);
GV3 = GV1;
GV4 = GV2;
bg2 = (Rx*bg1.').';
ed2 = (Rx*ed1.').';

L = length(GV1);

for k = 1:L
    GV3(k,:) = (Rx*GV3(k, :).').';
    GV4(k,:) = (Rx*GV4(k, :).').';
end

Rx = Rot(pi/2,0,0);
bg1 = (Rx*bg1.').';
bg2 = (Rx*bg2.').';
ed1 = (Rx*ed1.').';
ed2 = (Rx*ed2.').';
for k = 1:L
```

```

    GV1(k,:) = (Rx*GV1(k,:)).';
    GV2(k,:) = (Rx*GV2(k,:)).';
    GV3(k,:) = (Rx*GV3(k,:)).';
    GV4(k,:) = (Rx*GV4(k,:)).';
end

bg1 = bg1+trans;
bg2 = bg2+trans;
ed1 = ed1+trans;
ed2 = ed2+trans;
for k = 1:L
    GV1(k,:) = GV1(k, :)+trans;
    GV2(k,:) = GV2(k, :)+trans;
    GV3(k,:) = GV3(k, :)+trans;
    GV4(k,:) = GV4(k, :)+trans;
end

bg1 = (R1*bg1.').';
bg2 = (R1*bg2.').';
ed1 = (R1*ed1.').';
ed2 = (R1*ed2.').';
for k = 1:L
    GV1(k,:) = (R1*GV1(k, :)).';
    GV2(k,:) = (R1*GV2(k, :)).';
    GV3(k,:) = (R1*GV3(k, :)).';
    GV4(k,:) = (R1*GV4(k, :)).';
end

x1 = zeros(L+1, 4);
x2 = x1;
y1 = x1;
y2 = x1;
z1 = x1;
z2 = x1;

for k = 1:L
    x1(k,1) = bg1(1); x2(k,1) = bg2(1);
    x1(k,2) = GV1(k,1); x2(k,2) = GV3(k,1);
    x1(k,3) = GV2(k,1); x2(k,3) = GV4(k,1);
    x1(k,4) = ed1(1); x2(k,4) = ed2(1);

    y1(k,1) = bg1(2); y2(k,1) = bg2(2);
    y1(k,2) = GV1(k,2); y2(k,2) = GV3(k,2);
    y1(k,3) = GV2(k,2); y2(k,3) = GV4(k,2);
    y1(k,4) = ed1(2); y2(k,4) = ed2(2);

    z1(k,1) = bg1(3); z2(k,1) = bg2(3);
    z1(k,2) = GV1(k,3); z2(k,2) = GV3(k,3);
    z1(k,3) = GV2(k,3); z2(k,3) = GV4(k,3);
    z1(k,4) = ed1(3); z2(k,4) = ed2(3);
end
x1(L+1,:) = x1(1,:); x2(L+1,:) = x2(1,:);
y1(L+1,:) = y1(1,:); y2(L+1,:) = y2(1,:);
z1(L+1,:) = z1(1,:); z2(L+1,:) = z2(1,:);

```

```

Sma1 = surf(x1,y1,z1);
set(Sma1, 'FaceColor', [1,0,0]);
set(Sma1, 'LineStyle', 'none');
set(Sma1, 'FaceLighting', 'phong');

Sma2 = surf(x2,y2,z2);
set(Sma2, 'FaceColor', [1,0,0]);
set(Sma2, 'LineStyle', 'none');
set(Sma2, 'FaceLighting', 'phong');

%% Store finner
w = 15;
SV1 = [0 3.04 -29.66-w
       27.88 0.99 -24.75-w
       28.73 0 -24.6-w
       27.88 -0.99 -24.75-w
       0 -3.04 -29.66-w
       -27.88 -0.99 -34.57-w
       -28.73 0 -34.72-w
       -27.88 0.99 -34.57-w];

SV2 = [0 3.04 29.66+w
       27.88 0.99 24.75+w
       28.73 0 24.6+w
       27.88 -0.99 24.75+w
       0 -3.04 29.66+w
       -27.88 -0.99 34.57+w
       -28.73 0 34.72+w
       -27.88 0.99 34.57+w];
bg = [0 0 -29.66-w];
ed = [0 0 29.66+w];

L = length(SV1);

bg = (R2*bg.').';
ed = (R2*ed.').';
for k = 1:L
    SV1(k,:) = (R2*SV1(k,)).';
    SV2(k,:) = (R2*SV2(k,)).';
end

bg = bg+trans;
ed = ed+trans;
for k = 1:L
    SV1(k,:) = SV1(k,)+trans;
    SV2(k,:) = SV2(k,)+trans;
end

bg = (R1*bg.').';
ed = (R1*ed.').';

for k = 1:L
    SV1(k,:) = (R1*SV1(k,)).';
    SV2(k,:) = (R1*SV2(k,)).';
end

x = zeros(L+1, 4);
y = x;
z = x;

```

```

for k = 1:L
    x(k,1) = bg(1);
    x(k,2) = SV1(k,1);
    x(k,3) = SV2(k,1);
    x(k,4) = ed(1);

    y(k,1) = bg(2);
    y(k,2) = SV1(k,2);
    y(k,3) = SV2(k,2);
    y(k,4) = ed(2);

    z(k,1) = bg(3);
    z(k,2) = SV1(k,3);
    z(k,3) = SV2(k,3);
    z(k,4) = ed(3);
end

x(L+1,:) = x(1,:);
y(L+1,:) = y(1,:);
z(L+1,:) = z(1,:);

Stor = surf(x,y,z);
set(Stor, 'FaceColor', [0,0,0]);
set(Stor, 'LineStyle', 'none');
set(Stor, 'FaceLighting', 'phong');

end

```

B.14 Rot

```

function R = Rot(fi, theta, psi)
% R = Rot(fi, theta, psi)
% Denne funksjonen gir deg rotasjonsmatrisen hvis du
% har de tre vinklene som fi, theta, psi.

RX = [1, 0, 0
      0, cos(fi), -sin(fi)
      0, sin(fi), cos(fi)];

RY = [cos(theta), 0, sin(theta)
      0, 1, 0
      -sin(theta), 0, cos(theta)];

RZ = [cos(psi), -sin(psi), 0
      sin(psi), cos(psi), 0
      0, 0, 1];

R = RZ*RY*RX;

end

```