

# Mikrofonmatrise integrert med treghetsnavigasjonssystem

Zeeshan Aslam  
ELD5930 Masteroppgave



Oppgave for graden  
Master i Kybernetikk  
30 studiepoeng

Institutt for teknologisystemer  
Det matematisk-naturvitenskapelige fakultet

Vår 2019

# Mikrofonmatrise integrert med treghetsnavigasjonssystem

Zeeshan Aslam  
ELD5930 Masteroppgave



© 2019 Zeeshan Aslam

Mikrofonmatrise integrert med treghetsnavigasjonssystem

<http://www.duo.uio.no/>

Trykk: Representralen, Universitetet i Oslo

# Sammendrag

Denne masteroppgaven er basert på en problemstilling som er gitt av selskapet Mantena AS. Det er ønskelig å bruke en mikrofonmatrise for å utføre feilsøking av understellet til et tog. Dagens løsning bruker akselerometere og sensorer som kobles på understellet i en Dynamisk Testbenk. Hele denne prosessen tar lang tid å gjennomføre. Dette blir belyst nærmere i introduksjonskapittelet. Deretter vil det bli redegjort for eksisterende løsninger som blir brukt for feilsøking av tog.

En del av rapporten vil basere seg på alle aspekter rundt en mikrofonmatrise. Vi vil starte med å forklare hva en mikrofonmatrise er også vil det følges opp med å kartlegge hvor mikrofonmatriser anvendes i dagens sammenheng. Til slutt vil temaet for mikrofonmatriser rundes av med en generell fremstilling av hvordan en mikrofonmatrise kan settes opp både matematisk og fysisk. Det vil også bli gitt en forklaring på hvilken type mikrofonmatrise som vil egne seg best for formålet til Mantena AS.

Etter å ha dokumentert en del rundt temaet for mikrofonmatrise fortsetter rapporten videre med å gi en beskrivelse på hvordan det er mulig å bruke en treghetsnavigasjonssystem integrert med en mikrofonmatrise. Det er ønskelig å ha en mikrofonmatrise som kan bevege seg rundt i den Dynamiske Testbenken i en gitt bane. Hensikten med det er å orientere seg og posisjonere seg etter støykilder som blir detektert av mikrofonmatrisen. Orienteringen og posisjoneringen for mikrofonmatrisen med et TNS vil bli gjort med eksterne målinger ved hjelp av et kamera og ikke en GPS. Modellen for kameraet som blir brukt er den perspektive kameramodellen. Kameraet skal bruke den relative bevegelsen mellom plattformen og stasjonære punkter i treghetsrammen for å definere TNS. For å realisere denne delen av oppgaven blir det laget en banegenerator. Banegeneratoren er en analytiske løsning for en filtermodell. Denne filtermodellen vil bestå av en målt spesifikk kraft og en målt vinkelhastighet. Siden filtermodellen vil være en ulineært stokastisk prosess er det hensiktsmessig å bruke en estimator på filtermodellen. For estimatoren vil det bli brukt en tilstandsmatrise med alle tilstander, den målte spesifikke kraften og vinkelhastigheten som pådrag og målinger fra kameraet som sensorverdier. Estimatorens jobb vil være å gi oss verdier som er tilnærmet lik den analytiske løsningen. Estimatoren som skal brukes er en UKF og dette teamet vil bli belyst grundig i kapittelet for teori.

Som avslutningsvis for oppgaven, vil det bli redegjort for den målte filtermodellen og hvordan den er i forhold til den analytiske løsningen. Dette vil gi grunnlag for å se om data fra filtermodellen kan brukes i en estimator som UKF.

*Alle figurer, bilder, lenker, data og tilleggsinformasjon som er hentet fra de forskjellige nettsidene eller andre type kilder har det blitt hentet samtykke for fra rettighetshavere. Dette er blitt gjort enten i muntlig eller skriftlig form.*

# Forord

Oppgaven er skrevet av meg Zeeshan Aslam, som en del av masteroppgave utført ved ITS. Oppgaven var i utgangspunktet basert på en samarbeidsprosjekt mellom ITS ved UiO, Mantena AS og undertegnede. På grunn av uforutsette utfordringer ble oppgaven forandret underveis. Til tross for utfordringen knyttet til endret problemstilling underveis, har oppgaven vært spennende, utfordrende og veldig lærerikt.

Jeg vil starte med å takke min professor og veileder Oddvar Hallingstad ved ITS. Hans råd, støtte og veiledning under hele masteren har vært til stor hjelp. Deretter vil jeg takke min lærer Thomas Olsvik Opsahl ved FFI for all den tiden han har gitt meg med ekstra forelesninger for å forsterke mitt fagkunnskap relatert til masteroppgaven. Jeg vil også takke Peter Lefoka ved Mantena AS for å ha gitt meg muligheten til å skrive oppgaven for dem. Til slutt ønsker jeg å takke min familie som har støttet meg gjennom alle disse årene, slik at jeg fikk fullført min utdanning.

*“If something is important enough, even if the odds are against you, you should still do it.”*

- Elon Musk

# Innhold

	Side
<b>Sammendrag</b>	<b>i</b>
<b>Forord</b>	<b>ii</b>
<b>Figurer</b>	<b>vii</b>
<b>Symboler</b>	<b>viii</b>
<b>Ordlister</b>	<b>x</b>
<b>Akronymer og forkortelser</b>	<b>xi</b>
<b>Tabeller</b>	<b>xii</b>
<b>Liste av algoritmer</b>	<b>xiii</b>
<b>1 Introduksjon</b>	<b>1</b>
<b>2 Bakgrunn og hensikt</b>	<b>4</b>
2.1 Dagens løsning brukt av Mantena AS . . . . .	4
2.1.1 Feilsøking med sensorer og akselerometer . . . . .	5
2.2 Eksisterende løsninger hos andre aktører . . . . .	5
2.2.1 Gatekeeper Intelligent Security [7] . . . . .	5
2.2.1.1 Intelligent Train Understell Scanner (ITUS) . . . . .	6
2.2.2 Lakesight Technologies [20] . . . . .	6
2.2.2.1 Inspeksjon med bildeanalyse i 3D av rullerende tog [19] . . . . .	6
2.2.3 Baker Hughes a GE company [2] . . . . .	8
2.2.3.1 GE WheelStar [2] . . . . .	8
2.2.3.1.1 WheelStar - Gantry: for heavy maintenance: . . . . .	8
2.2.3.1.2 WheelStar - Underfloor (UFPE): . . . . .	9
2.2.3.1.3 WheelStar - Portable Phased Array: . . . . .	10
2.3 Forskjellige anvendelser av mikrofonmatriser . . . . .	10
2.3.1 Hva er en mikrofonmatrise? [21] . . . . .	11
2.3.2 Mikrofonmatrise brukt i lyd studio . . . . .	14
2.3.3 Mikrofonmatrise brukt i forelesningssal eller konferanserom . . . . .	16
2.3.4 Sorama [29] . . . . .	17
2.3.4.1 Sorama Cam64 [31] . . . . .	18
2.3.4.2 Sorama Cam1K [30] . . . . .	18
2.3.5 Brüel & Kjær [3] . . . . .	19
2.3.5.1 Sfærisk mikrofonmatrise: . . . . .	20
2.3.5.2 Håndholdt mikrofonmatrise . . . . .	21
2.3.5.3 Mikrofonmatrise i form av et gitter . . . . .	22
2.3.5.4 Sammenleggbare mikrofonmatrise . . . . .	23

2.3.5.5	Hjulformet mikrofonmatrise . . . . .	24
<b>3</b>	<b>Teori</b>	<b>25</b>
3.1	Teori - del 1 . . . . .	25
3.1.1	Mikrofonmatriser og beamforming [33] . . . . .	26
3.1.2	Planar mikrofonmatrise [32] [33] . . . . .	30
3.2	Teori - del 2 . . . . .	33
3.2.1	Banegenerator [17] [23] . . . . .	34
3.2.2	Perspektiv kameramodell [26] . . . . .	36
3.2.2.1	Pose i 2D og 3D . . . . .	37
3.2.2.1.1	Pose i 2D . . . . .	37
3.2.2.1.2	Pose i 3D . . . . .	41
3.2.2.2	Perspektiv kameramodell - extrinsic . . . . .	45
3.2.2.3	Perspektiv kameramodell - intrinsic . . . . .	48
3.2.3	TNS - Treghetsnavigasjonssystem [16] [22] . . . . .	50
3.2.3.1	Gyroskop . . . . .	51
3.2.3.1.1	MEMS-Gyroskop [35] . . . . .	51
3.2.3.2	Akselerometer . . . . .	52
3.2.3.2.1	MEMS-Akselerometer [35] . . . . .	52
3.2.3.3	Markov-prosess . . . . .	52
3.2.3.4	Støy . . . . .	53
3.2.3.4.1	Hvit støy [6] . . . . .	53
3.2.3.4.2	Farget støy . . . . .	54
3.2.3.5	Eulers integrasjonsmetode [14] . . . . .	54
3.2.3.6	Spesifikk kraft . . . . .	55
3.2.3.7	Vinkelhastighet . . . . .	55
3.2.3.8	Tre-akset plattform, flat ikke-roterende jord [9] . . . . .	56
3.2.3.8.1	Transformasjonsfeilen referert Eulervinklene . . . . .	56
3.2.3.8.2	Transformasjonsfeilen referert n-systemet . . . . .	56
3.2.4	Sammendrag av likninger for vårt system [25] . . . . .	58
3.2.5	Stokastiske prosesser [16] . . . . .	62
3.2.6	Diskretisering for en generelt ikke-lineær stokastiske prosess - del 1 [5][13] . . . . .	63
3.2.7	UKF - Unscented Kalmanfilter [11] . . . . .	67
3.2.7.1	Unscented transformasjon . . . . .	67
3.2.7.2	Utvidet symmetrisk sett av sigmapunkter - Sigmasett 2 . . . . .	69
3.2.7.3	UKF med ikke-additativ støy . . . . .	69
<b>4</b>	<b>Metode for simulering - presentert med matematiske likninger</b>	<b>73</b>
4.1	Oppsettet til banegeneratoren og kameraet, og deres initialverdier . . . . .	73
4.2	Kontinuerlig banegenerator . . . . .	75
4.2.1	Banegenerator uten støy på akselerometer og gyroer . . . . .	75
4.2.1.1	Derivasjon av kjent posisjon og stilling . . . . .	75
4.2.1.2	Formulert som differensiallikning . . . . .	76
4.2.1.3	Formulert som integrallikning . . . . .	76
4.3	Gyro- og akselerometermålinger med støy . . . . .	77
4.4	Kontinuerlig prosessmodell . . . . .	77

4.5	Banegenerator på diskret form . . . . .	79
4.5.1	Banegenerator uten støy på akselerometre og gyroer . . . . .	79
4.5.2	Målelikninga for Kalmanfilteret med målestøy . . . . .	79
4.5.3	Akselerometer- og gyromålinger med støy . . . . .	79
4.6	Pseudo-kode for banegenerator . . . . .	81
4.7	Initialverdier for støy til filtermodellen . . . . .	82
4.8	Filtermodell på diskretform . . . . .	83
4.8.1	Utleddning av filtermodellen fra integrallikningene . . . . .	83
4.9	Diskretisering av ikke-lineær stokastiske prosess - del 2 . . . . .	85
<b>5</b>	<b>Resultater og diskusjon</b>	<b>86</b>
5.1	De deterministiske testene . . . . .	86
5.1.1	Banegenerator: . . . . .	86
5.2	Sann filtermodell: . . . . .	88
5.2.1	Sammenheng mellom banegenerator og sanne filtermodell: . . . . .	89
5.3	De stokastiske testene . . . . .	91
5.3.1	Målte filtermodellen . . . . .	91
5.3.2	Sammenheng mellom banegenerator og den målte filtermodellen: . . . . .	91
5.4	Drøfting og diskusjon av resultatene . . . . .	98
<b>6</b>	<b>Konklusjon</b>	<b>101</b>
<b>7</b>	<b>Veien videre</b>	<b>102</b>
	<b>Referanser</b>	<b>103</b>
<b>A</b>	<b>Matlab koden</b>	<b>106</b>

# Figurer

1	Mantena sitt Dynamiske Testbenk innvendig og utvendig . . . . .	2
2	Dobbel-visnings skanner . . . . .	6
3	Chromasens skanner boks . . . . .	7
4	Chromasens skanner plassert på togskiner . . . . .	7
5	WheelStar - gantry: for heavy maintenance . . . . .	9
6	GE Krautkramer WheelStar™ In-Service Train Wheel Inspection System . . . . .	9
7	WheelStar - Portable Phased Array . . . . .	10
8	Simpleste formen av en mikrofonmatrise med en mikrofon plassert på venstre siden og en på høyre siden . . . . .	11
9	De tre forskjellige retnings egenskapene av en mikrofon . . . . .	12
10	ZYLIA ZM-1 mikrofonmatrise . . . . .	14
11	Stereoopptak med mobil telefon vs ZYLIA ZM-1 mikrofonmatrise . . . . .	15
12	ZYLIA ZM-1 mikrofonmatrise med opptak av forskjellige lydkilder fra forskjellige retninger . . . . .	15
13	ClearOne beamforming mikrofonmatrise . . . . .	16
14	Virkemåten til beamforming metoden i en mikrofonmatrise . . . . .	17
15	Sorama Cam64 . . . . .	18
16	Sorama Cam1K sett fra baksiden . . . . .	18
17	Sorama Cam1K sett forfra . . . . .	19
18	Sfærisk mikrofonmatris . . . . .	20
19	Forskjellige fremstillinger av håndholdt mikrofonmatriser . . . . .	21
20	Grid basert mikrofonmatriser . . . . .	22
21	Oppsett for sammenleggbare mikrofonmatrise . . . . .	23
22	Sammenleggbare mikrofonmatrise . . . . .	23
23	Hjulformet mikrofonmatrise . . . . .	24
24	Hjulformet mikrofonmatriser . . . . .	24
25	Beamforming . . . . .	26
26	Sfærisk koordinatsystem . . . . .	27
27	Uniform vektning vist for lineær mikrofonmatrise . . . . .	29
28	Mikrofonmatrise formet som et rektangulært plan . . . . .	30
29	Geometrisk fremstilling med alle rammene i Mantena AS sitt dynamisk testbenk . . . . .	33
30	Trigonometrisk fremstilling av enhets sirkel for å vise bevegelse langs sirkelen [18] . . . . .	34
31	Perspektiv eller pin-hole kameramodell . . . . .	36
32	Perspektiv/pin-hole kameramodell fremstilt på forskjellig måter . . . . .	36
33	To forskjellige representasjoner av 2D pose med både rammer og punkt i planet . . . . .	37
34	Trigonometrisk beskrivelse av 2D-pose med to akser . . . . .	38
35	Eulervinkel dreiesekvens $zxy$ også kalt 3-2-1 . . . . .	41
36	Vinkel-akse par $(\theta, \mathbf{v})$ . . . . .	43
37	Ortonormal rotasjon for ramme A og ramme B . . . . .	44
38	Extrinsic delen til en perspektiv kameramodell . . . . .	45
39	Perspektiv projeksjon fra 3D til 2D for et perspektiv kameramodell . . . . .	47
40	Intrinsic delen til et perspektiv kameramodell . . . . .	48
41	Intrinsic delen til et perspektiv kameramodell med fokal lengde $f$ . . . . .	49

42	Diverse bevegelsesretninger til et 3-akset akselerometer og et 3-akset gyro i et treghetsnavigasjonssystem med 6-DOF . . . . .	50
43	Gyroskop . . . . .	51
44	MEMS-Gyroskop fra ST . . . . .	51
45	En vibrerende masse gyroskop . . . . .	52
46	Blockdiagram av TNS . . . . .	62
47	Unscented transformasjon . . . . .	68
48	Posisjon i 3D for banegenerator . . . . .	86
49	Posisjon i 2D for banegenerator . . . . .	87
50	Fart for mikrofonmatrise presentert for banegenerator . . . . .	87
51	Posisjon i 3D for sann filtermodell . . . . .	88
52	Posisjon i 2D for sann filtermodell . . . . .	88
53	Fart for mikrofonmatrise presentert for sann filtermodell . . . . .	89
54	Begge banene overlappet for 30 sampler . . . . .	89
55	Begge banene overlappet for 300 sampler . . . . .	90
56	Begge banene overlappet for 3000 sampler . . . . .	90
57	Posisjon for den målte filtermodellen for omløpstid med 30 sek . . . . .	91
58	Banegenerator og målt filtermodell overlappet med omløpstid på 3 sek . . . . .	91
59	Banegenerator og målt filtermodell overlappet for 50 iterasjoner med omløpstid på 3 sek . . . . .	92
60	Banegenerator og målt filtermodell overlappet for 500 iterasjoner med omløpstid på 3 sek . . . . .	92
61	Banegenerator og målt filtermodell overlappet for 1500 iterasjoner med omløpstid på 3 sek . . . . .	93
62	Banegenerator og målt filtermodell overlappet med omløpstid på 30 sek . . . . .	93
63	Banegenerator og målt filtermodell overlappet for 50 iterasjoner med omløpstid på 30 sek . . . . .	94
64	Banegenerator og målt filtermodell overlappet for 500 iterasjoner med omløpstid på 30 sek . . . . .	94
65	Banegenerator og målt filtermodell overlappet for 1500 iterasjoner med omløpstid på 30 sek . . . . .	95
66	Banegenerator og målt filtermodell overlappet med omløpstid på 60 sek . . . . .	95
67	Banegenerator og målt filtermodell overlappet for 50 iterasjoner med omløpstid på 60 sek . . . . .	96
68	Banegenerator og målt filtermodell overlappet for 500 iterasjoner med omløpstid på 60 sek . . . . .	96
69	Banegenerator og målt filtermodell overlappet for 1500 iterasjoner med omløpstid på 60 sek . . . . .	97
70	Matlab kode for sensormålinger . . . . .	97



# Symboler

$\mathcal{F}^b$	Ramme-b er bodyramme
$\mathcal{F}^c$	Ramme-c er kameraramme
$\mathcal{F}^n$	Ramme-n er treghetsnavigasjonsramme
$\underline{p}^n$	Posisjonsvektor i n-ramme
$\underline{v}^n$	Fartsvektor i n-ramme
$\underline{a}^n$	Akselerasjonsvektor i n-ramme
$\underline{f}^b$	Vektor for spesifikk kraft i b-ramme
$\underline{\hat{f}}^b$	Vektor for målt spesifikk kraft i b-ramme
$\underline{\omega}_b^{nb}$	Vinkelhastighet i b-ramme, sett fra n-ramme representert i b-ramme
$\underline{\tilde{\omega}}_b^{nb}$	Målt vinkelhastighet i b-ramme, sett fra n-ramme representert i b-ramme
$\underline{b}^a$	Bias til akselerometer
$\underline{b}^g$	Bias til gyro
$\underline{v}^a$	Hvit støy til akselerometer
$\underline{v}^g$	Hvit støy til gyro
$\Delta_t$	Tidsinkrement for sampling
$\mathcal{N}(\cdot)$	Normalfordelt
$\delta_{kl}$	Kronecker $\delta : \begin{cases} \delta_{kl} = 1 & \text{for } k = l \\ \delta_{kl} = 0 & \text{for } k \neq l \end{cases}$
$\delta(t - \tau)$	Dirac-delta funksjon: $\int_{-\infty}^{\infty} f(\tau)\delta(t - \tau)d\tau = f(t)$

$R_a^b$	Retningskosinmatrise
$c_\theta$	$\cos(\theta)$
$s_\theta$	$\sin(\theta)$
$\xi_B^A$	Posen mellom $\mathcal{F}^B(\text{ramme}\{B\})$ i forhold til $\mathcal{F}^A(\text{ramme}\{A\})$ (en kombinasjon av rotasjonsmatrise og translasjonsvektor)
$t_B^A$	Translasjon mellom B-ramme og A-ramme
$\oplus$	En vilkårlig variabel brukt for multiplikasjon mellom 2 poser
$\ominus$	Brukt for den inverse av forskjellige variabler i perspektiv kameramodell
$\chi$	Stokastisk variabel
$\vec{x}$	Geometrisk vektor
$\underline{x}$	Algebraisk vektor (kolonnematrise)
$\underline{x}_0$	Initialverdi for tilstanden
$\hat{x}$	Estimert tilstand $x$
$\bar{x}$	Prediktert tilstand $x$
$\underline{P}_0$	Initialverdi for kovariansen
$\hat{P}$	Estimert kovarians
$\bar{P}$	Prediktert kovarians
$K$	Kalmanfilterforsterkning
$S(\underline{\omega}^q) \equiv [\underline{\omega} \times]^q$	Matrise representasjon av " $\vec{\omega} \times$ " i q-systemet (skjev-symmetrisk matrise)

# Ordliste

## Engelsk

Beamforming

Boogie

Camera calibration matrix  $K$

Extrinsic

Gain

Interference

Intrinsic

Microphone array

Plane wave

Post-processing solution

Power spectral density

Reverberation

Side lobe

Spatial

Spatio-Temporal delay

State of the art

Surround

## Norsk

stråleforming

togunderstell

kamera-kalibreringsmatrise  $K$

som kommer utenfra, som virker utenfra

forsterkning (i lyd sammenheng)

interferens

noe som er innvendig

mikrofonmatrise

plan bølge

etterbehandlings løsning

effektspektraltetthet for hvit støy

ekko/gjenklang i lyd sammenheng

sidelober, signaler som strekker seg ut i forskjellige

retninger fra kjernen med en lokal maksima

romlig

i tid- og rom- forsinkelse

dagens løsninger, det nyeste, siste nytt

flerkanals lydsystem

## Akronymer og forkortelser

ARW	Angular Random Walk
EKF	Extended Kalmanfilter også kalt Utvidet Kalmanfilter
FFI	Forsvarets forsknings institutt
IMU	Inertial Measurement Unit
INS	Inertial Navigation system
ITS	Institutt for teknologiske systemer
KF	Kalmanfilter
MEMS	Micro electro mechanical systems
MM	Mikrofonmatrise
NAV	Navigasjon
PMM	Planar mikrofonmatrise
PSD	Power Spectral Density
RKM	Retningskosinmatrise
RMS	Root Mean Square
RRW	Rate Random Walk
SPM	Støtpulsmåling
TNS	Treghetsnavigasjonssystem
UiO	Universitetet i Oslo
UKF	Unscented Kalmanfilter
VRW	Velocity Random Walk

# Tabeller

1	Transformasjonsfeilen referert Eulervinklene . . . . .	56
2	Transformasjonsfeilen referert n-systemet, del 1 . . . . .	56
3	Transformasjonsfeilen referert n-systemet, del 2 . . . . .	57

# List of Algorithms

1	Pseudo-kode for banegenerator . . . . .	81
---	---	----

# 1 Introduksjon

Denne masteroppgaven er skrevet av undertegnede ved Universitetet i Oslo. Det startet i utgangspunktet som et samarbeid med Mantena AS, men underveis ble situasjonen rundt samarbeidet endret. Jeg endte til slutt med en oppgave som var delvis separert fra Mantena AS og deres problemstilling. Grunnen til det var at det oppstod en del utfordringer midtveis i oppgaven som førte til at jeg og min veileder Oddvar Hallingstad måtte revurdere hele oppgaven til Mantena AS for mitt vedkommende. Siden det ikke var mulig å fullføre oppgaven som ble gitt i utgangspunktet, definerte vi den nye problemstillingen til å være relatert til det som var ønskelig i starten, men med en liten vri. Dette ble gjort for å spare tid og arbeid som allerede hadde blitt brukt frem til nå og for å komme i mål med min masteroppgave. I dette kapitlet skal vi introdusere begge problemstillingene. Videre i oppgaven vil det bli drøftet ulike løsningsmetoder for problemstillingene.

Mantena AS er et aksjeselskap hvor Samferdselsdepartementet er hovedeieren av selskapet. Mantena har virksomhet både i Norge og Sverige. Denne virksomheten er fordelt på forskjellige avdelinger og lokalisasjoner. Selskapet har en lang tradisjon innenfor jernbanedrift i Norden og er spesialist på vedlikehold av skinnegående kjøretøy.

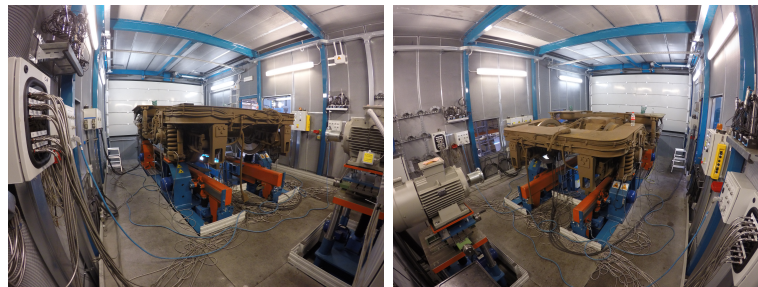
Mantena Grorud er en av underavdelingene til Mantena AS som holder til i Oslo. Arbeidsbeskrivelsen til ansatte på Mantena Grorud inkluderer alt fra vedlikehold av persontog, godstog, komponentvedlikehold, engineering, skader og ombygging av tog. Denne masteroppgaven har som formål å utforske på engineering og komponentvedlikehold.

En av fasilitetene som finnes på Mantena Grorud heter Dynamisk Testbenk som vist i figur 1. Denne testbenken brukes til å se tilstanden til forskjellige typer "boogier"(togunderstell).

Formålet med oppgaven er å få brukt kunnskapen som er tilegnet fra utdanningen i praksis. Mantena har en utfordring når det gjelder feilsøking og skader på boogier. De ønsker å optimalisere løsningen for dette problemet. Deres oppfatning er at dagens løsning på problemet krever mye ressurser og tid. Frem til nå er det blitt brukt et akselerometer på hvert hjullager. Den festes fast til hjullageret og henter data for hver rotasjon som blir gjort. I tillegg til dette festes det en viss antall sensorer på boogiene som vil måle vibrasjoner under testing. Det er svært ønskelig å halvere denne tiden noe som vil frigjøre en del ressurser. Mantena ønsker å bruke en ny tilnærming for å løse problemet. Med dagens teknologi finnes det en del muligheter for å effektivisere denne type utfordringer. En av ideene som er vurdert som den alternative løsningen for feilsøking i boogier er å løse problemet med en *mikrofonmatrise*. Bruk av mikrofonmatrise er en veldig kjent metode hvor fokuset primært er på lyd. Det finnes også mikrofonmatriser som har integrerte kameraer. Denne type mikrofonmatriser brukes for at støykilden fra objektet kan vises med både lyd og bilde. Mikrofonmatrisens formål er å rette seg etter kilden og filtrere bort alt annet form for forstyrrelser som påvirker kilden. I vårt tilfelle vil det si at ved bruk av mikrofonmatrise er det ønskelig å detektere støykilden i boogien for å få en indikasjon på hvor feilen ligger. Dette skal gjøres ved først å sammenligne målinger for begge to alternativene. Deretter vil oppgaven være å finne en matematisk modell til mikrofonmatrisen som kan brukes for forbedre resultater.



(a) Utsiden av testbenken (1) (b) Utsiden av testbenken (2)



(c) Innsiden av testbenken (1) (d) Innsiden av testbenken (2)

Figur 1: Mantena sitt Dynamiske Testbenk innvendig og utvendig

Visjonen til denne oppgaven var å lage et system basert på en mikrofonmatrise og generere et støykart som viser hvor vibrasjoner oppstår og hvor støykilden befinner seg på boogien. Støykartet skal ha en høy oppløsning og gi en visuelt representasjon av feilen eller støykilden. Etter å ha gjort det skulle målingene fremskaffet fra testingen brukes til å sammenligne med målinger fra akselerometrene og sensorene.

Det var også ønskelig å utvikle et system hvor det ble brukt tre referanserammer. En av referanserammen skal inneholde 3D modell av maskineriet i testrommet, en referanseramme skal inneholde et 3D støykart og en referanseramme skal inneholde et 3D varmekart. Alle disse rammene skal modelleres på den måten at det vil være enkelt å navigere mellom dem.

Som tidligere beskrevet i rapporten var det ikke mulig å innhente data og målinger fra boogien. Hensikten med det skulle være å finne svar på om bruk av mikrofonmatrise var mer effektivt kontra bruken av akselerometer. I tillegg skulle mikrofonmatrisen og plasseringen av den i Dynamisk Testbenk tas med i betraktning. Dette var for å få best mulig resultat ved bruk av en mikrofonmatrise spesifikk for denne jobben. Siden alt dette hang sammen bestemte jeg og min veileder for å jobbe med plasseringen av mikrofonmatrisen i den Dynamiske Testbenken. Denne delen av oppgaven var like relevant for problemstillingen vi hadde i utgangspunktet. Dette førte til at det var fortsatt mulig å jobbe med oppgaven uavhengig av data for mikrofonmatrisen og data for akselerometer fra Mantena AS.



*I den neste delen av introduksjons kapittelet skal den nye oppgaven presenteres. Sammenhengen mellom begge oppgavene vil også bli forklart.*

Den nye oppgavens beskrivelse var å bruke et treghtsnavigasjonssystem (TNS) som var koblet til en mikrofonmatrise med et kamera. Det var ønskelige å simulere en mikrofonmatrise som beveget seg rundt massesenteret til boogien i form av en halvsirkel. Etter vår oppfatning var dette den mest gunstige banen å bevege på for å lokalisere støykilden som var detektert på boogien. Kameraet på matrisen skulle brukes til å fastslå orienteringen og posisjonen til mikrofonmatrisen i den Dynamiske Testbenken. I en ideell fremstilling ville bevegelsen rundt boogien være veldig enkel, men i virkelige verden må det tas høyde for støyen som påføres TNS. Det er også av interesse å se hvilken punkter i rommet kameraet skal bruke som referanse. I første omgang har vi brukt punkter i taket til den Dynamiske Testbenken. Grunnen til dette er at synsvinkelen til kamera vil ha et større område å forholde seg til. Noe som igjen ville ha gitt en fordel i form av at punktet eller punktene ikke beveget seg ut av bildeplanet. Dette ville ikke ha vært tilfelle hvis kameraet hadde brukt punkter som lå i det horisontale planet når vi beveget oss rundt boogien.

I kapittel 2 - *Bakgrunn og hensikt* - vil det bli forklart i detaljer hvordan dagens løsning blir brukt hos Mantena AS og hva slags fremgangsmåte jernbaneverksteder fra andre steder i verden bruker for å feilsøke på sine enheter. Det vil også bli gitt en overordnet beskrivelse av andre typer problemstillinger som er blitt løst med forskjellige mikrofonmatriser. Denne delen av informasjonen vil være med på å forme en del av løsningen som blir beskrevet i denne masteroppgaven.

I kapittel 3 - *Teori del 1 og del 2* - blir teorien og en del av den matematiske modellen forklart i detaljer. Dette kapittelet er delt i to deler for å skille mellom begge aspektene som det har blitt jobbet med, under hele masteroppgaven.

I kapittel 4 - *Metode for simulering - presentert med matematiske likninger* - vil det bli gitt et sammendrag av likningene som er blitt brukt i et systematisk og sammenhengende rekkefølge. Den matematiske algoritmen som er ønskelig for løsningen vil også bli listet opp i riktig rekkefølge for videre bruk i et simulerings program. Det vil også bli gitt en fremstilling av pseudo-koden som er brukt i Matlab for den viktigste delen av programmet.

I kapittel 5 - *Resultat og diskusjon* - får vi resultatene presentert som figurer og deres forklaringer. Deretter redegjøres og diskuteres det for utfordringer, løsninger og eventuelt mangler som har vært med på å forme løsningen. I tillegg til det diskuteres det hva som er blitt gjort og hva som kunne ha blitt gjort når resultatene ligger til grunn.

I kapittel 6 - *Konklusjon* - blir hele oppgaven oppsummert i korte trekk.

I kapittel 7 - *Veien videre* - tas det for seg hva som kan være ønskelig å gjøre videre med denne løsningen vi har kommet frem til og hvordan bygge videre på hele problemstillingen som har vært utgangspunktet for Mantena AS.

I kapittel A - *Matlab kode* - er det lagt ved hele Matlab-koden som er blitt brukt som vedlegg.

## 2 Bakgrunn og hensikt

I dette kapittelet vil vi gå gjennom eksisterende løsninger som blir brukt på samme problemstilling eller lignende andre steder i verden. Det vil også bli gitt beskrivelse av hvor og hvordan mikrofonmatriser blir brukt. Dagens løsning som blir brukt av Mantena AS fungerer, men er ikke optimal. For å beskrive den nye ideen vår som kanskje kan optimalisere dagens løsning må vi forstå hva den allerede eksisterende løsningen er. Den første delen av kapittelet er selvforklarende. Neste del handler om hvordan andre virksomheter har løst sine utfordringer av samme type problemstilling og hva slags teknologi som er blitt brukt.

Grunnen til denne tilnærmingen i rapporten er fordi det er ønskelig å finne ut om disse virksomhetene har en "state of the art" løsning. Det kan hjelpe oss med å finne ut om løsningen vi ønsker å bruke er optimalt og effektivt i motsetning til den som blir brukt på nåværende tidspunkt.

Utfallet av hva som bør brukes mellom eksisterende og den tiltenkte løsningen er et tema som vil bli belyst i slutten av oppgaven.

### 2.1 Dagens løsning brukt av Mantena AS

Fremgangsmåten for å teste boogiene hos Mantena AS er at det blir tatt en grundig frekvensanalyse og vibrasjonsanalyse av boogiene. Disse analysene er med på å estimere tilstanden til de forskjellige roterende delene som finnes på boogiene. Ved å estimere tilstanden på denne måten vil det gå an å finne ut hvor feilen ligger.

Metoden som blir brukt ved vibrasjonsanalyse er at det gjøres analyse av intensiteten til vibrasjoner i maskineriet. Ved denne analysen så kategoriseres maskineriet til "Safe to operate" eller "Vibrations causes damaged to components and machinery". Metoden som blir brukt ved hjelp av frekvensanalyse gjør det mulig å se en unik mønster eller signatur i frekvensresponsen for komponentene i boogiene. Disse unike mønstrene eller signaturene er med på å definere hvor feilen ligger ved de forskjellige roterende delene. Også vil frekvensspektrumanalyse være med på å gi en indikasjon på om det er en girskade eller lagerskade, og om eventuelt skaden er på en tidlig eller sen stadium.

Frem til nå har Mantena AS brukt disse to metodene kombinert for å finne skader eller feil på boogiene. Andre metoder som blir brukt som f.eks demontering av boogier også feilsøke, er både tid- og ressurskrevende. Vi vil komme nærmere på det i neste kapittel. Dagens løsning er bra, men ikke optimal sett fra et tidsperspektiv. Det tar ca. 2 timer å sette på sensorer på boogien og igjen ca. 2 timer å ta av sensorer fra boogien. I tillegg til dette brukes det opp mot 2 timer å feilsøke med forskjellige verktøyer imellom monteringen av sensorene. Mantena AS har et ønske om å forkorte denne tiden og effektivisere arbeidet med ny og fremtidsrettet løsning/løsninger.

### 2.1.1 Feilsøking med sensorer og akselerometer

Test prosedyren beskrevet i følgende rekkefølge hos Mantena AS:

1. Plasser boggi i testbenk
2. Kobl opp hastighetsmåler. Der brukes hastighetsmåleren som sitter på hjulsatsen (hastigheten brukes som referanse, fordi alle feilfrekvensene er gitt ut i fra rotasjonshastigheten til den aktuelle komponenten).
3. Kobl opp sensorer. SPM Shock pull sensorer og akselerometer (sensorene settes på en fornuftig plassering og nærmest mulig lageret man er interessert i).
4. Kobl opp AC eller DC motorer.
5. Start test.
6. Under testen kjøres boogien i lav hastighet (ca. 40-50 km/t) og høy hastighet (ca. 90-120 km/t), begge veier (fremover og revers).

## 2.2 Eksisterende løsninger hos andre aktører

Det er veldig viktig å se hvordan samme type utfordringer er løst av forskjellige aktører på markedet. Det kan være greit å sammenligne hvor tilfredsstillende den tiltenkte løsningen er i forhold til det som allerede finnes på markedet. Det er også ønskelig å vite om løsninger som er brukt av andre aktører kan være mer lønnsomt enn den vi har tenkt å lage. Det er en del forskjellige løsninger som finnes der ute fra en del forskjellige firmaer. Disse firmaene har et stort sortiment på forskjellige løsninger og produkter. Noen leverer integrerte løsninger, mens andre leverer deler av løsningen som kan brukes til denne oppgaven. I denne oppgaven har man valgt ut de selskapene som virker mest relevant for vår del.

### 2.2.1 Gatekeeper Intelligent Security [7]

Gatekeeper Intelligent Security er et amerikansk selskap. De har samarbeids partnere i 30 forskjellige lokasjoner rundt om i verden. Deres hovedfokus er sikkerhet ved bruk av hjelpemidler som maskinsyn og dyp læring. I vårt tilfelle kan de utnyttes som et godt eksempel fordi de bruker den samme teknologien til sikkerhet som de bruker for vedlikehold av understellet til tog.

### 2.2.1.1 Intelligent Train Understell Scanner (ITUS)

Intelligent Train Understell Scanner er et skanning system som brukes for å ta høyoppløste bilder av rullende togs understell. Det gir mulighet til å oppdage raskt fremmedlegemer, avvik og potensielle vedlikeholds farer under alle miljøforhold. Toget passerer over en dobbelvisnings skanner, som vist på figur 2, mens to høy oppløste, virtuelle 3D digitale bilder for hvert understell lager en digital "fingeravtrykk" av understellet. Dette fingeravtrykket blir sammenlignet med et referanse bilde som ligger i databasen for å sjekke om det er noen avvik på understellet. Hvis det blir oppdaget avvik blir det satt en rød prikk hvor avviket skal være og utløser en alarm. Alt dette kan sees på en høyoppløsningsskjerm med den rette applikasjonen laget av Gatekeeper Intelligent Security.



Figur 2: Dobbelt-visnings skanner

### 2.2.2 Lakesight Technologies [20]

Lakesight Technologies er en paraply organisasjon som har hovedbase i Munich, Tyskland. Organisasjonen har tre selskaper som samarbeider og jobber tverrfaglig innenfor fagfeltet maskinsyn. Visjonen til selskapet er å bli ledende innen for denne type teknologi. De tre forskjellige selskapene er:

1. *Tattile*: Har base i Italia. De produserer og selger maskinsyn systemer for industrielle og mobilitets applikasjoner.
2. *Mikrotron*: Holder til i Tyskland. De er kjent i det internasjonale markedet for å lage høyhastighets kamera for industriell bildeprosessering. Deres kameraer er kjent for bildeoppløsning, lysfølsomhet og robusthet. Disse kameraene brukes både i industrielle- og vitenskapelige applikasjoner.
3. *Chromasens*: De holder også til i Tyskland. Chromasens er en internasjonal anerkjent selskap som er ledende innenfor linjeskanning teknologi i 2D og 3D. Chromasens produkter er installert i alle type applikasjoner innenfor mat inspeksjon, utskrift inspeksjon, alt fra elektronikk og halvleder inspeksjon til sikkerhetsteknologi.

#### 2.2.2.1 Inspeksjon med bildeanalyse i 3D av rullende tog [19]

Fra damplokomotiv til dagens tog har det vært en lang reise. Og gjennom denne reisen har teknologien også tatt store fremskritt. Lake Sight Technologies har tatt et steg enda lengre enn det som forventes fra dagens løsninger for tog og togets vedlikehold. Det er ingen hemmelighet at et feilfritt

tog er den første og største prioritering til togoperatørene. Og med alle de store og komplekse jernbane nettverkene, som i tillegg har flere tog kjørende på mindre sykluser på disse nettverkene, er det ingen ting som overlates til tilfeldigheter.

Det vanligste og gjennomgående konseptet som har pågått i alle år for vedlikeholdsarbeid på jernbanevogner har vært stasjonære diagnostisering systemer. Diagnostisering teknologi har spilt en viktig rolle i å effektivisere inspeksjons- og vedlikeholdsarbeid på jernbanevogner. Det har vært standarden for vedlikehold av jernbanevogner, men parallelt med å bruke denne standard måten har utviklingen i en digital informasjonsteknologi æra gitt rom for nye innovative løsninger. Det er blitt laget integrerte diagnostiske systemer om bord på tog for skadeteleksjon. Lake Sight Technologies har utviklet et høy presisjons 3D-bilde opptakssystem som gir en form for vedlikeholdsmuligheter. Dette opptaket blir gjort av en Chromasens skanner boks med 3D linjeskannings kamera og LED belysningsystem, se figur 3.



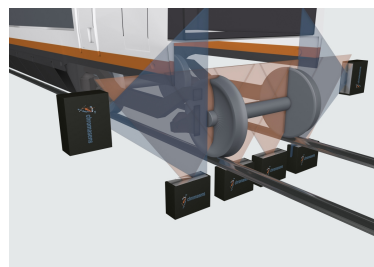
Figur 3: Chromasens skanner boks

Chromasens skanningssystem, legger til en annen viktig komponent i den allerede implementerte diagnostiske teknologien - den gjør en nøyaktig stereoskopisk visuell inspeksjon av understellet og deres sidevisninger, til høyre og venstre som vist på figur 4a og 4b.

Denne fremgangsmåten med Chromasens skanner viser seg å være en veldig praktisk og effektivt måte for å ta bilder med høy presisjon og oppløsning. Den største fordelen med denne metoden er at toget ikke må tas ut av drift og plasseres i et verksted som vist med figur 4. Digital skanning kan gjøres hvor som helst på skinnegangene langs tog ruten. Det eneste toget må gjøre er å passere disse boksene med vanlig cruising hastighet. Den stereoskopiske inspeksjonen med høy oppløsning registrerer ulike potensielle feil. Det kan være feil på koblinger, slanger, trykkbeholdere, hjul og eventuelt løse skruer som forblir ubemerket i et 2D-bilde.



(a) Inspeksjon gjort med Chromasens skanner fra kanter



(b) Nærbilde av inspeksjon gjort med Chromasens skanner

Figur 4: Chromasens skanner plassert på togskiner

### 2.2.3 Baker Hughes a GE company [2]

*Baker Hughes a GE company* er et selskap som tilhører GE (General Electronics) familien. Dette er en velrenommert og respektert selskap i den industrielle verden. GE er en paraply organisasjon hvor BHGC - Baker Hughes a GE company er et datterselskap. BHGC reklamerer for å dekke alle forskjellige felt i teknologi og industri. On-shore, off-shore og subsea er noen av disse feltene. De er det første og eneste selskapet i verden som jobber med alt fra sky-applikasjoner til undervannsløsninger. Deres portefølje er veldig stor og derav en lang liste med forskjellige produkter på forskjellige felt. En av deres produkter heter *GE WheelStar - Ultrasonic Rail Road Wheel Inspection Solutions*. Dette produktet er den nærmeste løsningen på markedet sammenlignet med det Mantena AS bruker idag. Demonstrasjon av dette produktet vises i denne videoen: GE - Phased Array Ultrasonic Testing of Railroad Wheels på youtube [27].

#### 2.2.3.1 GE WheelStar [2]

Som det ble nevnt i tidligere kapittel, har reisen vært lang for det som er dagens jernbane rundt om i verden. Med tiden har krav om sikkerhet og diverse andre spesifikasjoner økt i samsvar med størrelsen og hastighet på togene. Jernbaneoperatører, produsenter og verksteder har sikkerhet som en meget viktig prioritet. For å holde høy produktivitet og kvalitet brukes det stadig ikke-destruktive testløsninger. Hjul og aksler på tog er de mest sikkerhets kritiske elementer i kjøretøy da de er utsatt for høy stress med økende hastigheter og belastninger. Globale jernbanestandarder krever at disse delene blir inspisert ved hjelp av ultralyd etter produksjon og under regelmessig vedlikehold. GE Inspection Technologies tilbyr to forskjellige metoder for inspeksjon av hjul og aksler. Begge varianter har en robust mekanisk konstruksjon og bruker en kombinasjon av lineær faset matriser prober (for å kartlegge slitasje og form inspeksjon) og konvensjonelle transdusere for å utføre full kontroll av hjulet i en enkelt rotasjon. GE Inspection Technologies har tre typer metoder som blir brukt for inspeksjon av hjul og aksler.

##### 2.2.3.1.1 WheelStar - Gantry: for heavy maintenance:

GE Inspection Technologies sitt WheelStar - gantry: for heavy maintenance (figur 5) verktøyet er den klassiske løsningen for inspeksjon under tung og vanskelig vedlikehold. Begge hjulene blir testet for korte sykluser og samtidig blir bevegelses områdene på aksler også inspisert i samme testsyklus. Den har en høy inspeksjons produktivitet ved bare en enkel rotasjon. Dette er en veldig robust metode. Digitale inspeksjons data kan lagres for fremtidige referanser og aksel inspeksjon er en integrert løsning.



(a) Hjul plassert på en stasjonær stasjon

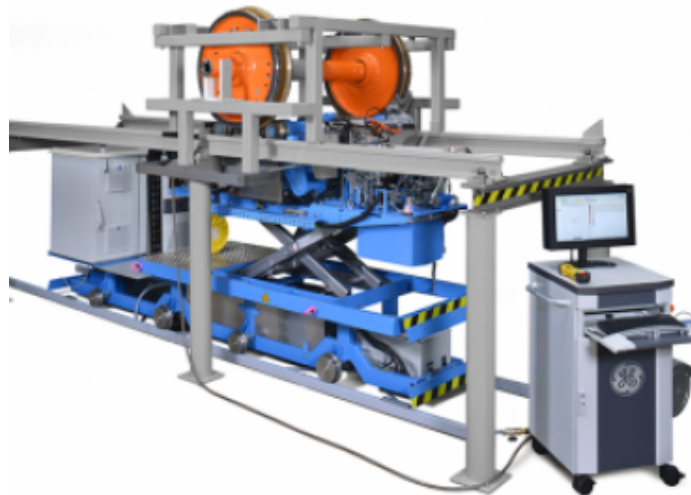


(b) Nærbilde av hjul plassert på en stasjonær stasjon

Figur 5: WheelStar - gantry: for heavy maintenance

### 2.2.3.1.2 WheelStar - Underfloor (UFPE):

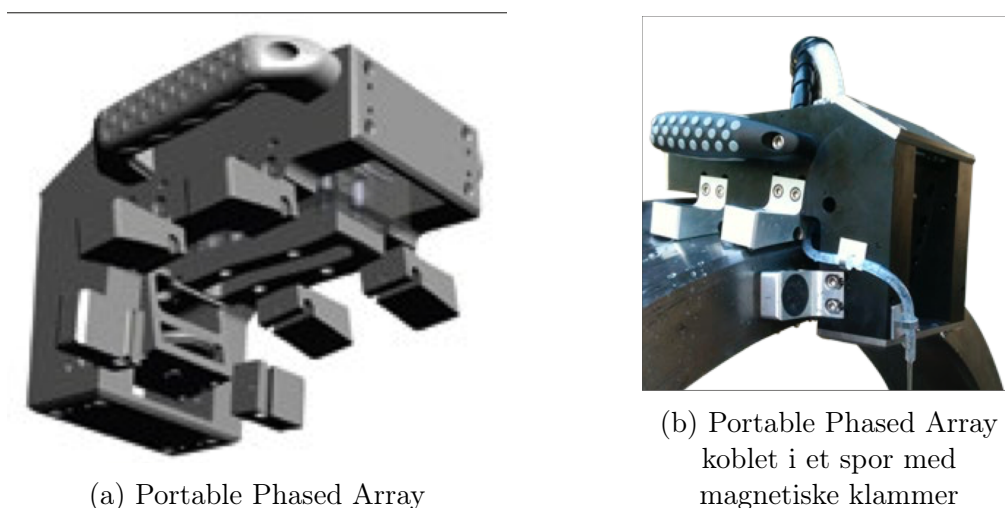
WheelStar - Underfloor (UFPE) metoden som vist på figur 6 er den optimale produktive løsningen for å gjøre service av understellet til høyhastighetstog. Med denne metoden er det ikke behov for å fjerne hjulsettet fra toget før man kontrollerer hjul, aksler og eventuelt andre komponenter. Hjulsettet løftes fra skinnen og roteres under testen med en integrert mekanisk enhet. En kombinasjon av enkle krystall- og phased array transdusere sørger for ideell tilpasning til de inspiserte områdene. Resultater visualiseres ved hjelp av et program og en enhet som er koblet til rampen. Hele denne prosessen maksimerer produktiviteten. Inspeksjonen kan gjøres helt automatisert. Noe som kan brukes til automatisk feil gjenkjenning som igjen reduserer menneskelig faktorer/feilkilder.



Figur 6: GE Krautkramer WheelStar™ In-Service Train Wheel Inspection System

### 2.2.3.1.3 WheelStar - Portable Phased Array:

GE sitt WheelStar - Portable Phased Array (figur 7a) er laget for å være et bærbar produkt som blir brukt for å inspisere hjullager og lageret på akslene. Dette produktet er en kostnads effektiv løsning for butikker med lavt produksjonsvolum og alternativ verktøy for butikker som bruker automatiserte inspeksjons løsninger. Den har integrerte magnetiske klammer som festes på sporet til hjulet og gir enkel håndtering som vist på figur 7b. Den har phased array sonde innebygd i en mekanisk armatur som garanterer stabile koblings forhold. Med Portable Phased Array så kan det utføres et mobilt ultralyd inspeksjon av hjullageret. Den er enkel å håndtere og krever ikke mye opplæring.



Figur 7: WheelStar - Portable Phased Array

## 2.3 Forskjellige anvendelser av mikrofonmatriser

I dette kapittelet skal vi gå igjennom hvilken sammenheng mikrofonmatriser anvendes i dagens samfunn. I senere tid er bruken av mikrofonmatriser blitt en velkjent teknologi. Du vil se denne type teknologi i forskjellige sammenheng. Felles for disse er å få tatt opp optimal lyd i krevende forhold. Det blir stadig mer utbredt å bruke mikrofonmatriser hvor det ønskes å få mest mulig autentisk lyd som et innsignal når støyen til omgivelsen har en påvirkning på lyden.

Før man forklarer om hvor og hvordan mikrofonmatrise brukes, er det veldig viktig å forstå hva en mikrofonmatrise egentlig er. I dette kapittelet vil det bli forklart grundig hva en mikrofonmatrise er og hva den gjør. Når man skal beskrive mikrofonmatrise kommer man ikke utenom å drøfte forskjellige kriterier som må oppfylles for at det skal være en velfungerende mikrofonmatrise.



### 2.3.1 Hva er en mikrofonmatrise? [21]

En mikrofonmatrise er en enhet som fungerer akkurat som en hvilken som helst annet enkeltstående mikrofon. Det som utgjør forskjellen er at det er flere mikrofoner som er blitt satt sammen i et spesifikk mønster. I motsetning til en enkeltstående mikrofon vil alle mikrofonene i en mikrofonmatrise ta opp lyd i fellesskap som et enkelt inngangssignal. Størrelsen på en mikrofonmatrise kan være alt fra 2 mikrofoner til flere tusener. Størrelsen avhenger veldig mye av formålet den skal brukes til og omgivelsene den skal brukes i.

Den største mikrofonmatrisen som finnes i verden idag er blitt laget av et selskap som heter Sorama i Nederland. Det er satt sammen av 4096 individuelle mikrofoner med en “*aperture*” på 128x128 cm. Dette gjenspeiler hvor små disse mikrofonmatrise enhetene kan være i størrelse, sett det i sammenheng med antall mikrofoner. Denne mikrofonmatrisen er en demonstrasjon av et “lyd kamera” og formålet med den er å kalkulere og lage et 3D kart av lyd kilder [8].

En mikrofonmatrise er bygd opp som en enhet hvor alle mikrofoner jobber sammen for å spille inn lyd samtidig. Størrelsen på en mikrofonmatrise kan som nevnt tidligere i avsnittet ovenfor være veldig forskjellig. Det vil først og fremst avhenge av hvor stort område det er ønskelig å dekke av omgivelsene og hva som er kilden for lyden som man ønsker å fokusere på. Det må også tas med i betraktning at støykildene som er ønskelig å filtrere ut ved opptak av lyden er et inngangssignal. Enkleste formen av en mikrofonmatrise vil i først omgang bestå av 2 mikrofoner. En mikrofon plassert på venstre siden av enheten og en mikrofon plassert på høyre siden av enheten, som vist på figur 8



Figur 8: Simpleste formen av en mikrofonmatrise med en mikrofon plassert på venstre siden og en på høyre siden

Effekten som lages ved å plassere mikrofonene på denne måten vil være at når du tar opp lyden i f.eks et rom, vil lyden bli tatt opp fra hver sin del av rommet. Det vil igjen føre til at det dynamiske stereoopptaket vil fremstå mer eller mindre som et surround lyd. Når dette opptaket spilles igjen på et sett med hodetelefoner vil lyden til venstre siden og høyre siden være tydelig forskjellig og det kan høres merkbart.

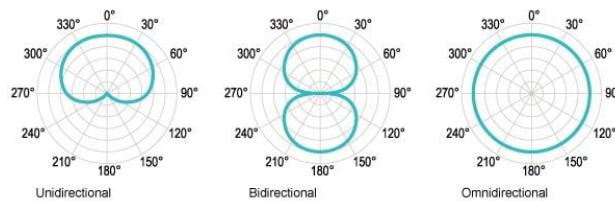
Det viktigste i en mikrofonmatrise er at alle mikrofonene må være like. Det vil si at modellen og produsenten må være den samme. Grunnen til dette er å få et godt opptak av lyden. Hvis mikrofonene er ulike vil det føre til at en mikrofon i forhold til den andre kan ha større forsterkning.

Mikrofonene kan også ha forskjellige faser ved opptak av lyden som igjen vil føre til at resultatet blir ulik enn hva det er ønskelig i utgangspunktet. Ved å bruke forskjellige mikrofoner vil det også føre til at en mikrofon tar opp lyd fra alle retninger, mens den andre tar opp fra en bestemt retning. Alle disse problemstillingene som er nevnt ovenfor kan gi fryktelige konsekvenser i en lyd-applikasjon. Derfor er man veldig nøye med valget av mikrofoner i en mikrofonmatrise. Mikrofonene må være tilsvarende like og oppfylle en rekke spesifikasjoner, slik at det ikke blir ujevn lydopptak.

For at mikrofoner skal være like i en mikrofonmatrise må tre kriterier oppfylles. Disse kriteriene er *retning*, *følsomhet* og *fase*.

### 1. Retning (directionality):

Med retning menes det hvilken retning mikrofonen tar opp lyden fra. Mikrofoner er laget for å ta opp lyd fra spesifikke retninger når det blir tatt opp lyd med dem. Noen mikrofoner kan bare ta opp lyd fra en retning, de blir kalt for *unidirectional microphones* (*enveis mikrofoner*). Noen mikrofoner plukker opp lyden fra to retninger, det vil si enten foran og bak eller venstre og høyre. Disse type mikrofoner blir kalt *bidirectional microphones* (*toveis mikrofoner*). Den siste typen av mikrofoner plukker opp lyden fra alle retninger og blir kalt for *omnidirectional microphones* (*multiretnings mikrofoner*). Figur 9 viser retningsegenskapene til en mikrofon.



Figur 9: De tre forskjellige retnings egenskapene av en mikrofon

Når det lages en mikrofonmatrise er det veldig viktig at alle mikrofoner i matrisen har opptak av lyd fra samme retning. Utenom noen unike tilfeller er det veldig lite ønskelig å ha mikrofoner som har forskjellige karakteristika i en mikrofonmatrise. For eksempel, hvis den ene mikrofonen plukker opp lyd bare fra en retning i matrisen og den andre mikrofonen plukker opp lyd fra alle retninger vil det føre til dårlige og ubalanserte lydopptak. Av den grunn blir nesten alle mikrofonmatriser laget med mikrofoner med samme retningsegenskaper.

### 2. Sensitivitet (sensitivity):

En annen faktor for at mikrofonene må være like i en mikrofonmatrise er sensitivitet. Sensitivitet er definert som forsterkningen til mikrofonen når den tar opp lyd. Det er ikke ønskelig å ta opp lyd når sensitiviteten til mikrofonene er forskjellig. Grunnen til det er at hvis den ene mikrofonen tar opp lyd med større sensitivitet i forhold til de andre mikrofonene når de er av forskjellige modell, vil opptaket av lyden være ulik som igjen vil føre til ubalansert lydopptak. Derfor er vanligvis den maksimale sensitiviteten i en mikrofonmatrise tillat til å være  $\pm 1.5$  dB slik at det ikke blir noe større differanse enn 3 dB sensitivitet for mikrofonene.

### 3. Fase (phase):

Til slutt er det viktig at fasen til enhver mikrofon må være lik for alle mikrofoner i en mikrofonmatrise. Det er denne premissen som blir brukt som referanse i tid når mikrofonen begynner å ta opp lyd. Det vil si at fasen definerer tiden for når alle mikrofonene skal starte og slutte å ta opp lyden i en mikrofonmatrise. Hvis fasen er forskjellig, vil dette føre til at opptak av signalene vil også komme inn på forskjellige tidspunkter. Dette er ikke ønskelig siden dette vil føre til usynkronisert opptak. Det som er ønskelig med lik opptak av signaler på samme tid, er at det ikke blir forsinkelser mellom signalene. Akkurat som sensitivitet for mikrofoner, må det også være en maksimal feiltoleranse for fasen mellom mikrofonene. Denne differansen er vanligvis på  $\pm 1.5$  grader for å sikre at opptaket til signalene skjer samtidig, som fører til harmonisert opptak.

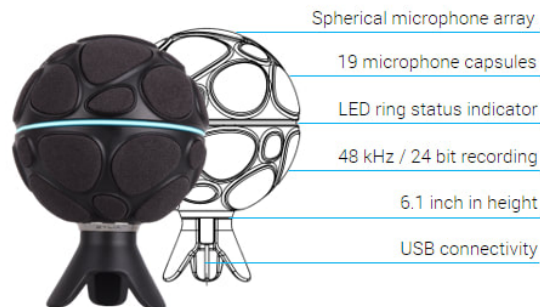
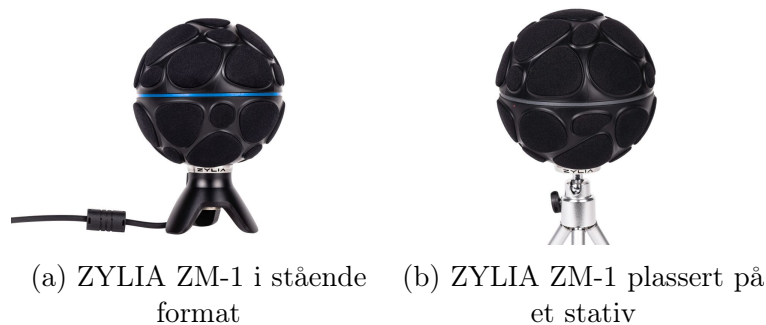
Mikrofonmatriser har begynt å få veldig mye popularitet i lydindustrien og hvor lyd er i fokus. En av grunnene til det er at det er blitt mye enklere å ta opp lyd dynamisk med mikrofonmatriser. Dette fører til at egenskapene til lydopptak blir veldig likt et surround system. Det er faktisk ingen begrensninger på antall mikrofoner som kan brukes og hvordan lyden kan synkroniseres for å lage dynamiske opptak. Det finnes mange forskjellige mikrofonmatriser og bruken av disse forskjellige mikrofonmatrisene avhenger av hva som er formålet. Desto mer teknologien går fremover, mer robust og praktisk blir bruken av mikrofonmatriser.

*I de neste kapitlene vil det bli gitt en beskrivelse av hvor disse forskjellige mikrofonmatrisene tas i bruk.*

### 2.3.2 Mikrofonmatrise brukt i lyd studio

Som nevnt tidligere i denne rapporten er det stor etterspørsel for å bruke mikrofonmatrise i omgivelser der lyd er i fokus, og et lydstudio er ingen unntak. For å få et originalt og autentisk lydopptak, hvor flere lydkilder (musikkinstrumenter) er i bruk samtidig, er bruken av mikrofonmatrise revolusjonerende. Muligheten til å gjøre opptak av flere lydkilder samtidig og skille dem fra hverandre, gir en spesiell opplevelse for den som lager musikken og den som hører på sluttproduktet. De forskjellige lydene som genereres av forskjellige musikkinstrumenter blir fremhevet, noe som gir mer klarhet i vokalene.

Det finnes mange forskjellige type mikrofonmatriser som brukes i lydstudio pr. dags dato. ZYLIA ZM-1 er en av dem. ZYLIA ZM-1 er laget av et selskap som har hovedkontor i Polen og heter Zylia [36]. Denne enheten er et godt eksempel på hvor enkelt og fleksibelt en mikrofonmatrise kan lages for lydopptak med flere kilder i musikk sammenheng, se figur 10. Det er blitt brukt 19 MEMS-mikrofoner i denne enheten. MEMS-mikrofoner er en type mikrofoner som er veldig robuste og billige når man ønsker å lage mikrofonmatrise. Vi kommer tilbake til MEMS-teknologi litt senere i oppgaven.

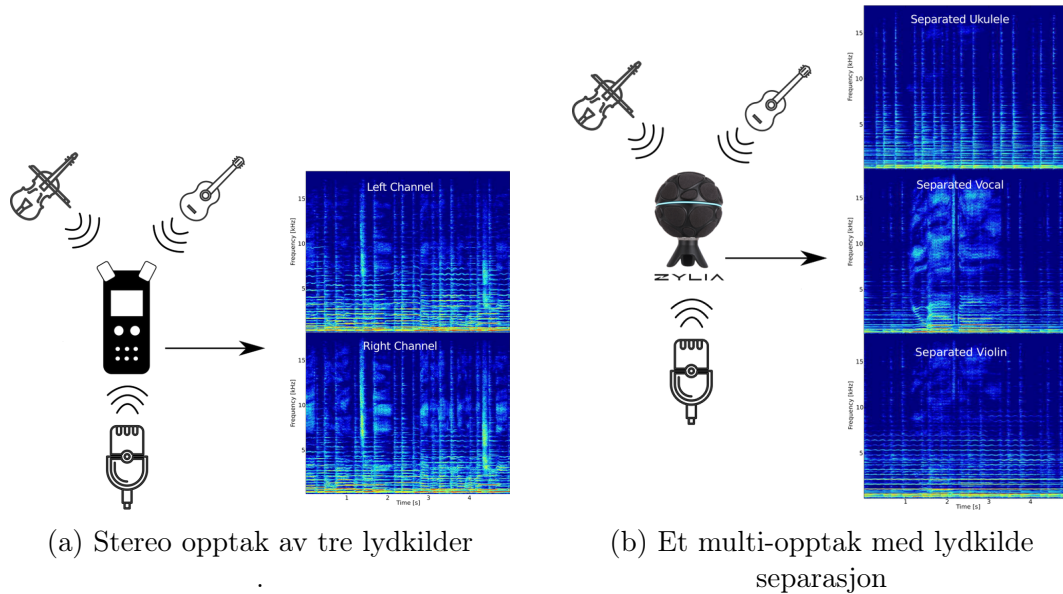


(c) ZYLIA ZM-1 med spesifikasjoner

Figur 10: ZYLIA ZM-1 mikrofonmatrise

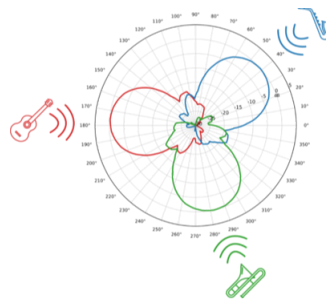
Sett med dagens teknologi, vil en mobiltelefon være tilstrekkelige for å ta opp lyd av flere lydkilder. Problemet med dette vil være at når det brukes en mobiltelefon, kommer opptaket til å være et blanding av alle lydkildene. Det er ikke mulig å skille mellom disse lydkildene i en tids-frekvens bilde som vist i figur 11a. Med slike opptak kan du ikke endre balansen til de individuelle lydkildene og lydredigering etter produksjonsprosessen er ekstremt begrenset.

Med en mikrofonmatrise av type ZYLIA ZM-1 eller lignende enheter, vil det åpnes for andre og mer avanserte muligheter. Dette vil først og fremst gi mulighet for å manipulere balansen og nivået av lydkilder som vil bli tatt opp. Også vil det være mulig å skille mellom de forskjellige lydkildene og bestemme intensiteten til hver kilde som vist på figur 11b i en tids-frekvens sammenheng.



Figur 11: Stereopptak med mobil telefon vs ZYLIA ZM-1 mikrofonmatrise

Figur 12 er et eksempel på hvordan lyden blir oppfattet og tatt opp fra forskjellige retninger med en ZYLIA ZM-1 mikrofonmatrise.

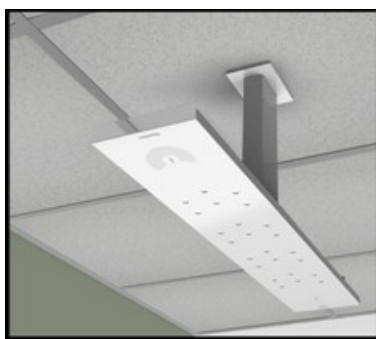


Figur 12: ZYLIA ZM-1 mikrofonmatrise med opptak av forskjellige lydkilder fra forskjellige retninger

### 2.3.3 Mikrofonmatrise brukt i forelesningssal eller konferanserom

Bruken av mikrofonmatrise begynner å bli vanlig i forskjellige sammenheng. Disse blir blant annet brukt i forelesningssaler, konferanserom, møterom og diverse steder hvor to til flere personer er samlet. Fordelen ved bruk av mikrofonmatrise på disse plassene gjør at all støy og ekko/etterklang som lages i omgivelsene kan filtreres ut. Mikrofonmatrisen kan ha totalt fokus på personen som snakker, uansett hvor man skulle befinne seg i rommet. Det fører til at personer på andre enden vil høre vedkommende veldig klart og tydelig. På denne måten vil samtalen både oppleves som naturlig og autentisk. Måten det gjøres på er forskjellig, men den matematiske algoritmen og fremgangsmåten er den samme for alle tilfellene nevnt ovenfor.

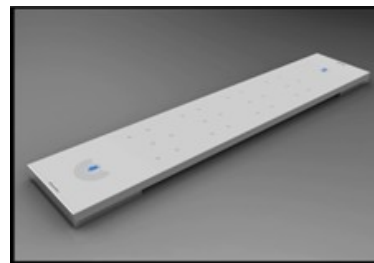
ClearOne [4] er et selskap som har spesialisert seg på forskjellige løsninger innenfor lydkonferanser, visuell samarbeid og nettverksbaserte streaming løsninger. En av deres produkter er beamforming mikrofonmatrise. Denne mikrofonmatrisen kan plasseres på tre forskjellige måter i et rom som vist i figur 13.



(a) ClearOne beamforming mikrofonmatrise festet i taket



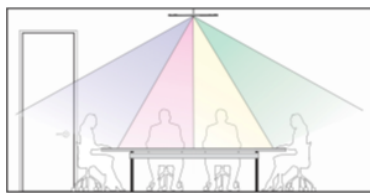
(b) ClearOne beamforming mikrofonmatrise festet til et vegg



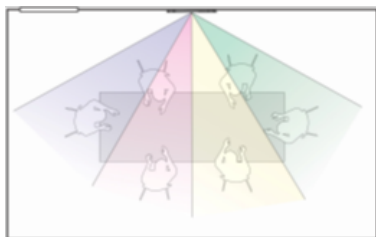
(c) ClearOne beamforming mikrofonmatrise plassert på et bord

Figur 13: ClearOne beamforming mikrofonmatrise

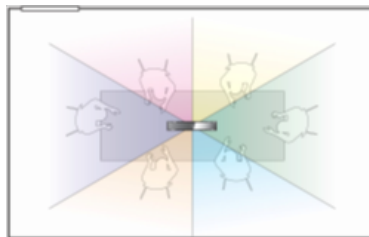
Beamforming er en måte hvor det kombineres signaler fra mange forskjellige mikrofoner i en matrise til å lage et mønster som er fokusert på å plukke opp lyd fra en kilde. Strålen retter seg automatisk etter den som snakker i rommet og avviser uønsket lyd og refleksjon. Denne fremgangsmåten er også med på å redusere den hule lyden og romstøyen som pleier å oppstå ved samtaler i et åpent rom. Beamforming metoden reduserer begrensninger på høyde (figur 14a), avstand (figur 14b) og vinkel (figur 14c).



(a) Beamforming med mikrofonmatrise som er plassert i et tak



(b) Beamforming med mikrofonmatrise som er plassert i en vegg



(c) Beamforming med mikrofonmatrise som er plassert på et bord

Figur 14: Virkemåten til beamforming metoden i en mikrofonmatrise

### 2.3.4 Sorama [29]

Sorama er et Nederlandsk selskap som er blitt nevnt tidligere i denne rapporten. De har vært en ledende og anerkjent aktør innen lyd og bilde. Dette selskapet er holder av Guinness verdens rekord [8] av den største mikrofonmatrisen som er blitt laget. Deres spesialitet er optikk og akustikk kombinert for å finne støy kilder. Sorama lager optiske mikrofonmatriser som detekterer støy og vibrasjon ved forskjellige søkeområder. Data som blir samlet fremstilles visuelt og deretter analyseres for å finne løsninger som reduserer støy.

Noen av Soramas bruskområder er listet nedenfor.

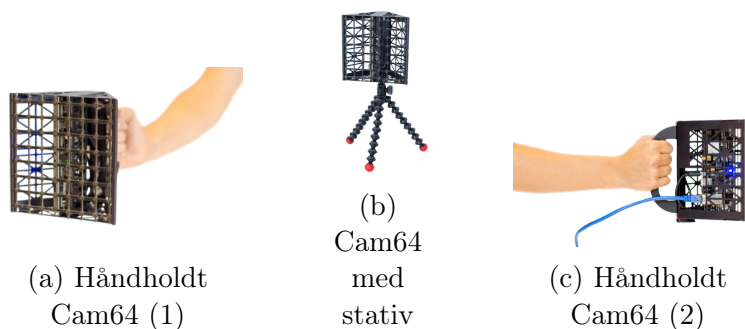
- Bygninger.
- Medisinsk utstyr.
- Automotive objekter.
- Industriområde
- Offshore

Sorama har ferdig komponerte produkter, Sorama CAM64 og Sorama CAM1K. De lager også mikrofonmatriser med kameraer etter forespørsel. I denne oppgaven tar vi for oss bare en kort gjennomgang

av produktene og deres spesifikasjoner. Hensikt er å vise at Mantena AS ved behov kan få tak i løsninger på andre måter enn å lage dem selv. Det kan spare Mantena AS for mye ressurs bruk.

### 2.3.4.1 Sorama Cam64 [31]

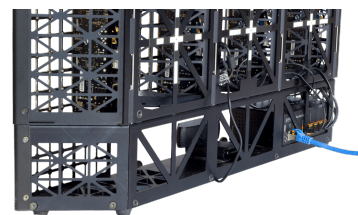
Cam64 (Figur 15) er en optisk mikrofonmatrise som er laget av 64 MEMS-mikrofoner og en integrert HD kamera. Størrelsen på arrayet er 16x16 cm. Den har et frekvensområde mellom 300Hz - 20KHz. Den fungerer for både near-field og far-field lyd- og bildebehandling. Det optimale frekvensområdet vil variere litt mellom disse to metodene, men den vil fortsatt gi en uovertruffen romlig nøyaktighet. Det er fullt mulig å koble denne mikrofonmatrisen til en Ethernet-kontakt slik at dataen kan overføres med høye hastigheter og prosesseres on-line i sanntid.



Figur 15: Sorama Cam64

### 2.3.4.2 Sorama Cam1K [30]

Cam1K (Figur 16, 17) er bygd med samme prinsipper som Cam64. Forskjellen som utgjør mellom disse mikrofonmatrisene er allikevel veldig stor. Mikrofonmatrisen har en størrelse på 64x64 cm. Cam1K har 16 ganger flere MEMS-mikrofoner. Den er 30 ganger raskere med prosessering av overflater ved near-field metoden som er på størrelsen med matrisen. Med far-field metoden kan matrisen observere 4 ganger lavere frekvenser ved å starte på 50Hz og oppover der hele frekvensområdet som blir dekket fra matrisen er 1Hz-20KHz.



Figur 16: Sorama Cam1K sett fra baksiden

Det er viktig å bemerke at begge disse optiske mikrofonmatrisene kan brukes på alt fra små flater til store konstruksjoner for å forminske støy. Det finnes nesten ingen konstruksjon laget av mennesker i denne verdenen som ikke genererer støy. WHO (World Health Organization) har sagt at konsekvensen av langvarig eksponering for lydforurensning er potensielt skadelig for mennesker. Dette viser igjen betydning for hvor samfunnsnyttig det er å bruke denne type teknologi for å forbedre levevilkår.





Figur 17: Sorama Cam1K sett forfra

### 2.3.5 Brüel & Kjær [3]

I en verden av lyd og vibrasjon vil du ikke komme unna det faktumet at hvis du bruker en eller annen objekt som er relatert til lyd og vibrasjon, vil det med stor sannsynlighet komme fra sortimentet til Brüel og Kjær sine produkter. Selskapet har de mest avanserte og robuste mikrofonmatrisene som finnes på markedet. Deres teknologi på feltet og tilleggsprodukter til mikrofonmatriser skiller seg ut fra deres konkurrenter. I internasjonalt sammenheng så er selskapet på et langt mer avansert stadium enn sine konkurrenter (denne påstanden baserer seg på undertegnede egne observasjoner gjort under denne masteroppgaven). Ikke bare har selskapet og deres produkter et godt omdømme, de dekker også alle felt relatert til analysering av lyd og vibrasjon. De har “state of the art” løsninger på alle sine produkter og i tillegg til det så utvikler de nye produkter hele tiden innen for sitt fagfelt for å ligge i topp sjiktet. Generelt pleier et selskap å ha et felt eller et produkt de spesialiserer seg på. Det kan også være at de fokuserer på en viss type kundekret, men dette er ikke tilfelle med Brüel og Kjær. Brüel og Kjær sine løsninger dekker forskjellige felt med en veldig stor overbevisning. Deres portefølje er veldig stor og dekker alle typer aktører i samfunnet som ønsker lyd- og vibrasjonsanalyse verktøy. Brüel og Kjær har ferdig utviklet løsninger, men lager også skreddersydde løsninger for sine kunder.

Valg av mikrofonmatrise bør basere seg på disse egenskapene:

- Omgivelsene (innendørs, utendørs, undervanns).
- Frekvensområde og størrelsen av det som er av interesse.
- Lyd-, støy- eller bildeoppløsning som kreves i et støykart.
- Avstanden til kilden (near-field eller far-field).
- Hvordan lydkilden oppfører seg (stasjonært eller bevegelig).

Hvis man er usikker eller trenger hjelp med å velge hvilken av egenskapene som passer ditt behov, så vil Brüel og Kjær dele sine erfaringer med deg for å finne den beste løsningen som passer. Det kan være både tid og ressurs sparende.

Vi skal se på løsninger som Brüel og Kjær produserer under sjangeren mikrofonmatriser og deres anvendelse på deres respektive felt.

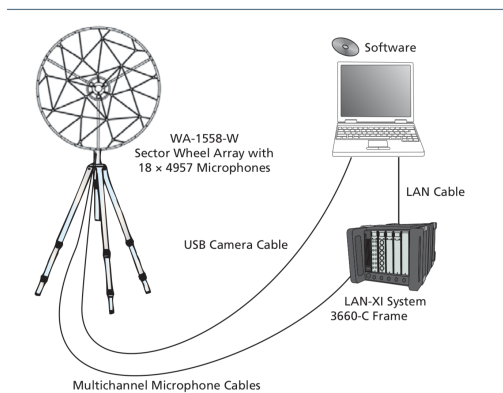
### 2.3.5.1 Sfærisk mikrofonmatrise:

Bruksområde: Innendørs.

Sfærisk stråleforming gir en komplett multi-retnings støykart i en hvilken som helst akustisk miljø, basert på en enkel måling. I motsetning til andre metoder der bare en del av støykartet blir kartlagt så gjør sfæriske mikrofonmatrise en utmerket jobb ved å dekke hele område. Den fanger støy fra alle retninger og samtidig tar bilder med 12 kameraer som er innebygd i den sfæriske kulen. Bildene blir deretter brukt for å vise bakgrunnen til støykartet for å få en riktig og mer presis orientering til støy- kilde eller kildene. Denne type løsninger blir brukt i både åpent terreng og omgivelser hvor det gir etterklang av lyden. Sfærisk mikrofonmatrise er et verktøy som er godt egnet for å lage et støykart slik at den kan analysere halvdempende rom, fly og kjøretøy. Sfærisk mikrofonmatrise er vist med figur 18.



(a) Sfærisk mikrofonmatrise i bil



(b) Oppsett av Sfærisk mikrofonmatrise



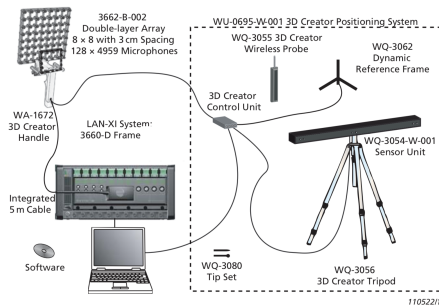
(c) Sfærisk mikrofonmatrise

Figur 18: Sfærisk mikrofonmatris

### 2.3.5.2 Håndholdt mikrofonmatrise

Bruksområde: Innendørs.

Håndholdt mikrofonmatrise måler lydfelt som er veldig tett opp mot lydkilden. De blir hovedsakelig brukt på områder hvor plass og rom er begrenset for å komme frem til lydkilden. Det kan være alt fra lekkasjedeteksjon og konform avbildning av innsiden til et kjøretøy eller nær kompliserte maskiner. Bildene på figur 19 viser hvor liten og praktisk enheten er for sitt formål. I tillegg til det kan det sees fra figur 19b hvordan muligheten er for å koble til områder som er vanskelig å komme til ved hjelp av en ledning og med en sensor på enden av ledningen.



(a) Oppsett av håndholdt mikrofonmatrise



(b) Håndholdt mikrofonmatrise brukt på en gressklipper



(c) Håndholdt mikrofonmatrise



(d) HAND-HELD ANALYZER TYPE 2250-S

Figur 19: Forskjellige fremstillinger av håndholdt mikrofonmatriser

### 2.3.5.3 Mikrofonmatrise i form av et gitter

Bruksområde: Innendørs.

En gitter basert mikrofonmatrise brukes vanligvis på testobjekter som skal måles på stasjonære lydfelt. Dette er en type som blir brukt med roboter (figure 20b) og er ekstremt nøyaktig på å kombinere punkter som blir målt.



(a) Stående gitter mikrofonmatrise



(b) Grid mikrofonmatrise med 2D-robot

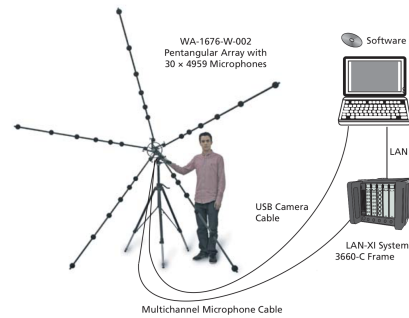
Figur 20: Grid basert mikrofonmatriser

### 2.3.5.4 Sammenleggbar mikrofonmatrise

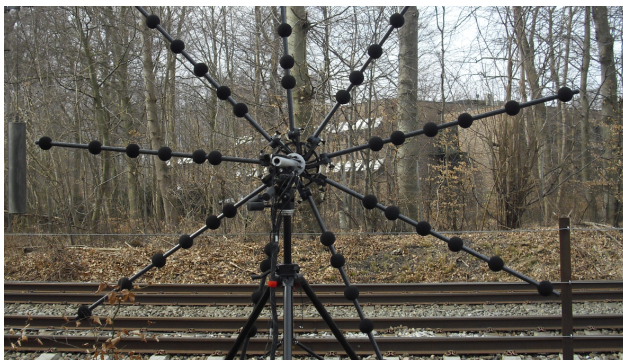
Bruksområde: Utendørs.

For store lydkilder/støykilder som befinner seg utendørs er sammenleggbar mikrofonmatrise den mest optimale løsningen. Siden den kan "brettes" sammen vil den være veldig praktisk med tanke på transportering. Denne mikrofonmatrisen kan være i flatformet eller i en form hvor armer som stikker ut i en viss vinkel som vist på figur 22a. Denne formen gir en traktformet rekkefølge av mikrofonene som gjør at eksternt støy fra baksiden av arrayet kan undertrykkes med opptil 10dB, avhengig av frekvensen. Et veldig viktig egenskap som du finner ved

sammenleggbar mikrofonmatrise er at den har mulighet til å utnytte refleksjoner i bakken. Denne metoden gir mulighet til å foreta målinger ved siden av et kjøretøy som kjører forbi som vist på figur 22b eller en høyhastighetstog som er rullerende som vist med figur 22c. Bruk av matrisen i en vindtunnel er annet alternativ.



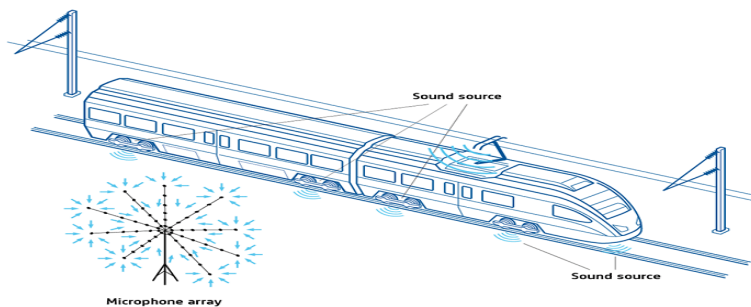
Figur 21: Oppsett for sammenleggbar mikrofonmatrise



(a) Sammenleggbar mikrofonmatrise med armer og i traktform



(b) Halv-formet sammenleggbar mikrofonmatrise brukt på bil



(c) Mikrofonmatrise brukt på rullerende tog

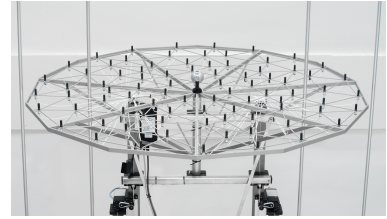
Figur 22: Sammenleggbar mikrofonmatrise



### 2.3.5.5 Hjulformet mikrofonmatrise

Bruksområde: Innendørs og utendørs.

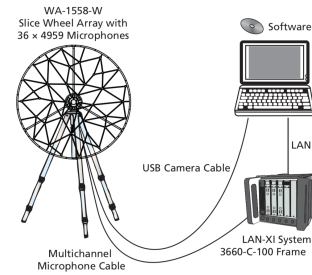
Mikrofonmatrise som er formet som et hjul (figur 23 og 24 ), er en type mikrofonmatrise som finnes i forskjellige størrelser. Noen er laget med stor diameter og noen med liten diameter. Størrelsen til mikrofonmatrisen er med på avgjøre antall mikrofoner det skal være med i matrisen. Denne type enheter er blitt laget for å være brukervennlige og lett å håndtere. Dette fører til at resultatene som produseres er også ganske optimale. Hvis enheten blir plassert veldig nær kilden så er det mulig å ta akustisk holografi av målingene, men den er egentlig brukt for å hente inn måledata med beamforming (stråleforming).



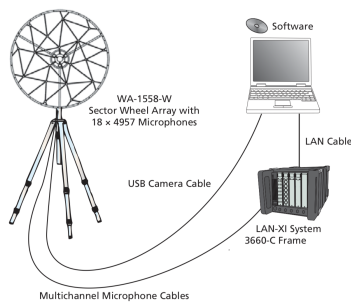
Figur 23: Hjulformet mikrofonmatrise



(a)



(b) Oppsett for hjulformet mikrofonmatrise med 30-mikrofoner



(c) Oppsett for hjulformet mikrofonmatrise med 18-mikrofoner



(d)

Figur 24: Hjulformet mikrofonmatriser

## 3 Teori

Dette kapittelet er delt opp i to deler. Grunnen til dette er at uforutsette utfordringer dukket opp underveis i masteroppgaven. Fordi problemstillingen vi startet med ikke var mulig å redegjøre for på grunn av mangel på både data, mikrofonmatrise og en del komponenter så ble denne nye problemstillingen laget for å få fullført masteroppgaven. Den nye problemstillingen er like mye relevant som den som vi startet med i utgangspunktet. Forskjellen mellom dem to er at den nye delen av oppgaven er ikke avhengig av data fra en mikrofonmatrise eller diverse andre forhold som er blitt diskutert i oppgaven frem til nå.

Den første delen vil være relatert til det som var vår utgangspunkt når vi startet med denne masteroppgaven. Det vil bli belyst hvordan en mikrofonmatrise kan fremstilles matematisk og hvilken type mikrofonmatrise passer best for formålet til Mantena AS.

Den andre delen vil handle om TNS integrert med mikrofonmatrise og en perspektiv kameramodell. Denne kameramodellen vil bli brukt for å kjenne igjen punkter i rommet for å orientere og posisjonere mikrofonmatrisen i forhold til støykilder fra boogien.

### 3.1 Teori - del 1

I dette kapittelet skal det forklares den teoretiske bakgrunnen i anvendelse av mikrofonmatrise både på generelt grunnlag og for feilsøking av boogiene. Med teoretisk bakgrunn menes det, de forskjellige matematiske likningene som er relevante og strukturen som må brukes for å komme til en matematisk algoritme. Dette vil gi oss en løsning for bruk av mikrofonmatrise. Det skal også diskuteres forskjellige aspekter som er relatert til mikrofonmatriser og deres forhold til problemstillingen.

Det er flere ting som må tas med i betraktning ved bruk av mikrofonmatriser i disse omgivelsene:

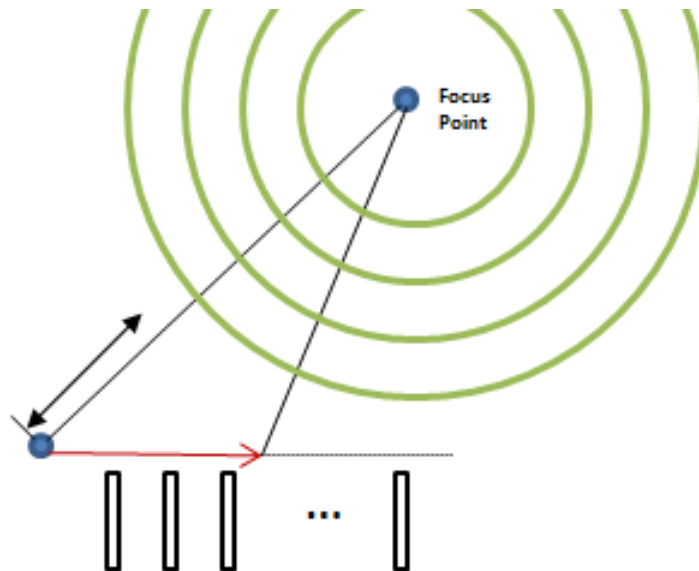
- Mikrofonmatrise vs akselerometer.
- Hvilket type mikrofoner som skal brukes.
- Hvor mange mikrofoner som skal brukes.
- Hvor stor avstand det skal være mellom mikrofonene i arrayet.
- Hvor stor avstand det er mellom støykilden og arrayet.
- Er det et near-field eller far-field lyd kilde.
- Hvor mye støy det genereres fra omgivelsene.
- Hvor lang tid det tar fra fra støykilden oppdages til de forskjellige mikrofonen i arrayet mottar den. Deretter synkroniserer man signalet som gjør at det ikke blir avvik mellom mikrofonene.
- Hvordan fokuset til arrayet kan rettes mot den riktige støykilden og utelukke de andre støykilden ved hjelp av beamforming.

Noen av disse spørsmålene som er stilt, er delvis blitt redegjort for i foregående kapiteler og de resterende spørsmålene vil det bli gitt en klarhet i dette kapitlet. Strukturen til dette kapitlet vil basere seg på den samme rekkefølgen som det er ønskelig å realisere løsningen i praksis.

Dataen hentet ut fra dagens løsning ved bruk av akselerometer vil være den overordnende referansen som skulle ha blitt brukt for å sammenligne med dataen ved bruk av mikrofonmatrise. Denne sammenligningen ville ha vært avgjørende for å se om hvilket av disse to løsningene er mest lønnsomt. Siden dette ikke er tilfellet vil det bare bli forklart oppsettet og karakteristikken til en mikrofonmatrise på et generelt grunnlag.

### 3.1.1 Mikrofonmatriser og beamforming [33]

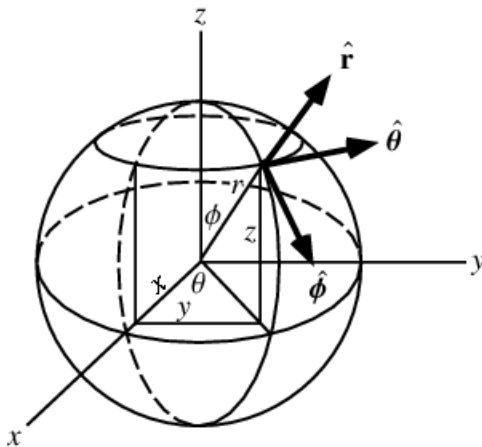
Konseptet med beamforming [32] er som vist i figure 25. Det er å forsterke fokuset på signalet som kommer fra kilden og forminske styrken til de andre signalene i omgivelsen som blir definert som støy. Beamforming metoden kombinert med mikrofonmatrise er noe av det mest avanserte verktøyet som brukes for å forbedre forholdene til lydildens kvalitet og retning. Mikrofonens egenskaper har sine begrensninger i forhold til å vite retningen til hvor kilden befinner seg. Derav også begrensning på forminsking av støy og etterbehandling av gjenklang som lages av omgivelsene rundt lydildens. Som nevnt tidligere i oppgaven grunnen til at vi ønsker å bruke en mikrofonmatrise er for å forbedre kvaliteten på en effektiv måte. Dette kan gjøres ved å fokusere på et mottakende strålmønster i retning av et ønsket signal, herved reduseres interferens og forbedres kvaliteten på den lyden som blir tatt opp.



Figur 25: Beamforming



Den matematiske og geometriske beskrivelsen til en mikrofonmatrise kan forklares med figur 26.



Figur 26: Sfærisk koordinatsystem

La oss anta at det er en lydbølge som beveger seg langs en enhetsvektor  $\rho$  som starter i sentrum av den sfæriske kulen. Vi velger vektoren som en enhetsvektor siden vi ikke er sikker på hvilken retning lyden har sin kilde. Gitt vektoren i sfæriske koordinater:

$$\vec{\rho} = [\cos(\theta_\rho)\sin(\phi_\rho), \sin(\theta_\rho)\sin(\phi_\rho), \cos(\phi_\rho)]$$

For å regne ut forsinkelsen som erfares ved den n-te mikrofonen, blir dens posisjonen definert som  $\vec{p}_i$  i forhold til opprinnelsessted. Hvis mikrofonen er lokalisert ved posisjon  $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ , da vil denne vektoren være definert som:

$$\vec{p}_m = [r_i \cos(\theta_{\vec{p}_m}) \sin(\phi_{\vec{p}_m}), r_i \sin(\theta_{\vec{p}_m}) \sin(\phi_{\vec{p}_m}), r_i \cos(\phi_{\vec{p}_m})]$$

Med lydets hastighet  $c \approx 333 \frac{m}{s}$ , vil tidsforsinkelsen som skjer når lydbølgen treffer den n-te mikrofonen, være gitt av:

$$\tau_i = \frac{(\vec{\rho} | \vec{p}_m)}{c}$$

I form av sampler vil det være gitt som:

$$\Delta_i = \frac{f_s(\vec{\rho} | \vec{p}_m)}{c}$$

og i form av rammen vil det være gitt som:

$$\Delta_i^m = \frac{\lfloor \frac{f_s}{O} \rfloor (\vec{\rho} | \vec{p}_m)}{c}$$

Her er  $O$  rammens "hop size". I frekvensdomene vil tidsforsinkelsen ansees som et faseskift. Denne faseskiftet kan uttrykkes ved:

$$\Theta_i(\omega) = e^{-j\omega\Delta_i}$$

Ofte i romlig (spatial) prosessering er det ønskelig å holde oversikt over bølgetall fordi bølgetall representerer antall bølger i en romlig volum. Derfor ved å konvertere faseskiftet i frekvensdomene til bølgetalldomene, vi får:

$$\Phi_i(\vec{k}) = e^{-j(\vec{k} | \vec{p}_m)}$$

Hvor  $\vec{k}$  er bølgevektoren gitt som:

$$\vec{k} = \frac{\omega}{c} \vec{\rho}$$

Vi kan få tidsforsinkelsen mellom den første mikrofonen i matrisen og den  $n$ -te mikrofonen i matrisen representert ved å skrive  $\phi_i(\omega, \theta, \phi)$ . Hver mikrofon har sin egen tidsforsinkelse. For en gitt frekvens og hendelsesvinkel, vi setter alle disse forsinkelsene i en vektor:

$$\vec{d}(\omega, \theta, \phi) = [1, e^{j(\vec{k} | \vec{p}_1)}, e^{j(\vec{k} | \vec{p}_2)}, \dots, e^{j(\vec{k} | \vec{p}_{N-1})}]^H$$

Hvor  $(\cdot)^H$  representerer Hermitian transponert. Det første elementet i vektoren refereres til mikrofonen med null forsinkelse. Denne mikrofonen blir brukt som tidsreferanse for de andre mikrofonene i matrisen. Vektoren er mest kjent som styringsvektoren og representerer hvordan matrisen oppfører seg i forhold til plan bølger (plane wave) av frekvens  $\omega$  og hendelsen langs retningen  $\vec{\rho}$  i et 3D rom. Ved å regne ut denne vektoren kontinuerlig for hvert frekvens og hver hendelsesvinkelpar vil det gi deg en matrise manifold  $\mathbf{D}$ .

Hvis du vet hvert mikrofonens retningsbane  $\vec{\alpha}(\omega, \theta, \phi)$ , kan dette tilføyes i hver manifold vektor ved å:

$$\vec{d}_A(\vec{\alpha}, \omega, \theta, \phi) = [\alpha_0, \alpha_1 e^{j(\vec{k} | \vec{p}_1)}, \alpha_2 e^{j(\vec{k} | \vec{p}_2)}, \dots, \alpha_N e^{j(\vec{k} | \vec{p}_{N-1})}]^H$$

Derfor gir vektor  $\vec{d}_A$  spesifikk din mikrofonmatrise manifold  $\mathbf{D}_A$ . Denne manifolden forteller deg alt du trenger å vite om hvordan din mikrofonmatrise vil reagere på hvilken som helst plan bølge med

hvilken som helst frekvens. Problemet med denne tilnærmingen er at det er vanskelig å opprettholde denne type matrise manifold. Fordi manifold er en matematisk kontinuerlig objekt og i virkelig verden vil egenskapene til mikrofoner ha en eller annen form for avvik fra teorien.

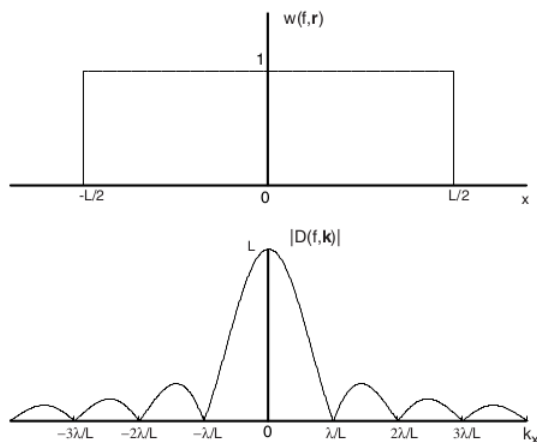
En realisert styringsvektor vil ikke være optimalt for alle frekvenser og vinkler. Det vil være rom for feil i den retning styringen av vektoren skjer i forhold til lydilden. Det gjør heller ikke ting enklere at individuelle mikrofonegenskaper er med på forvrengte strålen til uønsket sidelover og frekvens relatert lydemping. Derfor for å forsikre oss mot at dette skjer definerer vi stråleformet styringsvektor. Den retningen som signalet kommer fra vil da få en større faktor  $G$  for forsterkning. Imens de andre retningene vil dempes. Dette gjøres for å forminske signaler som er forvrengt i den virkelige verden. Den nye stråleformet styringsvektoren vil se slik ut:

$$\mathbf{A}_d(\omega, \theta, \phi) = (\vec{w} | \vec{d}_A) c$$

hvor  $\vec{w}$  er den vektete vektoren, og  $(\cdot | \cdot) c$  representerer standardverdiene for indreprodukt over et komplekst felt. En annen måte å definere stråleretningen på er ved  $|Ad|^2$  som igjen kan uttrykkes på en mer kjent måte:

$$A_d = \sum_{\mathbf{m}=0}^{N-1} w(\omega, m) e^{-j(k|p_m)}$$

Mikrofon indeksen er representert som  $m$ . For en bestemt frekvens er dette den diskrete Fourier transformasjonen av vektvektoren for alle mikrofonene i matrisen. Den enkleste formen for en kontinuerlig lineær matrise med lengde  $L$  med uniform vektning vil være  $w(\omega, m) = \text{rect}(\frac{m}{L})$ . Og responsen av stråleforming vil da bli  $r = L \sin(k_x L)$ . Det er også velt og merke at nullene forekommer med regelmessige brøkdeler for  $r = \frac{m\lambda}{L}$ . Alt dette er vist med figur 27.



Figur 27: Uniform vektning vist for lineær mikrofonmatrise

Hvor  $k_x$  er  $x$  komponenten til bølgevektoren. Når vi har regnet ut retningen for alle mulige frekvenser og innfallsvinkler, kan vi visualisere retningen til vår matrise. Vi bruker fase komponenten til vektvektoren for at mikrofonen skal være rettet mot ønsket retning. Det aller viktigste for vektvektoren er å skaffe retningen til ønsket signal og filtrere bort forstyrrelsene som påvirker signalet. Denne prosessen kalles for “Nullsteering”. La oss nå anta at vi ønsker å styre representasjonen av lydbølgen som inntreffer ved  $\vec{\rho}_i(\Theta_i, \phi_i)$ , vi kan da la vektene være:

$$w_m(\omega) = |w_m|e^{j(k(\theta-\theta_i, \phi-\phi_i)|p_m)}$$

Som igjen kan gi retningsmønster uttrykt med:

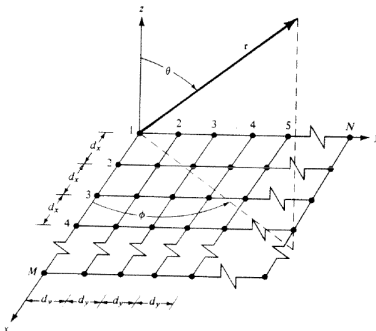
$$r(\omega, \theta, \phi) = \sum_{\mathbf{m}=0}^{M-1} |w_m|e^{-j(k(\theta-\theta_i, \phi-\phi_i)|p_m)}$$

Hovedsakelig er det to trinn for å velge vektvektoren. Det første er å lage en fasevektor basert på estimering av hvor lyden kommer fra. Det andre trinnet er at vi må finne det optimale settet av størrelser. Siden rom og tid har en tilknytning til hverandre, er også disse to prosessene tilknyttet hverandre på en ubøyelig måte. Med andre ord vil feilen for hvor kilden befinner seg føre til beregning av suboptimale vektorer i forhold til virkeligheten til hvor lydkilden befinner seg.

### 3.1.2 Planar mikrofonmatrise [32] [33]

Dette er en mikrofonmatrise som er formet som et plan som vist med figur 28. Siden boogien kjøres inn i den Dynamiske Testbenken og står i ro, tenker vi at denne løsningen av en mikrofonmatrise vil være til mest nytte. Ved å ha mikrofonmatrisen rettet mot boogiene, vil ikke støyen bak mikrofonmatrisen ha noe særlig påvirkning for målingene.

Størrelsen på mikrofonmatrisen vil bero på antall mikrofoner i matrisen og den vil utformes som et firkant med vilkårlig lengde og bredde. Det betyr at det kan være like mange mikrofoner begge veier eller det kan være flere mikrofoner den ene retningen i forhold til den andre.



Figur 28: Mikrofonmatrise formet som et rektangulært plan

Fra figuren kan vi forklare sammenheng mellom antall mikrofoner  $M$  i retning x-aksen og antall mikrofoner  $N$  i retning y-aksen ved uttrykt med likning:

$$r = \sum_{\mathbf{m}=0}^{N-1} \left( \sum_{\mathbf{m}=0}^{M-1} w_{x_m} e^{\frac{\omega}{c} x_m \cos(\theta) \sin(\phi) + \beta_{x_m}} \right) \cdot w_{y_m} e^{\frac{\omega}{c} y_m \cos(\theta) \sin(\phi) + \beta_{y_m}}$$

Beskrevet på en annen måte vil det si at den totale responsen til mikrofonmatrisen kan vi få ved å multiplisere den individuelle lineærmatrisen i  $x$  retningen og den individuelle lineærmatrisen i  $y$  retningen sammen.

Vi kan forenkle den totale likningen med to nye variabler  $S_{x_m}$  og  $S_{y_m}$

der:

$$x = k_x d_x \cos(\theta) \sin(\phi) + \beta_x$$

og

$$y = k_y d_y \cos(\theta) \sin(\phi) + \beta_y$$

Vi kan da ta den totale utledningen for x- og y-retningen ved å uttrykke vår likning på denne måten (dette gjøres separat for hver retning og deretter multipliseres de med hverandre):

$$\begin{aligned} r &= \sum_{\mathbf{m}=0}^{M-1} w(\omega, x_m) e^{jm\psi} \\ r e^{j\psi} &= \sum_{\mathbf{m}=0}^{M-1} e^{jm\psi} \\ r(e^{j\psi} - 1) &= (e^{jM\psi} - 1) \\ r &= \frac{e^{jM\psi} - 1}{e^{j\psi} - 1} \\ r &= e^{j\frac{M-1}{2}\psi} \frac{e^{j\frac{M}{2}\psi} - e^{-j\frac{M}{2}\psi}}{e^{j\frac{1}{2}\psi} - e^{-j\frac{1}{2}\psi}} \\ r &= e^{j\frac{M-1}{2}\psi} \frac{\sin(\frac{M}{2}\psi)}{\sin(\frac{1}{2}\psi)} \\ r &= \frac{\sin(\frac{M}{2}\psi)}{\sin(\frac{1}{2}\psi)} \\ r &\approx \frac{\sin(\frac{M}{2}\psi)}{\frac{\psi}{2}} \\ r &\approx \text{sinc}(\psi) \end{aligned}$$

Den samme fremgangsmåten vil være for y-aksen. Det vil gi oss to komponenter som  $r = \text{sinc}(\psi_x)$  og  $r = \text{sinc}(\psi_y)$ . Deretter som nevnt tidligere kan vi bruke de to forenklete variablene som vist med denne likningen:

$$r = S_{x_m} \cdot S_{y_m}$$

$$r = \frac{\sin(\frac{M}{2}\psi_x)}{\sin(\frac{1}{2}\psi_x)} \cdot \frac{\sin(\frac{N}{2}\psi_y)}{\sin(\frac{1}{2}\psi_y)}$$

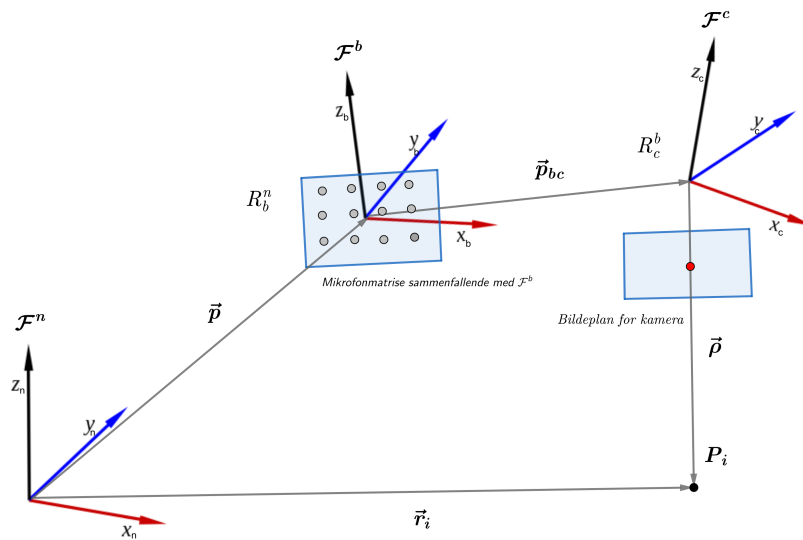
$$r \approx \frac{\sin(\frac{M}{2}\psi_x)}{\frac{\psi_x}{2}} \cdot \frac{\sin(\frac{N}{2}\psi_y)}{\frac{\psi_y}{2}}$$

$$r \approx \text{sinc}(\psi_x) \cdot \text{sinc}(\psi_y)$$

### 3.2 Teori - del 2

I denne delen av rapporten skal vi se på et treghetsnavigasjonssystem (TNS) kombinert med en mikrofonmatrise og et kamera. Hensikten med det er å bruke en mikrofonmatrise ved hjelp av kamera som kan beveges i rommet for å fokusere på støykilden. I dette tilfellet vil det si rundt en boogie. Vårt mål er å implementere et navigasjonssystem som kan brukes til å bevege mikrofonmatrisen rundt boogien. Det vil si at man starter på et punkt og beveger seg i sirkel rundt dette massesenteret. Modellen vil ha den enkleste form. Det vil si at det brukes parametriserte likninger for en sirkel, hvor vi i første omgang vil se posisjon, fart og akselerasjon over tid ved bevegelse rundt dette massesenteret. Etter hvert vil de bli tilført litt støy i form av at banen som det bevegtes rundt ikke blir rett bane, men med litt variasjon. For å få til det så har det blitt laget en banegenerator.

Den andre delen vil omhandle et perspektiv kameramodell, et TNS og en UKF (Unscented Kalmanfilter). Ved hjelp av noen gitte og bestemte verdier ut i rommet skal vi bestemme stillingen og posisjonen til mikrofonmatrisen. For å få til det så må kamera og TNS samhandle med hjelp fra en rotasjonsmatrise og en translasjon vektor. Neste steg vil være å hente data og sette det inn i en estimeringsalgoritme for å finne ut om verdiene for posisjon og stillingen vil samsvare med ønskede verdier. Det gjør vi med en UKF.

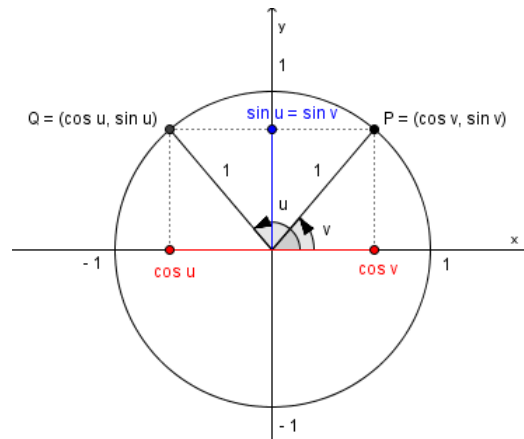


Figur 29: Geometrisk fremstilling med alle rammene i Mantena AS sitt dynamisk testbenk

Vi starter med teorien og matematikken for en banegenerator. Deretter en forklaring på hva et perspektiv kameramodell er og de matematiske likningene som blir brukt. Også tar vi for oss TNS og tilslutt en UKF og dens likninger.

### 3.2.1 Banegenerator [17] [23]

Det lages en banegenerator for å simulere bevegelsesmønsteret til et objekt. I vårt tilfelle er banegeneratoren med på å definere bevegelsen av mikrofonmatrisen og kameraet sammen med en TNS. Det er blitt brukt en bane i sirkel (figur 30) også kalt en sirkelbane, rundt boogien. Massesenteret til boogien er blitt brukt som senteret til sirkelen rundt boogien. Tanken bak en banegenerator er å registrere posisjon, hastighet og akselerasjon med hensyn til tid.



Figur 30: Trigonometrisk fremstilling av enhetssirkel for å vise bevegelse langs sirkelen [18]

Likninger for banegenerator hvor:

$r$  = radius til sirkelen

$t$  = tiden det er ønskelig for objektets bevegelse

$T_{\omega}$  = er tiden det tar for et omløp av hele sirkelen,  $360^{\circ}$ .

$\omega$  = vinkelhastighet

$\theta$  = vinkel theta

$\vec{p}$  = posisjonsvektor for å definere objektets posisjon langs sirkelen.

$$\vec{p} = \sum_{i=1}^3 \vec{n}_i \underline{p}^i(t) = r \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix}; \text{ kan også skrives } = r \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \quad (1)$$



$\vec{v}$  = hastighetsvektor for å gjengi farten til objektet rundt sirkelen.

$$\vec{v} = \sum_{i=1}^3 \vec{n}_i \underline{v}^n(t) = r\omega \begin{bmatrix} -\sin(\omega t) \\ \cos(\omega t) \end{bmatrix}; \text{ kan også skrives } = r \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \end{bmatrix} \quad (2)$$

$\vec{a}$  = akselerasjonsvektor for å gjengi akselerasjonen når objektet begynner å bevege seg.

$$\vec{a} = \sum_{i=1}^3 \vec{n}_i \underline{a}^n(t) = r\omega^2 \begin{bmatrix} -\cos(\omega t) \\ -\sin(\omega t) \end{bmatrix}; \text{ kan også skrives } = r \begin{bmatrix} -\cos(\theta) \\ -\sin(\theta) \end{bmatrix} \quad (3)$$

Oppsummert blir bevegelseslikningene for sirkelbane i rommet med  $z = \dot{z} = \ddot{z} = 0$ :

$$\underline{p}^n(t) = \begin{bmatrix} r\cos(\omega t) \\ r\sin(\omega t) \\ 1 \end{bmatrix}; \underline{v}^n(t) = \begin{bmatrix} -r\omega\sin(\omega t) \\ r\omega\cos(\omega t) \\ 0 \end{bmatrix}; \underline{a}^n(t) = \begin{bmatrix} -r\omega^2\cos(\omega t) \\ -r\omega^2\sin(\omega t) \\ 0 \end{bmatrix}; \quad (4)$$

Det er viktig å ta med en av de trigonometriske identitetene  $\rightarrow \cos^2\theta + \sin^2(\theta) = 1$  for å forenkle likningene 5 til 7.

$$v = p\omega \rightarrow \omega = \frac{v}{p} \quad (5)$$

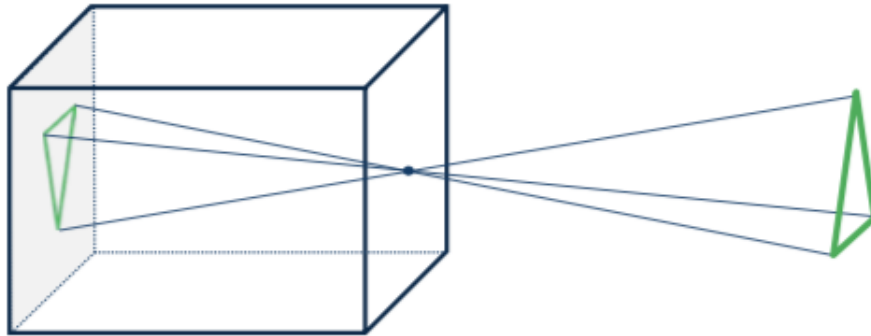
$$a = p\omega^2 \rightarrow a = p \cdot \left(\frac{v}{p}\right)^2 = \frac{v^2}{p} \quad (6)$$

$$\omega = \frac{2\pi}{T_\omega} = \frac{\underline{a}^n}{\underline{v}^n} = \frac{\underline{v}^n}{\underline{p}^n} \quad (7)$$

$$\theta(t) = \omega t \quad (8)$$

### 3.2.2 Perspektiv kameramodell [26]

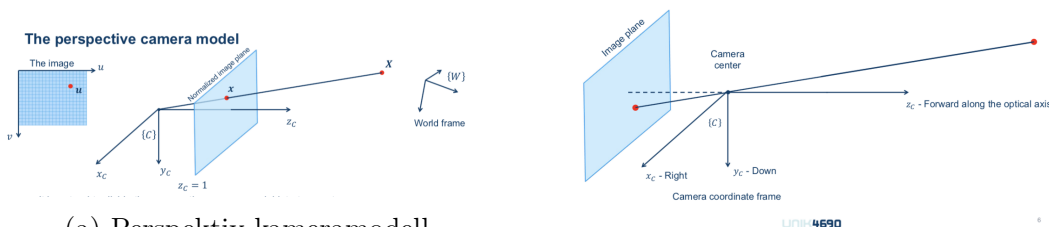
Perspektiv kameramodell (Figur 31 og 32) eller pin-hole kameramodell, er en enkel bildebehandlings enhet. Perspektiv kameramodell er et matematisk modell som beskriver sammenheng mellom observerte punkter ut i verden og piksler som blir projisert i bildeplanet til kamera. For å beskrive transformasjon fra 3D punkter i verden til 2D punkter i bildet trenger vi å representere kamera og verden som to rammer. I vårt tilfelle vil de si at vi ser på punktet fra treghetsrammen representert i kamerarammen.



Figur 31: Perspektiv eller pin-hole kameramodell

For å anvende matematikken i et perspektiv kameramodell så må kameramodellen deles i to deler. Disse to deler blir kalt *extrinsic* og *intrinsic*. Begge disse to delene har en sammenheng som gjør at geometrien til kameramodellen blir enklere å representere fra et punkt i 3D-rom til 2D-bildeplan. Begge disse to delene er også representert ved hjelp av et homogen matrise.

- Extrinsic:  $\underline{\mathbf{X}}^W \mapsto \underline{\mathbf{x}}^c$        $3D \mapsto 2D$
- Intrinsic:  $\underline{\mathbf{x}}^c \mapsto \underline{\mathbf{u}}$        $2D \mapsto 2D$



(a) Perspektiv kameramodell representert med kamera- og verdensrammen

(b) Perspektiv kameramodell representert med kamerarammen

Figur 32: Perspektiv/pin-hole kameramodell fremstilt på forskjellig måter

Den enkleste matematiske formen av et perspektiv kameramodell er vist i likning 9.

$$\tilde{\underline{u}} = K \begin{bmatrix} R & \underline{t} \end{bmatrix} \tilde{\underline{X}}^W \quad (9)$$

Den mer detaljerte matematiske formen gir en utfyllende beskrivelse av parametere som er med å definere en perspektiv kameramodell som vist i likning 10.

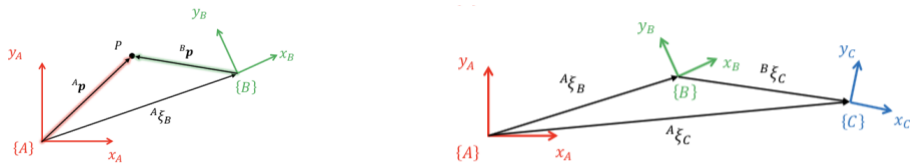
$$\tilde{\underline{u}} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & \underline{t}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \tilde{\underline{X}}^W \quad (10)$$

De forskjellige matrisene og parametrene i likning 9 og 10 blir forklart i detaljer i avsnittene som kommer nedenfor, som f.eks Pose i 2D og 3D, de extrinsic og intrinsic delene til et perspektiv kameramodell.

### 3.2.2.1 Pose i 2D og 3D

For å forklare de likningene ovenfor som representerer et perspektiv kameramodell, er det viktig å forklare hva som menes med *pose*. Hvis du setter posisjon og orientering sammen så vil du få en pose. Grunnen til at det trengs å vite om pose er for å få innsikt i geometriske problemer relatert til et bilde. For å studere og uttrykke disse problemene så er kjennskapen til posisjon og orientering til et objekt veldig viktig. Ved å representere alle kjente objekter med deres respektive koordinatrammer, gjør det mulig å representere posen relativt til et objekt i forhold til posen til et annet objekt.

#### 3.2.2.1.1 Pose i 2D



(a) 2D pose med 2 rammer

(b) 2D pose med 3 rammer

Figur 33: To forskjellige representasjoner av 2D pose med både rammer og punkt i planet

- Generelle egenskaper for pose i 2D:

Vi bruker notasjonen  $\xi_B^A$  for å fortelle om posen av ramme {B} relativt til ramme {A} som vist på figur 33a. Tillegg til det så ser vi på et punkt  $P$  med respekt til begge rammene. Det er viktig å forstå at  $\xi_B^A$  beskriver translasjon og rotasjon som kreves for at ramme {A} skal være sammenfallende med ramme {B}. På denne måten vil det være mulig å vise til punktet  $P$ , ved hjelp av vektor  $\underline{p}^A$  og vektor  $\underline{p}^B$ . Merk at begge disse to vektorene er forskjellige fra hverandre. Den matematiske representasjonen av figur 33a er vist med likning 11.

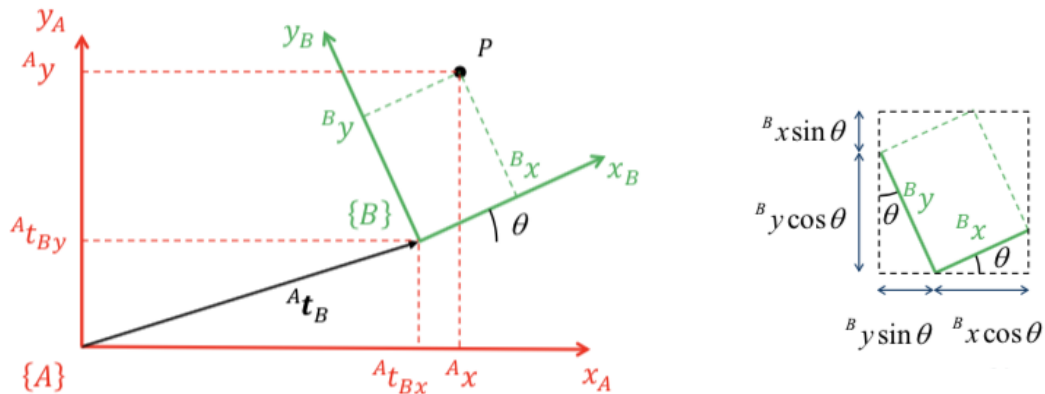
$$\underline{p}^A = \xi_B^A \cdot \underline{p}^B \quad (11)$$

Figur 33b viser sammensetningen mellom tre rammer. Den matematiske notasjonen for dette er vist med likning 12 der ved å multiplisere den ene posen med den andre vil gi produktet av dem to:

$$\xi_C^A = \xi_B^A \oplus \xi_C^B \quad (12)$$

- Utforsking av pose i 2D:

Før vi setter oss inn i pose for 3D, som vil være det mest naturlige for å lage et navigasjonssystem ved hjelp av en mikrofonmatrise og en kamera, er det viktig å forstå hvordan trigonometrien beskriver forholdet mellom rammene i et 2D-rom. Trigonometrien har en viktig rolle når vi skal ha en rotasjon rundt forskjellige akser. I tillegg til det kan det også være en translasjon mellom disse rammene. Rotasjonsmatrisen sørger for at rammene sammenfaller med respekt for hverandre når du ser et punkt ut i rommet eller et plan. Translasjonsvektoren definerer avstanden mellom de forskjellige rammene og m.m.



Figur 34: Trigonometrisk beskrivelse av 2D-pose med to akser

De matematiske likningene fra likning 9 t.o.m likning 11 forteller hvordan likning 12 kan lages med en numerisk representasjon. Tidligere i dette kapittelet var vi inne på hvordan  $\xi_B^A$  kan

sees som en translasjon og rotasjon for å få ramme {A} og ramme {B} til å sammenfalle. Dette kan gjøres med en translasjon  $\underline{t}_B^A$  og en rotasjon (med vinkel  $\theta$ ) for ramme {A}, for at disse to rammene skal harmonisere med hverandre (se Figur 34).

Utvider notasjonen på de forskjellige paramaterene fra likning 11:

$$\underline{t}_B^A = \begin{bmatrix} t_{Bx}^A & t_{By}^A \end{bmatrix}^T \quad (13)$$

$$\underline{p}^A = \begin{bmatrix} x^A & y^A \end{bmatrix}^T \quad (14)$$

$$\underline{p}^B = \begin{bmatrix} x^B & y^B \end{bmatrix}^T \quad (15)$$

Figur 34 viser hvordan du vil få likning 16 som kan skrives i matrisform som vist i likning 17.

$$(16)$$

$$x^A = t_{Bx}^A + x^B \cos \theta - y^B \sin \theta \quad (16a)$$

$$y^A = t_{By}^A + x^B \sin \theta + y^B \cos \theta \quad (16b)$$

$$\begin{bmatrix} x^A \\ y^A \end{bmatrix} = \begin{bmatrix} t_{Bx}^A \\ t_{By}^A \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x^B \\ y^B \end{bmatrix} \quad (17)$$

Den mer kompakte versjonen av likning 17 er:

$$\underline{p}^A = \underline{t}_B^A + R_B^A \underline{p}^B \quad (18)$$

Det er fullt mulig å beskrive et punkt i planet ved hjelp av å tilføre en tredje koordinat i likning 18 til tross for at den har bare to frihetsgrader. Da vil du få likning 19. Denne utvidelsen av matrisen sørger for at du går fra en ikke-lineær transformasjon  $\underline{p}^B \mapsto \underline{p}^A$  til en lineær transformasjon  $\tilde{\underline{p}}^B \mapsto \tilde{\underline{p}}^A$ .

$$\begin{bmatrix} x^A \\ y^A \\ 1 \end{bmatrix} = \begin{bmatrix} R_B^A & t_B^A \\ 0_{1 \times 2} & 1 \end{bmatrix} \begin{bmatrix} x^B \\ y^B \\ 1 \end{bmatrix} \quad (19)$$

Og den mer kompakte likningen for likning 19, det vil si den lineære transformasjonen skrives:

$$\tilde{\underline{p}}^A = T_B^A \tilde{\underline{p}}^B \quad (20)$$

Nå som vi har både likning for ikke-lineær transformasjon (18) og lineær transformasjon (20) ser vi at ved å gjøre noen få enkle justering får vi punkter presentert fra et *Euklidske* plan til punkter presentert i det *projektive* plan. Disse punktene blir beskrevet med kartesiske koordinater i det *Euklidske* planet og med homogene koordinater i det *projektive* planet. Det er vell å merke at *Euklidsk* geometri har en mye mer komplisert algebra utregning i motsetning til *projektiv* geometri. Der utregningen blir mye mer simple.

Euklids geometri og egenskaper:

$$(21)$$

$$\xi_B^A \mapsto (R_B^A, \underline{t}_B^A) \quad (21a)$$

$$\underline{p}^A = \xi_B^A \cdot \underline{p}^B \mapsto \underline{p}^A = R_B^A \underline{p}^B + \underline{t}_B^A \quad (21b)$$

$$\xi_C^A = \xi_B^A \oplus \xi_C^B \mapsto (R_C^A, \underline{t}_C^A) = (R_B^A R_C^B, R_B^A \underline{t}_C^B + \underline{t}_B^A) \quad (21c)$$

$$\ominus \xi_B^A \mapsto (R_C^{A^T}, -R_C^A \underline{t}_C^A) \quad (21d)$$

Projektiv geometri og egenskaper:

$$(22)$$

$$\xi_B^A \mapsto T_B^A = \begin{bmatrix} R_B^A & \underline{t}_B^A \\ 0 & 1 \end{bmatrix} \quad (22a)$$

$$\underline{p}^A = \xi_B^A \cdot \underline{p}^B \mapsto \tilde{\underline{p}}^A = T_B^A \tilde{\underline{p}}^B \quad (22b)$$

$$\xi_C^A = \xi_B^A \oplus \xi_C^B \mapsto T_C^A = T_B^A T_C^B \quad (22c)$$

$$\ominus \xi_B^A \mapsto T_B^{A^{-1}} \quad (22d)$$

Hvis man ser på geometrien og egenskapene til de to forskjellige planene, er det veldig tydelig at den matematikken for et homogen koordinatsystem er mindre komplisert å utregne enn et kartesisk koordinatsystem.

Likning 22a kan også beskrives på denne måten, som vist nedenfor:

$$\xi_B^A \mapsto T_B^A = \begin{bmatrix} R_B^A & \underline{t}_B^A \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & t_{Bx}^A \\ \sin \theta & \cos \theta & t_{By}^A \\ 0 & 0 & 1 \end{bmatrix} \quad (23)$$

### 3.2.2.1.2 Pose i 3D

- Hvordan bør ramme {B} rotere for at den skal være sammenfallende med ramme {A}

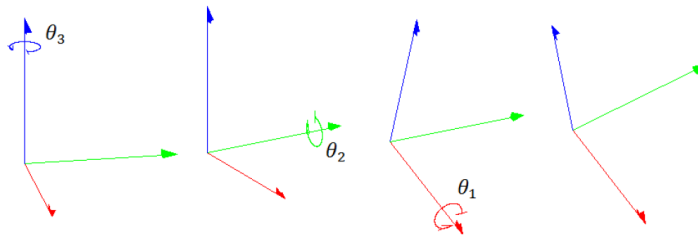
Pose i 2D gir en grunnleggende og enkel forståelse for hvordan vi kan fremstille et punkt når vi ser og representerer den fra forskjellige rammar. 3D pose er litt mer avansert. Det kan sies at forskjellen mellom disse to posene er at rotasjonen i 2D er mer intuitivt enn det rotasjon er i 3D. I 2D finnes det kun en måte å fremstille rotasjon på, men i 3D finnes det flere. I denne rapporten vil vi bruke rotasjon med *Eulervinkler*, men det er også viktig å forstå rotasjon med de andre fremgangsmåtene. Nedenfor er disse fremgangsmåtene listet etter avsnittet med *Eulervinkler*.

- *Eulervinkler*  $(\theta_x, \theta_y, \theta_z)$

Eulervinkler  $(\theta_x, \theta_y, \theta_z)$  beskriver rotasjon rundt 3 akser imens en av aksene holdes fast. Det er 12 måter å sammensette disse vinklene på som gir hver sitt unike dreiesekvens.

*xyz xzy xyx xzx*  
*yxz yzx yxy yzy*  
*zxy zyx zxz zyz*

Det er to dreiesekvenser som er mest kjent,  $3-2-1 \rightarrow zxy$  (se Figur 35) og den klassiske  $3-1-3 \rightarrow xxz$ . Dreiesekvensen 3-2-1 er ofte brukt ved simulering av fly og båt.



Figur 35: Eulervinkel dreiesekvens *zxy* også kalt 3-2-1

For å anvende dreiesekvensene vist på foregående side må det tas i bruk elementære RKM(retningskosinmatrise):

$$R_z(\theta_3) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

$$R_y(\theta_2) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (25)$$

$$R_x(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (26)$$

Et eksempel på hvordan en dreiesekvens ser ut hvis du bruker 3-2-1 Eulervinkler med RKM  $R_{zyx} = R_z(\theta_3)R_y(\theta_2)R_x(\theta_1) = R_B^A$  (likning 27).

$$R_{zyx} = \begin{bmatrix} c_{\theta_3}c_{\theta_2} & c_{\theta_3}s_{\theta_2}s_{\theta_1} - s_{\theta_3}c_{\theta_1} & c_{\theta_3}s_{\theta_2}c_{\theta_1} + s_{\theta_3}s_{\theta_1} \\ s_{\theta_3}c_{\theta_2} & s_{\theta_3}s_{\theta_2}s_{\theta_1} + c_{\theta_3}c_{\theta_1} & s_{\theta_3}s_{\theta_2}c_{\theta_1} - c_{\theta_3}s_{\theta_1} \\ -s_{\theta_2} & c_{\theta_2}s_{\theta_1} & c_{\theta_2}c_{\theta_1} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{231} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (27)$$

Det inverse problemet for 3-2-1 Eulervinkler:

$$\theta_1 = \arctan 2 \left( \frac{r_{32}}{c_{\theta_2}}, \frac{r_{33}}{c_{\theta_2}} \right) \quad -\pi < \theta_x < \pi \quad (28a)$$

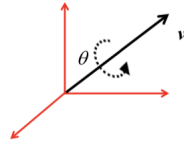
$$\theta_2 = \arctan 2 \left( -r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right) \quad -\frac{\pi}{2} < \theta_y < \frac{\pi}{2} \quad (28b)$$

$$\theta_3 = \arctan 2 \left( \frac{r_{21}}{c_{\theta_2}}, \frac{r_{11}}{c_{\theta_2}} \right) \quad -\pi < \theta_z < \pi \quad (28c)$$



- **Vinkel-akse**

En vinkel-akse par  $(\theta, \mathbf{v})$  beskriver rotasjon i forhold til en vinkel og en akse. Denne samme tilnærmingen brukes for å definere Eulers rotasjon teorem. Eulers rotasjon teorem sier at rotasjon i en 3 dimensjonal plan er lik en enkel rotasjon  $\theta$  rundt en akse  $\mathbf{v}$ .



Figur 36: Vinkel-akse par  $(\theta, \mathbf{v})$

Rodrigues rotasjon formel:

$$R = I_3 + \sin \theta [v]_x + (1 - \cos \theta)(vv^T - I_3)$$

Hvor  $[v]_x$  er matrise representasjon av kryss-produktet:

$$\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad (29)$$

- *Ortonormal rotasjon*

Fremgangsmåten med den ortonormale rotasjonens metoden, er den metoden som har en enkel beregningsmåte og gode algebraiske egenskaper.

Ortonormal rotasjonsmatrise  $R \in SO(3)$

$$R_B^A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (30)$$

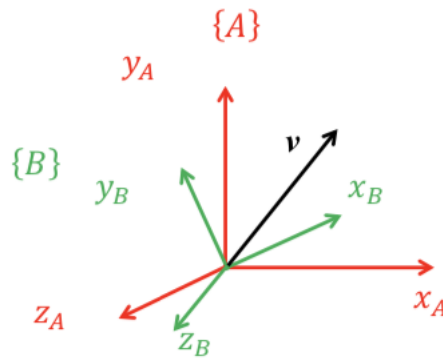
Egenskapene til en ortonormal rotasjon:

$$\det(R_B^A) = 1 \quad (31)$$

$$R_B^{A^{-1}} = R_B^{AT} \quad (32)$$

$$R_C^A = R_B^A R_C^B \quad (33)$$

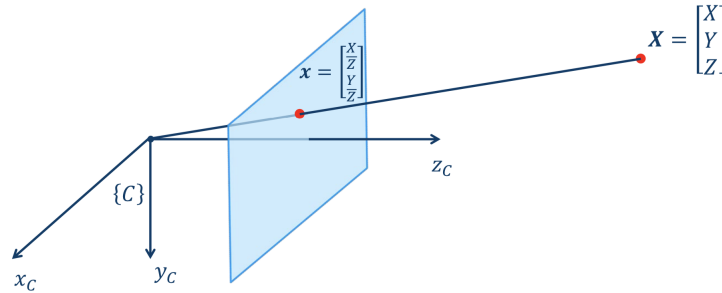
$$\underline{v}^A = R_B^A \underline{v}^B \quad (34)$$



Figur 37: Ortonormal rotasjon for ramme A og ramme B

### 3.2.2.2 Perspektiv kameramodell - extrinsic

Perspektiv kameramodell er delt opp i 2 deler. Den ene delen tar for seg matematikken for et punkt i verden som er representert i bildeplanet til et kamera. Denne delen kalles for den “extrinsic” delen og er vist med figur 38. Det er denne delen som er essensiell og blitt brukt i denne masteroppgaven.



Figur 38: Extrinsic delen til en perspektiv kameramodell

Extrinsic delen er laget av Euklidske transformasjonen av punktet fra verden til kamera, og perspektiv projeksjonen fra 3D til 2D. Denne kombinasjonen vises med likning 35. Hvor matrisen til høyre brukes for den perspektive projeksjonen og den til venstre brukes for transformasjonen mellom koordinatene.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & t_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (35)$$

For å presentere posen til kamera i forhold til verden brukes det noe som kalles for en homogen transformasjon av punkter fra kamera til verden ( se likning 37).

$$\xi_C^W = \begin{bmatrix} R_C^W & t_C^W \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (36)$$

$$\tilde{\mathbf{X}}^W = \xi_C^W \tilde{\mathbf{X}}^C \quad (37)$$

Dermed kan vi uttrykke den Euklidske transformasjonen fra verdensrammen til kamerarammen med posen til kamera i forhold til verdensrammen.

$$\begin{bmatrix} R_{3x3} & \mathbf{t}_{3x1} \\ 0_{1x3} & 1 \end{bmatrix} = \xi_C^{W-1} = \begin{bmatrix} R_C^W & \mathbf{t}_C^W \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R_C^{WT} & -R_C^{WT} \mathbf{t}_C^W \\ \mathbf{0}_{1x3} & 1 \end{bmatrix} \quad (38)$$

$$\tilde{\mathbf{X}}^C = \xi_C^{W-1} \tilde{\mathbf{X}}^W \quad (39)$$

Skrevet likning 39 på en annen måte vil være som vist med likning 40.

$$\tilde{\mathbf{X}}^C = \xi_W^C \tilde{\mathbf{X}}^W \quad (40)$$

Den perspektive projeksjon fra 3D til 2D kan representeres med en homogen matrise (se likning 41) og i koordinater som vist med figur 38 (se også likning 42).

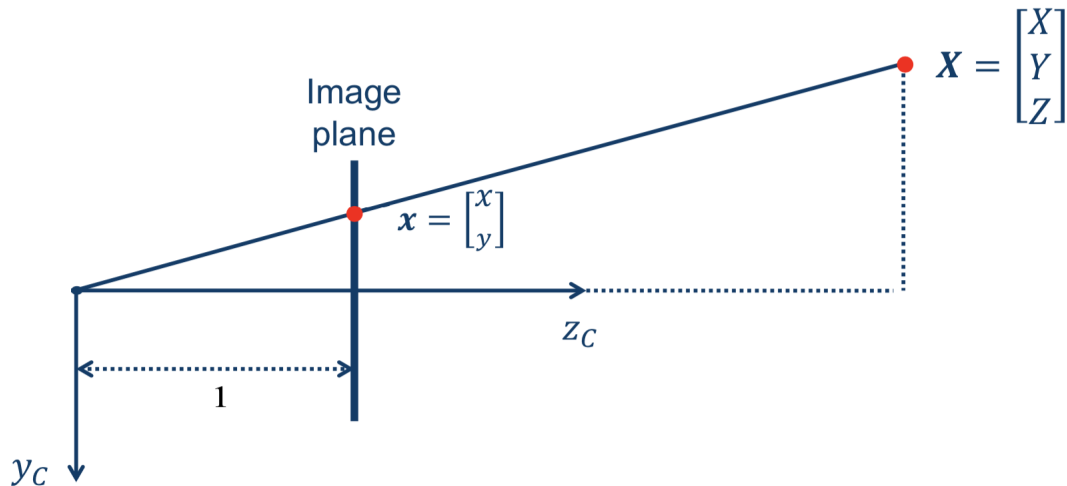
$$\tilde{\mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \tilde{\mathbf{X}} \quad (41)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = d \begin{bmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \\ 1 \end{bmatrix} \quad (42)$$

For å vise at dette er eksakt hva vi ønsker fra et perspektiv projeksjon, kan vi se på y- og z-koordinater. På figur 39 så vises det to helt like trekanter, men med forskjellige skalar. Det vil da

gi oss forholdet mellom dem som vist med likning 43.

$$\frac{y}{Y} = \frac{1}{Z} \Leftrightarrow y = \frac{Y}{Z} \quad (43)$$



Figur 39: Perspektiv projeksjon fra 3D til 2D for et perspektiv kameramodell

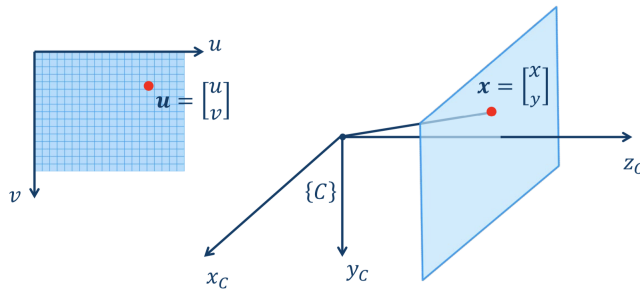
Ved å kombinere den perspektive projeksjonen og den Euklidske koordinat transformasjonen får vi en kompakt og enkel representasjon av den extrinsic delen til et perspektiv kameramodell. Det er denne representasjon som vist med likning 44 vi kommer til å bruke i denne oppgaven. Dette vil gi oss koordinatene i rommet representert i bildeplanet til kamera. Som igjen vil være med på å definere navigasjonen som skal gjøres av et TNS.

$$\begin{bmatrix} R & \underline{\mathbf{t}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \underline{\mathbf{t}} \\ \mathbf{0} & 1 \end{bmatrix} \quad (44)$$

$$\tilde{\mathbf{x}}^C = \begin{bmatrix} R & \underline{\mathbf{t}} \end{bmatrix} \tilde{\mathbf{X}}^W \quad (45)$$

### 3.2.2.3 Perspektiv kameramodell - intrinsic

Den andre delen av en perspektiv kameramodell er den intrinsic delen. Denne delen av perspektiv kameramodell er ikke noe som er direkte relevant til masteroppgaven, men fortsatt greit å ta med for å forstå hele prosessen og virkemåten til et perspektiv kameramodell. Intrinsic delen representerer den indre delen til kameraet. Denne delen forklarer forholdet mellom punkter i det normalisert bildeplanet med kartesiske koordinater til punkter i piksler (se figur 40).



Figur 40: Intrinsic delen til et perspektiv kameramodell

Transformasjonen som er representert med homogen matrise er vanligvis kalt kamera-kalibreringsmatrise  $K$ .

$$\tilde{\mathbf{u}} = K\tilde{\mathbf{x}} \quad (46)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (47)$$

Og matrisen  $K$  kan representeres som vist med likning 48:

$$K = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \quad (48)$$

Definisjon/forklaring på de 5 forskjellige parametrene til kamera-kalibreringsmatrise K:

- $f_u$  og  $f_v$ :

Fokal lengden er avstanden mellom kameraets origo og bildeplanet. Parametrene  $f_u$  og  $f_v$  er den skalerte versjonen av  $f$  som reflekterer at tettheten av detektor elementer kan være forskjellige i u- og v -retning til bildeplanet. Hvis vi angir detektortetthet i u- og v -retning som  $\rho_u$  og  $\rho_v$ , da kan vi uttrykke dette med likning 49.

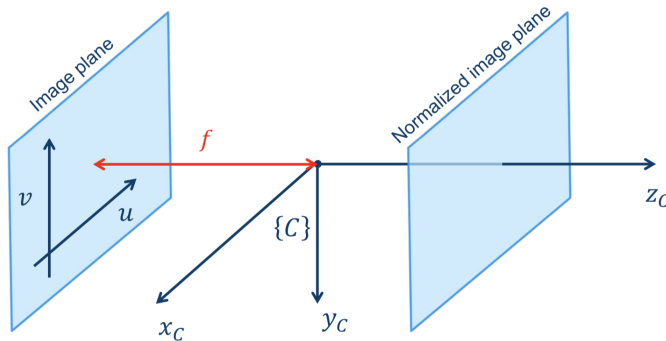
$$\begin{aligned} f_u = \rho_u \cdot f & \Rightarrow \frac{f_u}{\rho_u} = \frac{f_v}{\rho_v} \Leftrightarrow f_v = \frac{\rho_v}{\rho_u} f_u \\ f_v = \rho_v \cdot f & \end{aligned} \quad (49)$$

- $c_u$  og  $c_v$

Det optiske senteret ( $c_u$ ,  $c_v$ ) er hvor den optiske akse krysser bildeplanet. Det er vanlig å definere disse to parametre ca. fra bildets midtpunkt. Men de virkelige verdiene avhenger av hvordan sensormatrisen er plassert med den optiske akse.

- $s$

Denne parameteren representerer skjevhet som oppstår når detektormatrisen har en ikke-ortogonal struktur eller når matrisen ikke er ortogonal til den optiske akse. For de moderne kameraene kan denne parameteren ignoreres og settes til  $s = 0$ .

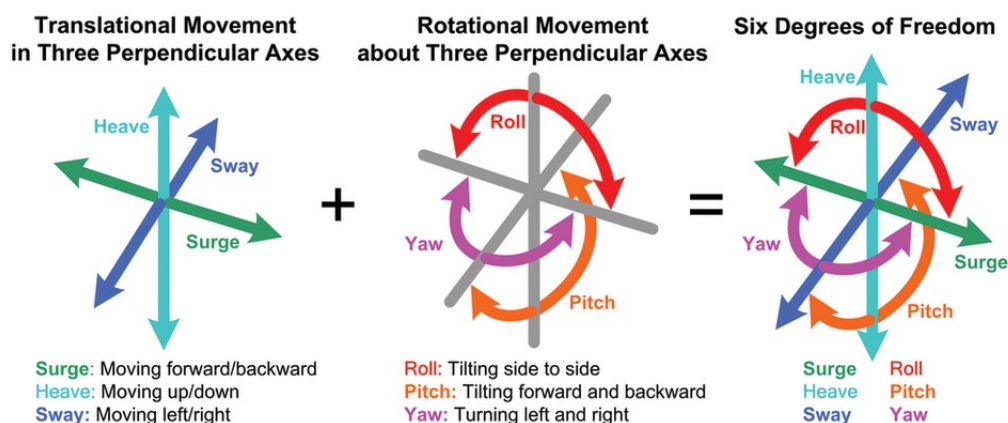


Figur 41: Intrinsice delen til et perspektiv kameramodell med fokal lengde  $f$ .

### 3.2.3 TNS - Treghetsnavigasjonssystem [16] [22]

TNS er et system som benyttes når det er ønskelig å vite posisjon og stilling til et objekt til enhver tid. Det er et system som blir brukt i militære fartøy og våpen. Det vil si militærfly, ubåter, raketter og mer. Du vil også finne TNS i vanlig fly, båt og romfartøyer. Grunnen til det er at alle disse nevnte fartøyene og redskapene er kritiske i forhold til deres orientering i sanntid. Siden det meste som blir laget idag har en eller annen form for elektronikk, så vil de til enhver tid være sikkerhetsutsatt. I militær sammenheng vil det være veldig viktig å navigere gjennom forskjellige terreng på land, under vann og i luftrommet uten at fienden skal få mulighet til å ha innflytelse eller kjennskap til bevegelsene. I de sivile fartøyene vil objektet selv, trafikken og menneskeliv være det kritiske. Hvis elektronikk skulle slutte å virke og kommunikasjonen mellom f.eks flytårnet og flyet opphører, kan det få fatale konsekvenser for både flyet og passasjerene. Derfor finnes det alltid en eller flere TNS i denne type fremkomstmiddel. Det vil sørge for å holde flyet på et stødig kurs og holde oversikt over bevegelsen og posisjonen. Det er viktig å fremheve at med tiden har også produksjonen av navigasjonssystemer blitt mindre kostbar og mer robuste. Måten industrien har redusert kostnaden på er ved å bruke MEMS teknologi. Det har ført til TNS kan benyttes i mange andre sammenheng enn nevnt ovenfor. I denne oppgaven er det ønskelig å benytte et TNS til å navigere et mikrofonmatrise sammen med et kamera i et rom. For å få til det er det blitt benyttet av MEMS-komponenter i form av MEMS-gyroskop og MEMS-akselerometer.

Treghetsnavigasjon baserer seg på akselerasjon og rotasjon av flere treghtessensorer. Det er tre typer treghtessensorer som benyttes i en TNS. Disse er gyroskop, akselerometer og magnetometer. Den meste vanlige kombinasjonen av disse er gyroskop og akselerometer. Magnetometer er en treghtessensorer som er med i et navigasjonssystemet hvis det er behov for det. Du vil ofte finne TNS uten en magnetometer.

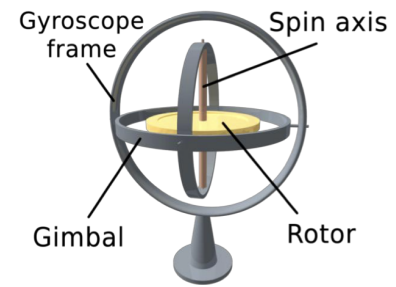


Figur 42: Diverse bevegelsesretninger til et 3-akset akselerometer og et 3-akset gyro i et treghtessnavigasjonssystem med 6-DOF



### 3.2.3.1 Gyroskop

Gyroskop er en enhet som brukes for å måle og opprettholde orientering og vinkelhastighet i et TNS. Et gyroskop som vist på Figur 43 består av flere ringer utenpå hverandre og et hjul i midten. Hjulet i midten av ringene er kalt "rotor" og ringene er kalt "gimbal". Disse ringene gir hjulet friheten til å bevege seg i alle retninger til et 3D-plan. Ringene er selv støttet med hvert sitt spindel. Så lenge hjulet spinner så prøver gyroskopet å holde sin posisjon stabilt på samme sted. Det er viktig å merke at sanntids orientering må være kjent til enhver tid. Dette oppnås ved å bruke opprinnelige orientering av TNS som startverdi og integrere vinkelhastigheten. For at momentet i gyroskopet ikke skal bli mindre på grunn av friksjon i luften som bremser ned hjulet over tid, må gyroskopet spinne med høy hastighet.



Figur 43: Gyroskop

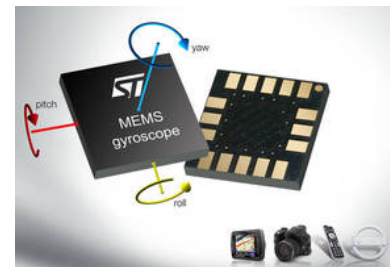
#### 3.2.3.1.1 MEMS-Gyroskop [35]

I den virkelige verden vil et konvensjonell gyroskop være av typen mekanisk- eller optisk-gyroskop. Begge disse typene krever at deres deler skal ha høy presisjon og vanskelig monterings-teknikk. For å lage disse gyroskopene brukes det fortsatt et høyt antall med deler. Dette medfører høye kostnader ved produksjon av gyroskop. MEMS-gyroskop er det motsatte av det konvensjonelle gyroskopet. Den er først og fremst billig å produsere. Det brukes ikke mange deler for å lage en MEMS-gyroskop. Det blir brukt silisium mikro-maskineringsteknikk for å lage dem. MEMS-gyroskop virkemåte er basert på corioliseffekten. Det vil si at en referanseramme som har en rotasjon med vinkelhastigheten  $\omega$ , en masse  $m$  som beveger med en fart  $v$  vil oppleve en kraft:

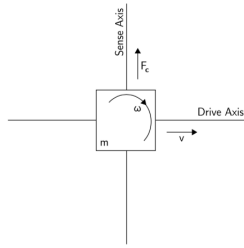
$$F_c = -2m (\omega \times v)$$

Et MEMS-gyroskop inneholder vibrerende elementer for å måle corioliseffekten. Det finnes en del geometriske fremstillinger av vibrerende elementer. Den enkleste geometrien inneholder en enkelt-masse som drives til å vibrere langs en drivakse. Dette er vist med figur 45.

Det som er ulempen ved MEMS-gyroskop er at de kan være litt mindre presise enn et vanlig gyroskop, men fordelene med et MEMS-gyroskop er veldig mange. De er robuste og har lav vedlikehold. De er billigere og derav kan masseproduseres uten store kostnader. De har lav energiforbruk/effektforbruk og i tillegg er små og veier lite. Dette fører til at de passer i navigasjonssystemer som er integrert i f.eks smart-telefoner, digitale-kamera og diverse andre lignende objekter.



Figur 44: MEMS-Gyroskop fra ST



Figur 45: En vibrerende masse gyroskop

### 3.2.3.2 Akselerometer

Et akselerometer måler endringer i hastigheten til et fartøy eller et objekt. Den måler akselerasjonen og tyngdekraften den gjennomgår. Det vil alltid være komponenter på objektet som blir påvirket av tyngdekraften. Denne påkjenningen må erstattes fordi akselerometer har ikke mulighet til å skille tyngdekraften fra hastighetsendringen som blir målt. Hastighet og tilbakelagt distanse i kjente retninger må beregnes. Det er grunnen til at retningen akselerometeret måler, må være kjent av systemet for at systemet skal erstatte virkning av denne type tyngdekraft. Akselerometere finnes i enkel-, dobbel- og fleraksmodell. Disse modellene er blitt laget fordi det skal være mulig å måle størrelsen og retningen på akselerasjonen som en vektorstørrelse. Gyroskopet kan «huske» en retning i rommet, og ved å montere slike ved akselerometrene kan det etableres et sett med referanseretninger [22].

#### 3.2.3.2.1 MEMS-Akselerometer [35]

MEMS-akselerometer er på samme måte som MEMS-gyroskop også laget av silisium mikromaskeringsteknikk. De samme egenskapene som finnes i MEMS-gyroskop vil du også finne i MEMS-akselerometer. Forskjellen mellom disse to prinsippene er som nevnt tidligere at den ene tar for seg målinger av vinkelhastighet imens den andre tar for seg målinger av akselerasjonen i et TNS.

### 3.2.3.3 Markov-prosess

Den kontinuerlige prosessen  $x(t)$  defineres som en 1. ordens Markov-prosess hvis:

$$t_1 < t_2 < \dots < t_k \tag{50}$$

$$p[x(t_k)|x(t_{k-1}), \dots, x(t_1)] = p[x(t_k)|x(t_{k-1})] \tag{51}$$

Fra definisjonen og likningene ovenfor kan det forklares at 1. ordens Markov-prosess brukes for å formulere de fysiske prosessene på en diskret måte. Dette gjøres når det er ønskelig å simulere MEMS-treghetssensorer på en datamaskin. En Markov-prosess sier at sannsynlighetstettheten  $p(x)$  er kun avhengige av den forrige verdien for en nåværende tilstandsverdi. Gauss-Markov prosess er et velkjent utvidelse av Markov-prosessen. Denne prosessen blir brukt når støyen er hvit og normalfordelt, det vil si:

$$N \sim p[x(t_k)|x(t_{k-1})] \tag{52}$$

### 3.2.3.4 Støy

Alle treghetssensorer har en eller annen form for støy. Denne støyen fører til at målte verdier vil få en viss form for unøyaktighet. Den beste måten til å finne et estimat av tilstandene som kan gjøre opp for denne unøyaktigheten er ved å lage en modell av støyen. Denne modellen må representere støyen så nøyaktig som mulig. Markov prosess eller også kalt Gauss-Markov prosess er en måte å få treghetssensorer til å modelleres som farget støy. Da brukes det hovedsakelig 1.orden Gauss-Markov prosess for å få til modelleringen [16].

Det er tre begreper som er viktige å legge merke til når vi refererer til støy i gyroen og akselerometeret til et treghetsnavigasjonssystem. Den første er Angular Random Walk (ARW) som er hvit støy i vinkelhastigheten. Rate Random Walk (RRW) er den type fargestøy som har lang tidskonstant og har egentlig sitt opphav fra hvit støy i vinkelakselerasjonen. Tilslutt har vi noe som heter Bias Instability som er modellert som fargestøy med kort tidskonstant. Overgangen fra hvit støy til fargestøy for RRW gjøres ved å integrere opp prosessen.

Som vi vet består en TNS av gyro og akselerometer. Og støybegrepene for begge sensorene er helt like utenom at Velocity Random Walk (VRW) er hvit støyen på akselerasjonsmålingen [16].

#### 3.2.3.4.1 Hvit støy [6]

Hvit støy er en kontinuerlig stokastisk signal med en konstant PSD  $S(j\omega) = S$ . Dette betyr at amplituden til signalet er uavhengig av den absolutte frekvensen. Noe som igjen betyr at likningen for autokorrelasjon kan skrives

$$R(\tau) = S\delta(\tau)$$

der  $\delta(\tau)$  er Dirac delta funksjonen som defineres ut ifra integralet  $\int_{-\infty}^{\infty} \delta(t)dt = 1$  og  $\delta(t) = 0$  for  $t \neq 0$ . Med denne definisjonen vil hvilken som helst prøve fra den hvite støyen ha en uendelig varians, noe som ikke gir en fornuftig fremstilling. Det er fortsatt en nyttig modell, men når den hvite støyen brukes som en inngang til en modell av et fysisk system, er båndbredden av signalet begrenset, slik at det er mulig å oppnå utgangssignalet fysisk.

Kontinuerlig hvit støy er analog med en diskret hvit sekvens. Sekvensen har null middelvei og ukorrelerte tilfeldige variabler. Siden variablene er ukorrelerte, kan kovariansen av den hvite sekvensen bli skrevet som

$$E[x_k x_l] = \sigma^2 \delta_{kl}$$

hvor  $\delta_{kl}$  er kjent som Kronecker delta, og kan defineres som  $\delta_{kl} = 1$  for  $k = l$  og  $\delta_{kl} = 0$  for  $k \neq l$ .

### 3.2.3.4.2 Farget støy

Målingene til treghetssensorer er på en eller annen måte påvirket av farget støy. For å få gode modeller av den reelle støyen fra en treghetssensor er det best å bruke 1. ordens Gauss-Markov prosess til å modellere den fargede støyen. Måten det gjøres på er å benytte seg av differensiallikningen 53 [16]:

$$\frac{dx}{dt} + \frac{1}{T}x = v \quad (53)$$

Denne differensiallikningen har sitt opphav fra den kontinuerlige prosessen  $x(t)$  hvis den er en 1. ordens Gauss-Markov prosess. I denne likningen er  $v$  hvit støy og  $x$  drevet av hvit støy. Den fargede støyen kan vi definere ved å bytte tilstandsvariabelen  $x$  med  $\mu$ . Da vil differensiallikningen se sånn ut:

$$\dot{\mu}^g = -\frac{1}{T}\mu^g + v_\mu \quad (54)$$

hvor  $v_\mu$  er hvit støyen til bias-instability.

### 3.2.3.5 Eulers integrasjonsmetode [14]

Eulers integrasjonsmetode er en numerisk metode for å løse 1. ordens differensiallikninger med en gitt initial verdi ( $x_0$ ). Det er den mest grunnleggende eksplisitte metoden for numerisk integrasjon av vanlige differensiallikninger. Euler metoden er i første omgang brukt som den grunnleggende metoden, før man går over til mer komplekse metoder. Det blir ofte referert til den simpleste Runge-Kutta metoden.

For en første ordens differensiallikning:

$$\frac{dx}{dt} = f(x, t)$$

som beskriver en kurve, vil ved bruk av Eulers metode bli tatt to avgjørelser for videre utregning. Den første er å bestemme en steplengde kalt "h". Det andre er å bestemme en initialverdi ( $x_0$ ) for 1. ordens differensiallikning. Deretter så beregnes det punkter for hvert iterasjon:

$$x_{n+1} = x_n + h(x_n, t), \text{ hvor } t_n = n * h. \quad (55)$$

### 3.2.3.6 Spesifikk kraft

Notasjon -  $\underline{f}^b$ .

Når vi snakker om spesifikk kraft, refereres det til akselerometeret i TNS. Denne kraften baserer seg på målinger til et akselerometer. Disse målingene er summen av gravitasjonskrefter som er påvirkende i rammen og akselerasjonen for treghetsrommet. Derfor er det misvisende å kalle det for kraft når det egentlig er en form for akselerasjon. Den spesifikke kraften, sett i  $\mathcal{F}^b$  rammen og representert i  $\mathcal{F}^n$  vil bli presentert matematisk med likningen:

$$\vec{f} = \vec{a}^n + \vec{g}^n \quad (56)$$

Videre kan likning 56 utledes til å vise hvordan vi går fra sann-verdi til målt-verdi med likningene listet nedenfor:

$$\underline{f}^b = R_n^b \underline{a}^n + R_n^b \underline{g}^n \quad (57)$$

$$\delta \underline{f}^b = \underline{f}^b - \tilde{f}^b \quad (58)$$

$$\delta \underline{f}^b = \underline{\beta}^a + \underline{v}^a \quad (59)$$

Den målte spesifikke kraften  $\tilde{f}^b$  fra akselerometer kan nå uttrykkes som:

$$\underline{\tilde{f}}^b = \underline{f}^b + \underline{\beta}^a + \underline{v}^a \quad (60)$$

### 3.2.3.7 Vinkelhastighet

Notasjon -  $\underline{\omega}_b^{nb}$

Det er blitt forklart i kapitlet for gyroskopet hva en gyro har som egenskaper i et TNS. Kort oppsummert har den som formål å finne retningen eller stillingen til objektet. Den måler vinkelhastighet fra treghetsrammen, men i vårt tilfelle så kan vi anta det samme som spesifikke kraften at disse målingene vises i  $\mathcal{F}^b$  rammen.

Vinkelhastigheten er representert som et kolonnevektor:

$$\underline{\omega}_b^{nb} = [\omega_1^{nb}; \omega_2^{nb}; \omega_3^{nb}] \quad (61)$$

Videre kan vi utlede hele likningen akkurat som før, fra sann-verdi til måleverdi:

$$\delta \underline{\omega}_b^{nb} = \underline{\omega}_b^{nb} - \tilde{\omega}_b^{nb} \quad (62)$$

$$\delta \underline{\omega}_b^{nb} = \underline{\beta}^g + \underline{v}^g + \underline{\mu}^g \quad (63)$$

$$\tilde{\omega}_b^{nb} = \underline{\omega}_b^{nb} + \underline{\beta}^g + \underline{v}^g + \underline{\mu}^g \quad (64)$$

Vi bruker banegeneratoren til å skaffe verdier til det fysiske systemet og måle systemet. Med verdier så referer vi til  $\underline{\omega}_b^{nb}$  og  $\underline{f}^b$  fra gyroskopet og akselerometeret.

### 3.2.3.8 Tre-akset plattform, flat ikke-roterende jord [9]

For en tre-akset kan feilen defineres på flere måter. I dette kapittelet redegjør vi transformasjonsfeilen til 2 modeller, men kommer til å bruke en av dem.

#### 3.2.3.8.1 Transformasjonsfeilen referert Eulervinklene

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^b - \tilde{\underline{f}}$	$\delta \underline{x}^n = \underline{x}^n - \tilde{\underline{x}}^n$	$\delta \underline{\theta} = \underline{\theta} - \tilde{\underline{\theta}}$
$\delta \underline{\omega} = \underline{\omega}_{np}^b - \tilde{\omega}_{np}^b$	$\delta \underline{v}^n = \underline{v}^n - \tilde{\underline{v}}^n$	

Tabell 1: Transformasjonsfeilen referert Eulervinklene

#### 3.2.3.8.2 Transformasjonsfeilen referert n-systemet

Vi skal her se på feillikningene når feilen i beregningen av rotasjonsmatrisen  $R_b^n$  refereres til n-systemet.

Feildefinisjoner		
$\delta \underline{f} = \underline{f}^b - \tilde{\underline{f}}$	$\delta \underline{x}^n = \underline{x}^n - \tilde{\underline{x}}^n$	$R_b^n = R(\underline{\varepsilon}) \tilde{R}_b^n$
$\delta \underline{\omega} = \underline{\omega}_{nb}^b - \tilde{\omega}_{nb}^b$	$\delta \underline{v}^n = \underline{v}^n - \tilde{\underline{v}}^n$	$R(\underline{\varepsilon}) = I + S(\underline{\varepsilon})$

Tabell 2: Transformasjonsfeilen referert n-systemet, del 1

Fysisk system	Mekanisert system	Feillikninger
$\dot{\underline{x}} = \underline{v}^n$	$\dot{\underline{x}}^n = \underline{\tilde{v}}^n$	$\delta \underline{x} = \delta \underline{v}$
$\underline{\dot{v}}^n = R_b^n \underline{f}^b - \underline{g}^n$	$\underline{\dot{v}}^n = \tilde{R}_b^n \underline{\tilde{f}}^b - \underline{g}^n$	$\delta \underline{\dot{v}}^n = -S(\tilde{R}_b^n \underline{\tilde{f}}^b) \underline{\varepsilon} + \tilde{R}_b^n \delta \underline{f}$
$\dot{R}_b^n = R_b^n S(\underline{\omega}_{nb}^b)$	$\dot{\tilde{R}}_b^n = \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b)$	$\dot{\underline{\varepsilon}} = \tilde{R}_b^n \delta \underline{\omega}$

Tabell 3: Transformasjonsfeilen referert n-systemet, del 2

Bevis av differensial likningen for  $\delta \underline{v}^n$ :

$$\begin{aligned}
\delta \underline{\dot{v}}^n &= R_b^n \underline{f}^b - \underline{g}^n - (\tilde{R}_b^n \underline{\tilde{f}}^b - \underline{g}^n) \\
&= (I + S(\underline{\varepsilon})) \tilde{R}_b^n (\underline{\tilde{f}}^b + \delta \underline{f}) - \tilde{R}_b^n \underline{\tilde{f}}^b \\
&= S(\underline{\varepsilon}) \tilde{R}_b^n \underline{\tilde{f}}^b + \tilde{R}_b^n \delta \underline{f} \\
&= S(\underline{\tilde{f}}^b) \tilde{R}_b^n \underline{\varepsilon} + \tilde{R}_b^n \delta \underline{f}
\end{aligned} \tag{65}$$

Bevis av differensial likningen for  $\underline{\varepsilon}$ :

$$\begin{aligned}
\text{Venstre siden: } \dot{R}_b^n - \dot{\tilde{R}}_b^n &= S(\underline{\dot{\varepsilon}}) \tilde{R}_b^n + (I + S(\underline{\varepsilon})) \dot{\tilde{R}}_b^n - \dot{R}_b^n \\
&= S(\underline{\dot{\varepsilon}}) \tilde{R}_b^n + S(\underline{\varepsilon}) \dot{\tilde{R}}_b^n \\
&= S(\underline{\dot{\varepsilon}}) \tilde{R}_b^n + S(\underline{\varepsilon}) \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b)
\end{aligned} \tag{66}$$

$$\begin{aligned}
\text{Høyre siden: } R_b^n S(\underline{\omega}_{nb}^b) - \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b) &= (I + S(\underline{\varepsilon})) \tilde{R}_b^n (S(\underline{\tilde{\omega}}_{nb}^b) + S(\delta \underline{\omega})) - \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b) \\
&= S(\underline{\varepsilon}) \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b) + \tilde{R}_b^n S(\delta \underline{\omega})
\end{aligned} \tag{67}$$

Satt sammen:

$$S(\underline{\dot{\varepsilon}}) \tilde{R}_b^n + S(\underline{\varepsilon}) \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b) = S(\underline{\varepsilon}) \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b) + \tilde{R}_b^n S(\delta \underline{\omega}) \tag{68}$$

$$S(\underline{\dot{\varepsilon}}) \tilde{R}_b^n = \tilde{R}_b^n S(\delta \underline{\omega}) \tag{69}$$

$$S(\underline{\dot{\varepsilon}}) = \tilde{R}_b^n S(\delta \underline{\omega}) \tilde{R}_b^n \Leftrightarrow \underline{\dot{\varepsilon}} = \tilde{R}_b^n \delta \underline{\omega} \tag{70}$$

### 3.2.4 Sammendrag av likninger for vårt system [25]

Frem til nå har de fleste kapitler i denne rapporten handlet om det generelle aspektet rundt oppgaven og drøftet ulike måter å løse problemstillinger. Det vil si at teorien, matematikken, bilde representasjonen og diverse andre typer informasjon har hovedsakelig vært med på å vise hvordan det er mulig å forme løsningen på et generelt grunnlag. Alle likninger frem til nå kan også brukes til andre type problemstillinger som ikke er direkte relatert til denne oppgaven. Det samme vil gjelde for teorien forklart frem til nå. Videre i oppgaven vil alle relevante likninger som er nevnt i rapporten brukes spesifikt og systematisk til å lage et TNS.

1. Det første som skal redegjøres for er kamera og dens koordinater i rommet:

Fra vår plantegning (figur 29) kan vi beskrive de forskjellige parametrene ved å definere at vi kjenner til posisjonen  $P_i$  i rommet og vil finne ut koordinater for denne posisjonen i kameraet.

Kamera koordinater finner vi med hjelp av likning 42, men skrevet på en ryddig måte som er forklarende og spesifisert for denne oppgaven:

$$\begin{bmatrix} z_i \\ z_i \end{bmatrix} = \begin{bmatrix} s_x \\ s_y \end{bmatrix} = \frac{f}{\rho_3^c} \begin{bmatrix} \rho_1^c \\ \rho_2^c \\ 1 \end{bmatrix} \quad (71)$$

Hvor  $\underline{\rho}^c = [\rho_1^c; \rho_2^c; \rho_3^c]$  er posisjonvektoren for kamera i forhold til punktene  $P_i$  i verden. Videre kan vi uttrykke  $\vec{\rho}$  ved å omrokkere likningen til representasjon av  $P_i$  i forhold til alle rammene ( $\mathcal{F}^n$ ,  $\mathcal{F}^b$  og  $\mathcal{F}^c$ ):

Med  $\vec{\rho}_c$  definert som:

$$\underline{\rho}^c = \begin{bmatrix} \rho_1^c \\ \rho_2^c \\ \rho_3^c \end{bmatrix} \quad (72)$$

Og punkter i rommet som er kjent og sett fra ramme  $\mathcal{F}^n$ :

$$P_i = O_n + \vec{r}_i \quad (73)$$

Kan vi uttrykke dette på vektorform:

$$\vec{r}_i = \vec{p}_n + \vec{p}_{bc} + \vec{\rho}_i \quad (74)$$



Videre utleder vi for vektor  $\vec{\rho}_c$  som representerer punkter i kameraets bildeplan:

$$\vec{\rho}_i = \vec{r}_i - \vec{p}_n - \vec{p}_{bc} \quad (75)$$

Likning 74 og 75 skrevet på algebraisk form for å gjøre utregning til punkter i kamera enklere:

$$\underline{r}^n = \underline{p}^n + R_b^n \underline{p}_{bc}^b + R_b^n R_c^b \underline{\rho}^c \quad (76)$$

$$\underline{\rho}^c = R_b^{c-1} R_n^{b-1} (\underline{r}^n - \underline{p}^n - R_b^n \underline{p}_{bc}^b) \quad (77)$$

Hvor vi definerer:

$$R_c^b \text{ og } \underline{p}_{bc}^b : \text{ kjent og konstant} \quad (78)$$

og vi vet at en ortogonal RKM vil være:

$$R_b^{c-1} = R_b^{cT} \quad (79)$$

Kan vi skive likning 77 litt mer ryddig:

$$\underline{\rho}^c = R_b^c R_n^b (\underline{r}^n - \underline{p}^n - R_b^n \underline{p}_{bc}^b) \quad (80)$$

2. Siden sirkelbanen som er laget med banegeneratoren er en del av treghetsrammen  $\mathcal{F}^n$  kan vi hente både spesifikk kraft  $\underline{f}^b$  og vinkelhastigheten  $\underline{\omega}_b^{nb}$  fra likningene vist i kapittel 3.2.1. Posisjonen  $\underline{p}^n$ , farten  $\underline{v}^n$  og akselerasjonen  $\underline{a}^n$  rundt sirkelbanen kan vi hente fra likning 4.

For å komponere spesifikk kraft er formelen gitt:

$$\underline{f}^b = \underline{a}^b + \underline{g}^b \text{ (se likning 57)} \quad (81)$$

hvor gravitasjonen vil være nedover i z-retning:

$$g^n = \begin{bmatrix} 0 \\ 0 \\ g_3 \end{bmatrix}$$

Vinkel hastigheten  $\omega_b^{nb}$  for vi ved å:

$$\omega_b^{nb} = R_n^b \omega_b^{nn} \quad (82)$$

I kapittel 3.2.3 var vi innom litt av det overfladiske vedrørende et treghetsnavigasjonssystem (TNS). Derfor er det litt viktig å spesifisere prinsippet bak et TNS for å komme til den matematiske modellen. TNS er basert på Newtons 1. bevegelseslov.

*“Every object persists in its state of rest or uniform motion - in a straight line unless it is compelled to change that state by forces impressed on it.”*

- Sir Isaac Newton

Enkelt forklart vil det si at når et legeme er i ro forblir den i ro og når et legeme er i bevegelse forblir den i bevegelse hvis ikke den har påvirkning fra ytre krefter. For at denne loven skal være gjeldende må legemet befinne seg i et treghetsramme (en ikke-roterende ramme uten interne krefter som kan være påvirkende, som for eksempel gravitasjon).

Newtons 2. bevegelseslov som er

*“ Force is equal to the change in momentum per change in time. For a constant mass, force equals mass times acceleration.”*

- Sir Isaac Newton

kan uttrykkes matematisk:

$$F = ma \quad (83)$$

der  $F$  er kraft,  $m$  er masse og  $a$  er akselerasjon. Disse to lovene er veldig viktig i et TNS sammenheng.

Et treghetsnavigasjonssystem er basert på tre egenskaper. Disse er blitt nevnt tidligere, akselerasjon, fart og posisjon. Alle disse fysiske størrelsene er relatert til hverandre. Ved å derivere posisjon  $p$  så vil du få fart  $v$ . Og ved derivere fart vil du få akselerasjon som vist med likning 84.

$$v = \frac{dp}{dt}; \quad a = \frac{dv}{dt} = \frac{d^2p}{dt^2} \quad (84)$$

Derivasjon er forandring av et fysisk størrelse i forhold til en annen fysisk størrelse, hovedsakelig med respekt for tid. Integrasjon er det motsatte av derivasjon. Med integrasjon summeres

alle forandringer som skjer mellom gitte grenseverdier. Det kan være alt fra en til flere iterasjoner. Matematisk er det representert som:

$$v = \int a dt; \quad p = \int v dt = \int \int a dt dt \quad (85)$$

En TNS er ett integrerende system. Den inneholder en sensor og en integrator. Det vil si at den detekterer akselerasjon, integrerer det for å oppnå fart og deretter integrer en gang til for å finne forskyvningen/forflytningen til posisjonen. Ved å måle akselerasjonen til et legeme i et treghets ramme og deretter transformere dette til en navigasjon ramme med respekt for tid. Vil det være mulig å opprettholde fart, posisjon og posisjonsforskjell. Det vil fortsatt være nødvendig å finne ut rotasjonen til legemet for å finne posisjonen og for å få en transformasjon fra treghetsramme til en navigasjonramme.

Med denne informasjonen kan vi nå utlede navigasjons likninger. Den første utledning vil være av den ideelle tilstanden som vist med likning 86. Deretter vil den målte tilstanden utledes som vist med likning 87. Her vil vi også ta høyde for de eksterne påkjenningene som vil påvirke prosessen i form av støy. Vi har allerede vært innom de forskjellige typer støy som er en påvirkende faktor i prosessen.

3. Med punkter i bildeplanet og banegenerator forklart kan vi nå definere våre likninger til navigasjonssystemet. Det vi ønsker er å lage et systemmodell som er fri for noen som helst ekstern påvirkning. Dette i form av støy. Den vil se ut som vist med likning 86:

(86)

$$\dot{\underline{p}} = \underline{v}^n \quad (86a)$$

$$\dot{\underline{v}}^n = \underline{R}_b^n \underline{f}^b - \underline{g}^n \quad (86b)$$

$$\dot{\underline{R}}_b^n = \underline{R}_b^n S(\underline{\omega}_b^{nb}) \quad (86c)$$

Deretter ønsker vi å lage et system basert på målinger som inkluderer støy. Denne modellen er vist med likning 87, hvor det også er blitt tatt høyde for støymodell.

(87)

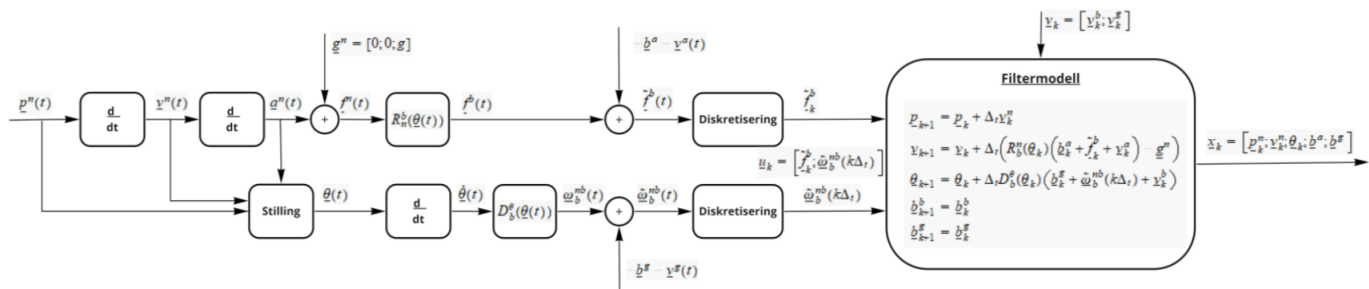
$$\dot{\underline{p}} = \underline{\tilde{v}}^n \quad (87a)$$

$$\dot{\underline{v}}^n = \underline{\tilde{R}}_b^n \underline{\tilde{f}}^b - \underline{g}^n \quad (87b)$$

$$\dot{\underline{\tilde{R}}}_b^n = \underline{\tilde{R}}_b^n S(\underline{\tilde{\omega}}_b^{nb}) \quad (87c)$$

hvor  $\underline{\tilde{f}}^b$  og  $\underline{\tilde{\omega}}_b^{nb}$  er allerede blitt redegjort for i likning 60 og 64.

Det totale systemet for et TNS med kamera er vist med figur 46 som et blockdiagram [12].



Figur 46: Blockdiagram av TNS

### 3.2.5 Stokastiske prosesser [16]

En stokastisk prosess er en prosess som blir definert fra en eller flere stokastiske variabler. En stokastiske variabel er et variabel som blir tilfeldige tildelt eller definert med et verdi for et system. Dette kan beskrives som en funksjon der et vilkårlig forsøk vil være med på å bestemme utfallet. Hensikten med denne prosessen er å indeksere den eller de stokastiske variablene for å forklare endringen som skjer med prosessen. Det er for det meste snakk om endring over tid, derfor er indekseringen definert med en  $t$  for tid. I denne type prosesser vil sannsynlighetsregning være en viktig del for å komme frem til en løsning for problemet.

Den kumulative sannsynlighetstetthetsfordelingen  $F(x)$  er definert som:

$$F(x) = p(\mathbf{X} \leq x)$$

eller som sannsynlighetstetthetsfunksjon  $f(x)$ :

$$f(x) = \frac{dF(x)}{dx}$$

Vi kan forklare forholdet mellom dem ved videreføre de to foregående likningene:

$$F(x) = \int_{-\infty}^x f(x) dx$$

For en stokastisk variabel utledes likningen med en forventningsverdi  $\tilde{x}$  og en varians  $\sigma$  med måten vist nedenfor:

$$\tilde{x} = E\{\mathbf{X}\} = \int_{-\infty}^{\infty} x f(x) dx$$

$$\begin{aligned} \sigma = Var\{\mathbf{X}\} &= \int_{-\infty}^{\infty} (x - E\{\mathbf{X}\})^2 f(x) dx \\ &= E\{(\mathbf{X} - \tilde{x})^2\} \end{aligned}$$

Gjennomsnittet til alle trekkninger som blir utført kalles for forventningsverdi med notasjon  $\tilde{x}$ . Det vil alltid være en vis spredning av utfallene rundt det som er forventet verdi. Dette kalles for varians med notasjon  $\sigma$ , men hvis det er gitt to stokastiske variabler  $\mathbf{X}$  og  $\mathbf{Y}$  kaller vi det for kovarians, som representeres på denne måten:

$$Cov(\mathbf{X}, \mathbf{Y}) = E\{(\mathbf{X} - \tilde{x})(\mathbf{Y} - \tilde{y})\}$$

I Kalmanfilter bruker vi betegnelsen  $P$ . Dette har sitt opphav fra at det er en sammenheng mellom elementene på diagonalen i matrisen som fremstiller kovariansen.

$$P = E\{(\underline{x} - \tilde{x})(\underline{x} - \tilde{x})^T\}$$

### 3.2.6 Diskretisering for en generelt ikke-lineær stokastiske prosess - del 1 [5][13]

Prosesen til et system er vanligvis presentert i et kontinuerlig form, mens målingene er diskrete. For å simulere et ikke-lineært stokastisk system ved hjelp av matematisk verktøy på datamaskiner må prosessen gjøres om til diskret form. En måte å gjøre det på er ved å bruke Eulers integrasjon. Som forklart tidligere i oppgaven er Eulers metode et første ordens numerisk metode til å løse ordinære differensial likninger med en gitt initial verdi.

Vi har et kontinuerlig ikke-lineært system på formen

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{v}), \quad \underline{x}(t_0) \sim \mathcal{N}(\hat{\underline{x}}_0, \hat{P}_0), \quad \underline{v}(t) \sim \mathcal{N}(\underline{0}, \tilde{Q} \cdot \delta(t - \tau)) \quad (88)$$

der  $\underline{x}$  er definert som enten en tilstands vektor eller tilstands matrise,  $\underline{u}$  er pådrags vektor og  $\underline{v}$  er

prosess støy. Initial tilstanden  $\underline{x}(t_0)$  og initial kovariansmatrise  $\hat{P}_0$  er hentet fra normalfordeling med middelvei  $\hat{\underline{x}}_0$  og kovariansen  $\hat{P}_0$ . Støyen er også trukket ut fra normalfordeling med middelvei 0 og en kovarians  $\tilde{Q} \cdot \delta(t - \tau)$  hvor  $\tilde{Q}$  er spektraltetthet for prosessstøyen og  $\delta(t - \tau)$  er Dirac delta funksjon.

Den ikke-lineære målings funksjonen er allerede definert som et diskret likning og kan uttrykkes

$$\underline{z}_k = \underline{h}(\underline{x}_k, \underline{w}_k), \quad \underline{w}_k \sim \mathcal{N}(\underline{0}, R_k \cdot \delta_{kl}) \quad (89)$$

der  $\underline{x}_k$  er den diskrete tilstands vektoren og  $\underline{w}_k$  er den målte støy vektoren på er diskret form. Den målte støyens kovariansmatrise har notasjon  $R_k$  og  $\delta_{kl}$  er Kronecker delta.

For å kunne simulere på datamaskin må prosessen på formen

$$\underline{x}_{k+1} = \underline{f}_k(\underline{x}_k, \underline{u}_k, \underline{v}_k), \quad \underline{v}_k \sim \mathcal{N}(\underline{0}, Q_k \cdot \delta_{kl}) \quad (90)$$

hvor  $Q_k$  er den diskrete prosessstøyen kovariansmatrise.

Vi ønsker at støyen  $\underline{v}_k$  har samme virkning på  $\underline{x}_k \rightarrow \underline{x}_{k+1}$  som  $\underline{v}(t)$  har på  $\underline{x}(k \cdot \Delta) \rightarrow \underline{x}((k+1) \cdot \Delta)$ . For prediksjon bruker vi sub-intervallet  $\Delta$ .

La  $\underline{x}^d$  være løsningen av den deterministiske differensiallikningen

$$\dot{\underline{x}}^d = \underline{f}(\underline{x}^d, \underline{u}, \underline{0}), \quad \underline{x}^d(t_0) = \hat{\underline{x}}_0 \quad (91)$$

Vi antar videre at prosessstøyen er liten slik at bidraget fra den kan antas å finnes ved hjelp av 1. ordensleddet i en Taylorrekke.

$$\left. \begin{aligned} \dot{\underline{x}} &= \underline{f}(\underline{x}, \underline{u}, \underline{v}), & \underline{x}(t_k) &= \underline{x}_k \\ \dot{\underline{x}}^d &= \underline{f}(\underline{x}, \underline{u}, \underline{0}), & \underline{x}^d(t_k) &= \underline{x}_k \end{aligned} \right\} \begin{aligned} \underline{x}(t) &= \underline{x}^d(t) + \delta \underline{x}(t) \\ \delta \underline{x}(t_k) &= 0 \end{aligned} \quad (92)$$

denne prosessen kan skrives som

$$\begin{aligned} \dot{\underline{x}}(t) &= \dot{\underline{x}}^d + \delta \dot{\underline{x}} \\ &= \underline{f}(\underline{x}^d + \delta \underline{x}, \underline{u}, \underline{0} + \underline{v}) \\ &= \underline{f}(\underline{x}^d, \underline{u}, \underline{0}) + F(\underline{x}^d, \underline{u}) \delta \underline{x} + G \underline{v} + h.o.l \end{aligned} \quad (93)$$

Virkningen av prosessstøyen om intervallet  $\Delta$  er tilnærmet gitt av:

$$\delta \underline{\dot{x}} = F \delta \underline{x} + G \underline{v} \quad (94)$$

hvor

$$F = \left. \frac{\partial \underline{f}}{\partial \underline{x}^T} \right|_{\underline{x}^d, \underline{u}} \quad (95)$$

$$G = \left. \frac{\partial \underline{f}}{\partial \underline{v}} \right|_{\underline{x}^d, \underline{u}}$$

og

$$\delta \underline{x}(t_k) \sim \mathcal{N}(\underline{0}, \hat{P}_k) \quad (96)$$

Kovariansen vil ved  $t_{k+1}$  bli

$$\bar{P}_{k+1} = \Phi_k \hat{P}_k \Phi_k^T + \Gamma Q \Gamma \quad (97)$$

hvor

$$\dot{\Phi}(t, t_k) = F(\underline{x}^d(t), \underline{u}(t)) \cdot \Phi(t, t_k), \quad \Phi(t, t_k) = I \quad (98)$$

$$\dot{\Gamma} Q \Gamma^T = \int_{t_k}^{t_k + \Delta} \Phi(t, t_k) G \tilde{Q} G^T \Phi^T(t, t_k) dt \quad (99)$$

For å simulere den stokastiske differential likningen

$$\underline{\dot{x}} = \underline{f}(\underline{x}, \underline{u}, \underline{v}); \quad \underline{x}(t_k) \sim \mathcal{N}(\hat{\underline{x}}, \hat{P}_k) \quad (100)$$

løser vi likningen:

$$\left. \begin{aligned} \underline{\dot{x}}^d &= \underline{f}(\underline{x}^d, \underline{u}, \underline{0}); \quad \underline{x}^d(t_k) = \underline{x}_k \\ \delta \underline{\dot{x}} &= F(\underline{x}^d, \underline{u}) \delta \underline{x} + G(\underline{x}^d, \underline{u}) \underline{v}; \quad \delta \underline{x}(t_k) \sim \mathcal{N}(\underline{0}, \hat{P}_k) \end{aligned} \right\} t \in [t_k, t_{k+1}] \quad (101)$$

dersom  $\underline{u}(t) = \underline{u}_k$  for  $t \in [t_k, t_{k+1}]$  kan likningen skrives

$$\underline{x}_{k+1}^d = \underline{f}_k(\underline{x}_k, \underline{u}_k), \quad \delta \underline{x}_{k+1} = \Gamma_k \underline{v}_k \implies \underline{x}_{k+1} = \underline{f}_k(\underline{x}_k, \underline{u}_k) + \Gamma_k \underline{v}_k \quad (102)$$

Diskretisering vha. Euler-integrasjon og 1. ordensledd i Taylorrekker.

$$\dot{\underline{x}}^d = \underline{f}(\underline{x}^d, \underline{u}, 0); \quad \underline{x}^d(t_k) = \underline{x}_k \quad (103)$$

$$\underline{x}_{k+1}^d = \underline{x} + \Delta \underline{f}(\underline{x}_k, \underline{u}) = \underline{f}(\underline{x}_k, \underline{u})$$

$$F(\underline{x}^d(t), \underline{u}(t)) \approx F(\underline{x}_k, \underline{u}_k) = F_k \quad (104)$$

$$\begin{aligned} \Gamma_k Q_k \Gamma_k^T &= \int_{t_k}^{t_{k+1}} \Phi(t, t_k) G(\underline{x}^d, \underline{u}) \tilde{Q}(t) G^T(\underline{x}^d, \underline{u}) \Phi^T(t, t_k) dt \\ &\approx \int_{t_k}^{t_{k+1}} e^{F_k(t-t_k)} G(\underline{x}^d, \underline{u}) \tilde{Q}(t) G^T(\underline{x}^d, \underline{u}) (e^{F_k(t-t_k)})^T dt \\ &\approx \Delta G(\underline{x}^d, \underline{u}) \tilde{Q}(t) G^T(\underline{x}^d, \underline{u}) \end{aligned} \quad (105)$$

Dersom  $\tilde{Q} = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_{n_v}^2) = \underline{\sigma} \underline{\sigma}^T$  blir  $\Gamma_k = \Delta^{\frac{1}{2}} G \underline{\sigma}$  og  $Q_k = I_{n_v n_v}$ .

Da kan prosessen skrives på en diskret form

$$\underline{x}_{k+1} = (\underline{x} + \Delta \underline{f}(\underline{x}_k, \underline{u})) + \Gamma_k \underline{v}_k = \underline{f}(\underline{x}_k, \underline{u}) + \Gamma_k \underline{v}_k \quad (106)$$

og når  $\tilde{Q}$  er diagonal vil du få

$$\Gamma_k = \Delta^{\frac{1}{2}} G \underline{\sigma}, \quad \underline{v}_k \sim \mathcal{N}(\underline{0}, I_{n_v n_v}) \quad (107)$$



### 3.2.7 UKF - Unscented Kalmanfilter [11]

Før vi starter å forklare hvordan en Unscented Kalmanfilter fungerer og utleder de forskjellige likningene som vil bli brukt i denne oppgaven. Så er det viktig å nevne kort hva en Kalmanfilter og en Extended Kalmanfilter (utvidet kalmanfilter) er.

- KF - Kalmanfilter

KF er en matematisk algoritme for å optimalisere tilstanden til en prosess basert på estimerte og faktiske målinger. Disse resultatene vil også være påvirket av diverse type støy. Støy i dette tilfellet være definert som prosessstøy og målestøy. KF brukes helst på lineære systemer. KF bygger på minimum-varians-prinsippet [1] og det vil alltid være mulig å bruke gaussisk fordeling på støy som er relatert til systemet.

- EKF - Extended Kalmanfilter

KF kan bare brukes på lineære systemer, men i den virkelige verden så er de fleste systemer ikke-lineære. Det vil si at systemet kan inneholde alt fra vinkler til sinus og/eller cosinus funksjoner. I dette tilfelle vil ikke du få brukt gaussisk fordeling på systemet. Og når du ikke får brukt gaussisk fordeling så vil ikke det være mulig å beregne en forventningsverdi eller variansen til systemet. Derfor er det viktig å først linearisere systemet. Måten det gjøres på er å bruke en matematisk formel som heter Taylors formel [15]. Med denne formelen vil det gå an å få en lineært tilnærming av en ikke-lineært system. Denne tilnærmingen er det som gir oss et EKF.

Unscented Kalmanfilter [28] - Det som en Extended Kalmanfilter gjør, det gjør en Unscented Kalmanfilter mye bedre. Det er den enkleste beskrivelsen av et Unscented Kalmanfilter. Den litt mer utfyllende beskrivelsen av en Unscented Kalmanfilter [34] er at den brukes for å få bedre resultat ved å forsiktig pukke ut noen prøve punkter også kalt sigma punkter.

#### 3.2.7.1 Unscented transformasjon

Unscented transformasjon innebærer å lage et sett med sigmapunkter. Hvert sigmapunkt vektet. Deretter gjennom en ikke-lineær funksjon transformeres disse sigmapunktene. Til slutt lages det en gaussisk fordeling av de vektete sigmapunktene, som vist med figur 47.

Gitt en funksjon

$$\underline{y} = \underline{g}(\underline{x}) \quad (108)$$

$$E\{\underline{x}\} = \bar{\underline{x}}; \text{Kov}\{\underline{x}\} = \bar{P} \quad (109)$$

hvor  $\underline{x}$  er en stokastisk vektor med kjent middelvei og kovariansmatrise.

En unscented transformasjon kan deles i tre trinn:

1. Beregn sigmapunktene,  $\underline{x}^i$ , og de tilhørende vekten,  $w^i$ , for  $\bar{x}$  og  $\bar{P}$ .
2. Kjør sigmapunktene gjennom transformasjonen (funksjonen) og beregn sigmapunktene til utgangen,  $\underline{y}^i$ .
3. Beregn middelværdi og kovarians til funksjonen vha sigmapunktene  $\underline{y}^i$  og vektene  $w^i$ .

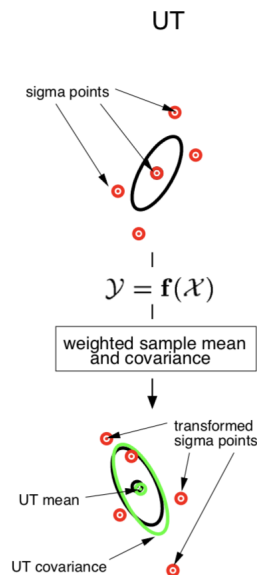
$$\bar{y} = \sum_i w^i \underline{y}^i \quad (110)$$

$$\bar{P} = \sum_i w^i (\underline{y}^i - \bar{y})(\underline{y}^i - \bar{y})^T \quad (111)$$

Vektene må alltid tilfredstille

$$\sum_i w^i = 1 \quad (112)$$

Alle vektene trenger ikke å være større enn null, men dersom de er det kan de tolkes som en punktmassefunksjon.



Figur 47: Unscented transformasjon

### 3.2.7.2 Utvidet symmetrisk sett av sigmapunkter - Sigmasett 2

En utvidet sett av sigmapunkter:

$$p = 2n_x + 1 \quad (113)$$

$$P = UU^T; U = \begin{bmatrix} \underline{u}^1 & \underline{u}^2 & \dots & \underline{u}^{n_x} \end{bmatrix} \quad (114)$$

$$\underline{x}^0 = \bar{x} \quad (115)$$

$$\text{Velg en } w^0 = 1 - \frac{n_x}{3}, \text{ se nedenfor} \quad (116)$$

$$\underline{x}^i = \bar{x} + \sqrt{\frac{n_x}{1-w^0}} \underline{u}^i \text{ for } i \in \{1, 2, \dots, n_x\} \quad (117)$$

$$\underline{x}^{i+n_x} = \bar{x} - \sqrt{\frac{n_x}{1-w^0}} \underline{u}^i \text{ for } i \in \{1, 2, \dots, n_x\} \quad (118)$$

$$w^i = \frac{1-w^0}{2n_x} \text{ for } i \in \{1, 2, \dots, 2n_x\} \quad (119)$$

$$\underline{y}^i = \underline{g}(\underline{x}^i) \text{ for } i \in 0, 1, 2, \dots, 2n_x \quad (120)$$

$$\bar{y} = \sum_{i=0}^{2n_x} w^i \underline{y}^i; \bar{P}_y = \sum_{i=0}^{2n_x} w^i (\underline{y}^i - \bar{y})(\underline{y}^i - \bar{y})^T \quad (121)$$

### 3.2.7.3 UKF med ikke-additiv støy

Gitt systemet

$$\underline{x}_{k+1} = \underline{f}(\underline{x}_k, \underline{u}_k, \underline{v}_k) \quad (122)$$

$$\underline{z}_k = \underline{h}(\underline{x}_k, \underline{w}_k) \quad (123)$$

$$E\{\underline{x}_0\} = \bar{x}_0; E\{\underline{v}_k\} = \underline{0}; E\{\underline{w}_k\} = \underline{0} \quad (124)$$

$$Kov\{\underline{x}_0\} = \bar{P}_0; E\{\underline{v}_k \underline{v}_l^T\} = \delta_{kl} Q_k; E\{\underline{w}_k \underline{w}_l^T\} = \delta_{kl} R_k; \quad (125)$$

$$E\{\underline{x}_0 \underline{v}_k^T\} = 0; E\{\underline{x}_0 \underline{w}_k^T\} = 0; E\{\underline{v}_k \underline{w}_l^T\} = 0; \quad (126)$$

Med sigmasett 2 er  $n_p = +1$ :

1. Beregnet  $n_p = 2(n_x + n_v + n_w) + 1$  sigmapunkter (sigmasett 2) og vektor for det augmenterte system:

$$\hat{\underline{x}}_k^a = \begin{bmatrix} \hat{\underline{x}}_k \\ \underline{0} \\ \underline{0} \end{bmatrix}; \hat{P}_k^a = \begin{bmatrix} \hat{P}_k & 0 \\ 0 & Q_k \\ & & R_{k+1} \end{bmatrix} \quad (127)$$

Vektene og sigmapunktene blir:

$$w^0 = 1 - \frac{n_a}{3}; n_a = n_x + n_v + n_w \quad (128)$$

$$w^i = \frac{1 - w^0}{2n_a} \text{ for } i \in \{1, 2, \dots, 2n_a\} \quad (129)$$

$$U_k^a = \text{chol}(\hat{P}_k^a)^T; \underline{u}_k^{a,i} = \text{col}_i(U_k^a); \hat{P}_k^a = U_k^a U_k^{a,T} \quad (130)$$

$$\hat{\underline{x}}_k^{a,i} = \hat{\underline{x}}_k^a + \sqrt{\frac{n_a}{1 - w^0}} \underline{u}_k^{a,i} \text{ for } i \in \{1, 2, \dots, n_a\} \quad (131)$$

$$\hat{\underline{x}}_k^{a,i+n_x} = \hat{\underline{x}}_k^a - \sqrt{\frac{n_a}{1 - w^0}} \underline{u}_k^{a,i} \text{ for } i \in \{1, 2, \dots, n_a\} \quad (132)$$

Sigmapunktene blir:

$$\underline{x}_k^{a,i} = \begin{bmatrix} \hat{\underline{x}}_k^i \\ \underline{v}_k^i \\ \underline{w}_{k+1}^i \end{bmatrix}; i = 0, 1, \dots, n_p - 1 \quad (133)$$

2. Beregning av predikert tilstand og kovarians:

$$\bar{x}_{k+1}^{\mathbf{i}} = \underline{f}(\hat{x}_k^{\mathbf{i}}, v_k^{\mathbf{i}}, u_k); \mathbf{i} = 0, 1, \dots, n_p - 1 \quad (134)$$

$$\bar{x}_{k+1} = \sum_{\mathbf{i}=0}^{n_p-1} w^{\mathbf{i}} \bar{x}_{k+1}^{\mathbf{i}} \quad (135)$$

$$\bar{P}_{k+1} = \sum_{\mathbf{i}=0}^{n_p-1} w^{\mathbf{i}} (\bar{x}_{k+1}^{\mathbf{i}} - \bar{x}_{k+1})(\bar{x}_{k+1}^{\mathbf{i}} - \bar{x}_{k+1})^T \quad (136)$$

3. Beregning av predikert måling og dens kovarians:

$$z_{k+1}^{\mathbf{i}} = \underline{h}(\bar{x}_{k+1}^{\mathbf{i}}, w_{k+1}^{\mathbf{i}}); \mathbf{i} = 0, 1, \dots, n_p - 1 \quad (137)$$

$$\bar{z}_{k+1} = \sum_{\mathbf{i}=0}^{n_p-1} w^{\mathbf{i}} z_{k+1}^{\mathbf{i}} \quad (138)$$

$$\bar{P}_{zz}(k+1) = \sum_{\mathbf{i}=0}^{n_p-1} w^{\mathbf{i}} (z_{k+1}^{\mathbf{i}} - \bar{z}_{k+1})(z_{k+1}^{\mathbf{i}} - \bar{z}_{k+1})^T \quad (139)$$

$$\bar{P}_{xz}(k+1) = \sum_{\mathbf{i}=0}^{n_p-1} w^{\mathbf{i}} (x_{k+1}^{\mathbf{i}} - \bar{x}_{k+1})(z_{k+1}^{\mathbf{i}} - \bar{z}_{k+1})^T \quad (140)$$

4. Beregning av estimert tilstand og dens kovarians:

$$K_{k+1} = \bar{P}_{xz}(k+1) \bar{P}_{zz}^{-1}(k+1) \quad (141)$$

$$\hat{P}_{k+1} = \bar{P}_{k+1} - K_{k+1} \bar{P}_{zz}(k+1) K_{k+1}^T \quad (142)$$

$$\hat{x}_{k+1} = \bar{x}_{k+1} + K_{k+1} (z_{k+1} - \bar{z}_{k+1}) \quad (143)$$

5. Start hele prosessen igjen fra punkt 1.

Fra notat 1 av Oddvar Hallingstads notater så har vi: [10]

$$1. K_k = \bar{P}_k H^T (H \bar{P}_k H^T + R)^{-1} \quad (144)$$

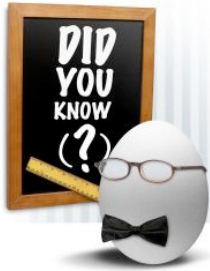
$$2. \hat{P}_k = (I - K_k H) \bar{P}_k \quad (145)$$

$$\begin{aligned} P_{xz} &= E\{(\underline{x} - \bar{x})(\underline{z} - \bar{z})^T\} \\ &= E\{(\underline{x} - \bar{x})(H(\underline{x} - \bar{x}) + w)^T\} \\ &= \bar{P} H^T \end{aligned}$$

$$\begin{aligned} P_{zz} &= E\{(\underline{z} - \bar{z})(\underline{z} - \bar{z})^T\} \\ &= H \bar{P}_k H^T + R \end{aligned}$$

$$\begin{aligned} K &= \bar{P}_{xz} \bar{P}_{zz}^{-1} \\ &= \bar{P} H^T (H \bar{P} H^T + R)^{-1} \quad (dvs 1) \end{aligned} \quad (146)$$

$$\begin{aligned} \hat{P} &= \bar{P} - K \bar{P}_{zz} K^T \\ &= \bar{P} - \bar{P} H^T (H \bar{P} H^T + R)^{-1} (H \bar{P} H^T + R) K^T \\ &= \bar{P} - \bar{P} H^T (H \bar{P} H^T + R)^{-1} H \bar{P} \\ &= (I - KH) \bar{P} \quad (dvs 2) \end{aligned}$$



Fun Fact - Unscented Kalmanfilter har fått sitt navn fra 2 studenter fra Oxford universitet Jeffrey Uhlmann og Simon Julier som mente at Extended Kalmanfilter også kalt "Utvidet kalmanfilter" var en filter som ikke var bra nok. Prinsippet med linearisering rundt et punkt vil ikke gi det beste estimatet i motsetning til hvis du har linearisering rundt flere punkter kalt sigma punkter. De mente at det ville gi mye bedre linearisert estimat. Deres professor var uenig i deres hypotese og sto fortsatt fast på at EKF (utvidet kalmanfilter) var den beste løsningen. Studentene på sin side var sikker i sin sak og mente at utvidet kalmanfilter var en "Shitty filter", men det er ikke derfor Unscented (uparfymert/luktfri) Kalmanfilter fikk sitt navn

fordi den i motsetning til utvidet kalmanfilter "ikke luktet". Opphavet til navnet kommer fra en deodorant som lå på lab-bordet til en av studentene, Jeffrey Uhlmann.

## 4 Metode for simulering - presentert med matematiske likninger

*I dette kapitlet presenteres en del av programmet og likninger som er blitt brukt for å lage vårt system. Det vil også oppgis kode som pseudo-kode for å presisere viktigheten og fremgangsmåten til en del av programmet. Det meste av denne koden og fremstillingen er blitt hentet eller gjenbrukt - med noen modifikasjoner - fra masteroppgavene til Ivan Flatval [5] og Tor Erik Fredrikstad [6]. I tillegg til det er notat fra Oddvar Hallingstad - 'Simulering av TNS' [12] - et viktig bidrag til hele dette kapitlet.*

Frem til nå skal foregående kapitler ha dekket all den nødvendige informasjonen som trengs for å forstå oppgavebeskrivelsen. Problemstillingen som oppgaven startet med i utgangspunktet har både blitt utfordret og forandret. Det har blitt sett på forskjellige tilnærminger til problemstillingen. Dette er blitt gjort for å få til en sammenheng mellom nåværende løsning til oppgaven og den problemstillingen som var tiltenkt i starten. Teori kapitlet tok for seg grundig teorien som var tiltenkt for problemstillingen som var i utgangspunktet og teorien for problemstillingen som til slutt er blitt redegjort for i denne avsluttende delen av oppgaven. Deretter ble de generelle matematiske likningene forklart for alle nødvendige aspekter til oppgaven.

I denne delen av rapporten skal det forklares stegvis programmet som er tiltenkt å simulere for å få resultater til oppgaven. Denne simulering er ment å gjøres i Matlab. Derfor vil også pseudo-koden som kan brukes bli lagt frem sammen med annet relevant informasjon for simulering av programmet.

Det første som gjøres er å generere en bane i halvsirkel som vist med figur 48. Dette plottet viser en visuelt fremstilling av test benken til Mantena AS med massesenteret til boogien som midtpunktet det beveges rundt. Denne banegeneratoren er den analytiske fremstillingen av løsningen. Det er ønskelig å bruke verdiene for banegeneratoren som referanse for filtermodellen. Det er blitt tatt frihet til å velge verdier som gjør matematikken enkel for oppgaven, siden simuleringen skal være en ideell/optimal fremstilling av løsningen.

### 4.1 Oppsettet til banegeneratoren og kameraet, og deres initialverdier

Banegenerator - bevegelse langs en halvsirkel med et massesenter som vist med figur 48.

$$\underline{p}^n = [r\cos(\omega t); r\sin(\omega t); 1]$$

$$\underline{v}^n = [-r\omega\sin(\omega t); r\omega\cos(\omega t); 0]$$

$$\underline{a}^n = [-r\omega^2\cos(\omega t); -r\omega^2\sin(\omega t); 0]$$

Vi finner den sanne spesifikke kraften  $f^b$  og den sanne vinkelhastigheten  $\omega_b^{nb}$  fra banegeneratoren. I tillegg til det må det tas med i betraktning at rotasjonsmatrisen  $R_n^b$  har en horisontal bevegelse rundt boogien. Det betyr at det roterer rundt z-aksen og de to andre aksene er konstante.

$$R_n^b = \begin{bmatrix} \cos(\omega t) & \sin(\omega t) & 0 \\ -\sin(\omega t) & \cos(\omega t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} R_n^b(\underline{\theta}(t)) &= R_b^n(\underline{\theta}(t))^{-1} \\ \text{hvor : } \underline{\omega}_b^{nn} &= [0; 0; \omega^n] \\ \text{og : } g^n &= [0; 0; 9.81] \end{aligned}$$

der av

$$\begin{aligned} \underline{f}^b &= R_n^b \underline{a}^n + R_n^b \underline{g}^n \\ \omega_b^{nb} &= R_n^b \omega_b^{nn} \end{aligned}$$

her er radiusen til banen på 5 meter. Bevegelsen rundt hele banen skjer med en tidsperiode på 30 sekunder imens vi beveger med bare med en halv runde rundt banen. Gravitasjon konstanten er  $9.81 \frac{m}{s^2}$  og tidsinkrementet  $\Delta_t$  er 0.01. I tillegg til det er vinkelhastigheten konstant.

For målelikninger til filtermodellen ser vi på punkter i rommet som projiseres i bildeplanet til kameraet og vises i ramme  $f^c$ . Denne tilnærmingen vil ha en rotasjon og translasjon av punkter i rommet. Hele denne prosessen definerer vi som utgangsverdier  $\underline{z}_k^i$  og kan skrives:

$$\begin{aligned} \underline{\rho}_i^c &= [\rho_1^c; \rho_2^c; \rho_3^c] \\ P_i &= O_n + \vec{r}_i \\ \vec{r}_i &= \vec{p}_n + \vec{p}_{bc} + \vec{\rho}_i \\ \vec{\rho}_i &= \vec{r}_i - \vec{p}_n - \vec{p}_{bc} \\ \underline{r}^n &= \underline{p}^n + R_b^n \underline{p}_{bc}^b + R_b^n R_c^b \underline{\rho}^c \\ \underline{\rho}^c &= R_c^{b^{-1}} R_b^{n^{-1}} (\underline{r}^n - \underline{p}^n - R_b^n \underline{p}_{bc}^b) \\ \underline{\rho}^c &= R_b^c R_n^b (\underline{r}^n - \underline{p}^n - R_b^n \underline{p}_{bc}^b) \end{aligned}$$

Her er det viktig å presisere at  $R_c^b$  og  $\underline{p}_{bc}^b$  er kjent og konstant og at fokallengde  $f$  er lik 1.



$$\begin{bmatrix} z_i \\ z_i \end{bmatrix} = \begin{bmatrix} s_x \\ s_y \end{bmatrix} = \frac{f}{\rho_3^c} \begin{bmatrix} \rho_1^c \\ \rho_2^c \\ 1 \end{bmatrix}$$

## 4.2 Kontinuerlig banegenerator

I det følgende kapitlet utledes de kontinuerlige likningene for hastighet, akselerasjon, akselerometer- og gyromålinger (både med og uten støy) når vi antar at vi kjenner banen  $\underline{p}^n(t)$ , og stillingen til legemet  $R_b^n(\underline{\theta}(t))$ , uttrykket ved 3-2-1 Eulervinklene  $\underline{\theta}(t)$ .

### 4.2.1 Banegenerator uten støy på akselerometer og gyroer

Det er tre måter vi kan sette opp likningene for banegeneratoren. Alle disse tre vil gi det samme resultatet, men med forskjellig tilnærming.

#### 4.2.1.1 Derivasjon av kjent posisjon og stilling

Når posisjonen  $\underline{p}^n(t)$  er kjent og vi kan representere stillinga med Eulervinklene  $\underline{\theta}(t)$  3-2-1, kan vi beregne

$$\begin{aligned} \underline{v}^n(t) &= \dot{\underline{p}}^n(t) \\ \underline{a}^n(t) &= \dot{\underline{v}}^n(t) \\ \underline{f}^b(t) &= R_b^n(\underline{\theta}(t))(\underline{a}^n(t) + \underline{g}^n) \\ \dot{\underline{\theta}}(t) &= \frac{\partial}{\partial t} \underline{\theta}(t) \\ \underline{\omega}_b^{nb}(t) &= D_\theta^b(\underline{\theta}(t)) \dot{\underline{\theta}}(t) \end{aligned} \tag{147}$$

hvor

$$R_b^n(\underline{\theta}(t)) = \begin{bmatrix} c_{\theta_3} c_{\theta_2} & c_{\theta_3} s_{\theta_2} s_{\theta_1} - s_{\theta_3} c_{\theta_1} & c_{\theta_3} s_{\theta_2} c_{\theta_1} + s_{\theta_3} s_{\theta_1} \\ s_{\theta_3} c_{\theta_2} & s_{\theta_3} s_{\theta_2} s_{\theta_1} + c_{\theta_3} c_{\theta_1} & s_{\theta_3} s_{\theta_2} c_{\theta_1} - c_{\theta_3} s_{\theta_1} \\ -s_{\theta_2} & c_{\theta_2} s_{\theta_1} & c_{\theta_2} c_{\theta_1} \end{bmatrix} \tag{148}$$

$$D_{\underline{\theta}}^b(\underline{\theta}(t)) = \begin{bmatrix} 1 & 0 & -\sin(\theta_2) \\ 0 & \cos(\theta_1) & \sin(\theta_1)\cos(\theta_2) \\ 0 & -\sin(\theta_1) & \cos(\theta_1)\cos(\theta_2) \end{bmatrix} \quad (149)$$

#### 4.2.1.2 Formulert som differensiallikning

Hvis  $\underline{f}^b(t)$  og  $\underline{\omega}_b^{nb}(t)$  er kjent, kan vi løse differensiallikningene ved på beregne  $\underline{p}^n(t)$ ,  $\underline{v}^n(t)$  og  $\underline{\theta}(t)$

$$\begin{aligned} \dot{\underline{p}}^n(t) &= \underline{v}^n(t) \\ \dot{\underline{v}}^n(t) &= R_b^n(\underline{\theta}(t))\underline{f}^b(t) - \underline{g}^n \\ \dot{\underline{\theta}}(t) &= D_b^\theta(\underline{\theta}(t))\underline{\omega}_b^{nb}(t) \end{aligned} \quad (150)$$

$\underline{p}^n(t_0)$ ,  $\underline{v}^n(t_0)$  og  $\underline{\theta}(t_0)$  er gitt

#### 4.2.1.3 Formulert som integrallikning

Dersom vi kjenner  $\underline{f}^b(t)$  og  $\underline{\omega}_b^{nb}(t)$  kan også differensiallikningene ovenfor formuleres om integrallikninger (det gir oss mulighet til diskretisering)

$$\begin{aligned} \underline{p}^n(t) &= \underline{p}^n(t_0) + \int_{t_0}^t \underline{v}^n(\tau) d\tau \\ \underline{v}^n(t) &= \underline{v}^n(t_0) + \int_{t_0}^t R_b^n(\underline{\theta}(\tau))\underline{f}^b(\tau) d\tau - (t - t_0)\underline{g}^n \\ \underline{\theta}(t) &= \underline{\theta}(t_0) + \int_{t_0}^t D_b^\theta(\underline{\theta}(\tau))\underline{\omega}_b^{nb}(\tau) d\tau \end{aligned} \quad (151)$$

$\underline{p}^n(t_0)$ ,  $\underline{v}^n(t_0)$  og  $\underline{\theta}(t_0)$  er gitt

fra likningsett 151 er det bare den siste likningen der hvor variabelen opptrer både under og utenfor integraltegnet. Etter å ha funnet  $\underline{\theta}(t)$  kan vi bestemme  $\underline{v}^n(t)$  og deretter  $\underline{p}^n(t)$  som integraler av kjente funksjoner.

### 4.3 Gyro- og akselerometermålinger med støy

Fysiske gyroer og akselerometer gir ut henholdsvis vinkelinkremitter (integral av vinkelhastigheter over samplingsintervallet) og hastighetsinkremitter (integral av spesifikk kraft over samplingsintervallet). En samplingsfrekvens kan være alt mellom 500 til 32000 Hz, men det vil også ha betydning hvilke type sensor det er. For å finne sensorens støybeskrivelse beregnes autokorrelasjon og effektspekter, eventuelt Allan-variansen. For å gi en støybeskrivelse uavhengig av samplingsfrekvensen beregnes de kontinuerlige dataene (dvs vi opererer med spektraltettheter for de hvite støyene).

Når vi velger en tidsoppdateringsfrekvens (enten for simulatoren eller estimatoren) må vi regne om de oppgitte dataene for gyroer og akselerometer til den aktuelle samplingsfrekvensen.

Vi antar at støyen for triader av akselerometre og gyroer bare består av biaser og hvitstøy. For å få plussfortegn på støyen i filterlikningene antar vi at akselerometer- og gyromålinger beregnes som

$$\begin{aligned}
 \underline{\tilde{f}}^b(t) &= \underline{f}^b(t) - \delta \underline{f}^b(t) = \underline{f}^b(t) - \underline{b}^a - \underline{v}^a(t) \\
 \underline{\tilde{\omega}}_b^{nb}(t) &= \underline{\omega}_b^{nb}(t) - \delta \underline{\omega}_b^{nb}(t) = \underline{\omega}_b^{nb}(t) - \underline{b}^g - \underline{v}^g(t) \\
 \underline{b}^a &\sim \mathcal{N}(\underline{0}, \sigma_{b^a}^2 I); \underline{v}^a \sim \mathcal{N}(\underline{0}, \sigma_{b^a}^2 \delta(t - \tau)) \\
 \underline{b}^g &\sim \mathcal{N}(\underline{0}, \sigma_{b^g}^2 I); \underline{v}^g \sim \mathcal{N}(\underline{0}, \sigma_{b^g}^2 \delta(t - \tau))
 \end{aligned} \tag{152}$$

### 4.4 Kontinuerlig prosessmodell

Ved å bruke differensiallikningene for banen fra likningssett 150 og støybeskrivelsene fra likningssett 152 er det mulig å definere  $\underline{f}^b(t) = \underline{\tilde{f}}^b(t) + \delta \underline{f}^b(t)$ ,  $\underline{\omega}_b^{nb}(t) = \underline{\tilde{\omega}}_b^{nb}(t) + \delta \underline{\omega}_b^{nb}(t)$ ,  $\underline{u}^a = \underline{\tilde{f}}^b(t)$ ,  $\underline{u}^g = \underline{\tilde{\omega}}_b^{nb}(t)$  og generatoren kan skrives:

$$\begin{aligned}
 \dot{\underline{p}}^n &= \underline{v}^n \\
 \dot{\underline{v}}^n &= R_b^n(\underline{\theta})(\underline{u}^a + \underline{b}^a + \underline{v}^a) - \underline{g}^n \\
 \dot{\underline{\theta}} &= D_b^\theta(\underline{\theta})(\underline{u}^g + \underline{b}^g + \underline{v}^g) \\
 \dot{\underline{b}}^a &= 0 \\
 \dot{\underline{b}}^g &= 0
 \end{aligned} \tag{153}$$

Hvis vi skriver likningene i en matrise format vil fremstillingen av likningssettet bli mye mer tydelig

$$\begin{bmatrix} \underline{\dot{p}}^n \\ \underline{\dot{v}}^n \\ \underline{\dot{\theta}} \\ \underline{\dot{b}}^a \\ \underline{\dot{b}}^g \end{bmatrix} = \begin{bmatrix} \underline{0} & I & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & R_b^n(\underline{\theta}) & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & \underline{0} & D_b^\theta(\underline{\theta}) \\ \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{p}^n \\ \underline{v}^n \\ \underline{\theta} \\ \underline{b}^a \\ \underline{b}^g \end{bmatrix} \quad (154)$$

$$+ \begin{bmatrix} \underline{0} & \underline{0} & \underline{0} \\ R_b^n(\underline{\theta}) & \underline{0} & -I \\ \underline{0} & D_b^\theta(\underline{\theta}) & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{u}^a \\ \underline{u}^g \\ \underline{g}^n \end{bmatrix} + \begin{bmatrix} \underline{0} & \underline{0} \\ R_b^n(\underline{\theta}) & \underline{0} \\ \underline{0} & D_b^\theta(\underline{\theta}) \\ \underline{0} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{v}^a \\ \underline{v}^g \end{bmatrix}$$

Hvis vi fremstiller vektorene som

$$\begin{aligned}
\underline{x} &= [\underline{p}^n; \underline{v}^n; \underline{\theta}; \underline{b}^a; \underline{b}^g] \\
\underline{u} &= [\underline{u}^a; \underline{u}^g] = [\underline{f}^b; \underline{\tilde{\omega}}_b^{nb}] \\
\underline{v} &= [\underline{v}^a; \underline{v}^g] \\
\underline{z}_k &= [z_k^1; \dots; z_k^{n_p}] \\
\underline{w}_k &= [w_k^1; \dots; w_k^i; \dots; w_k^{n_p}]
\end{aligned} \quad (155)$$

kan vi skrive banegeneratoren på formen

$$\begin{aligned}
\dot{\underline{x}} &= f(\underline{x}, \underline{u}, \underline{v}) \\
\dot{z}_k &= h(\underline{x}) + \underline{w}_k
\end{aligned} \quad (156)$$

## 4.5 Banegenerator på diskret form

Som nevnt tidligere i rapporten må vi diskretisere likningene som er skrevet på en standardform. Dette gjøres fordi det skal være mulig å simulere likningene på en datamaskin med en matematisk program.

### 4.5.1 Banegenerator uten støy på akselerometre og gyroer

Hvis vi setter  $t = k\Delta_t$  for et bestemt tidspunkt vil vi få verdien til en vektorfunksjon som er kjent eller kan beregnes. Dette vi da gi oss

$$\begin{aligned}
 \underline{\theta}_k &= \underline{\theta}(k\Delta_t) \\
 \underline{p}_k^n &= \underline{p}^n(k\Delta_t) \\
 \underline{v}_k^n &= \underline{v}^n(k\Delta_t) \\
 \underline{a}_k^n &= \underline{a}(k\Delta_t) \\
 \underline{f}_k^b &= \underline{f}^b(k\Delta_t) \\
 \underline{\omega}_b^{nb}(k) &= \underline{\omega}_b^{nb}(k\Delta_t)
 \end{aligned} \tag{157}$$

### 4.5.2 Målelikninga for Kalmanfilteret med målestøy

Målelikninga for fastpunkt nr  $i$ :

$$\underline{z}_k^i = \frac{1}{\rho_{i,3}^c(k)} [\rho_{i,1}^c(k); \rho_{i,2}^c(k)] + \underline{w}_k^i \tag{158}$$

$$\rho_i^c = R_b^c R_n^b(\underline{\theta})(\underline{r}_i^n - \underline{p}^n) - R_b^c \underline{p}_{bc}^b \tag{159}$$

### 4.5.3 Akselerometer- og gyromålinger med støy

Siden akselerometrene og gyroene sender ut hastighets- og vinkelinkremerter med høyere frekvenser enn tidsoppdateringinkrementet  $\Delta_t$  antar vi at vi kan modellere dette som en integrasjon av de kontinuerlige akselerasjons- og gyromålingene over  $\Delta_t$ .

Vi krever at de diskrete akselerometer- og gyromålingene skal gi tilnærmet samme middelvei og kovarians for hastighets- og vinkelinkrement over  $\Delta_t$

$$\begin{aligned}
 \Delta_k^v &= \int_{t_k}^{t_{k+1}} \underline{\tilde{f}}^a(t) dt \\
 &= \int_{t_k}^{t_{k+1}} (\underline{f}^a(t) - \underline{b}^a - \underline{v}^a(t)) dt \\
 &= \frac{\Delta_t}{2} (\underline{f}_{k+1}^a + \underline{f}_k^a) - \Delta_t \underline{b}^a - \underline{v}_k^a
 \end{aligned}$$

$$\begin{aligned}
\Delta_k^\theta &= \int_{t_k}^{t_{k+1}} \tilde{\omega}_b^{nb}(t) dt \\
&= \int_{t_k}^{t_{k+1}} (\omega_b^{nb}(t) - \underline{b}^g - \underline{v}^g(t)) dt \\
&= \frac{\Delta t}{2} (\omega_b^{nb}(k+1) + \omega_b^{nb}(k)) - \Delta t \underline{b}^g - \underline{v}_k^g
\end{aligned}$$

Vi beregner derfor de diskrete akselerometer- og gyromålingene som ( innfører standard normalfordeling)

$$\tilde{f}_k^a = \frac{1}{2}(\underline{f}_{k+1}^a + \underline{f}_k^a) - \underline{b}^a - \sqrt{\Delta t} \sigma_{v^a} \underline{n}(3) \quad (160)$$

$$\tilde{\omega}_b^{nb}(k) = \frac{1}{2}(\omega_b^{nb}(k+1) + \omega_b^{nb}(k)) - \underline{b}^g - \sqrt{\Delta t} \sigma_{v^g} \underline{n}(3) \quad (161)$$

$$\underline{b}^a \sim \mathcal{N}(\underline{0}, \sigma_{b^a}^2 I); \underline{v}_k^a = \int_{t_k}^{t_{k+1}} \underline{v}^a(t) dt = \sqrt{\Delta t} \sigma_{v^a} \underline{n}(3) \sim \mathcal{N}(\underline{0}, \Delta t \sigma_{v^a}^2 \delta_{kl}) \quad (162)$$

$$\underline{b}^g \sim \mathcal{N}(\underline{0}, \sigma_{b^g}^2 I); \underline{v}_k^g = \int_{t_k}^{t_{k+1}} \underline{v}^g(t) dt = \sqrt{\Delta t} \sigma_{v^g} \underline{n}(3) \sim \mathcal{N}(\underline{0}, \Delta t \sigma_{v^g}^2 \delta_{kl}) \quad (163)$$

$$\underline{n}_v = \text{randn}(n_s) \quad (164)$$

Utleddning av fordelinga til integralet av hvit støy over et intervall:

$$\begin{aligned}
v(t) &\sim \mathcal{N}(\underline{0}, \tilde{\sigma}^2 \delta(t - \tau)) \\
v_k &= \int_0^{\Delta t} v(t) dt \\
E\{v_k\} &= \int_0^{\Delta t} E\{v(t)\} dt = 0 \\
Kov\{v_k\} &= E\left\{ \int_0^{\Delta t} E\{v(t)\} dt \int_0^{\Delta t} E\{v(\tau)\} d\tau \right\} \\
&= \int_0^{\Delta t} \int_0^{\Delta t} E\{v(t)v(\tau)\} dt d\tau \\
&= \int_0^{\Delta t} \int_0^{\Delta t} \tilde{\sigma}^2 \delta(t - \tau) dt d\tau = \Delta t \tilde{\sigma}^2 \\
v_k &\sim \mathcal{N}(\underline{0}, \tilde{\sigma}^2 \delta_{kl})
\end{aligned}$$

## 4.6 Pseudo-kode for banegenerator

Algorithm 1: Pseudo-kode for banegenerator	
1	<b>Initialverdier og parametere</b>
2	
3	$r = 5\text{m}; T = 30\text{s}; g = 9.81 \frac{\text{m}}{\text{s}^2}; N_z = 3001;$
4	$\sigma_{ba} = ?; \tilde{\sigma}_{va} = ?; \sigma_{bg} = ?; \tilde{\sigma}_{vg} = ?; \sigma_w = ?;$
5	$p_{bc}^b = [0; 0; 1]; \underline{\theta}_c^b = [0; \frac{\pi}{2}; 0]; N_p = 2; \underline{r}_1^n = [1; 1; 3]; \underline{r}_2^n = [1; -1; 3];$
6	$\omega = \frac{\pi}{T}; \Delta_t = \frac{T}{N_z}$
7	$\underline{g}^n = [0; 0; g]$
8	$\sigma_{va} = \sqrt{\Delta_t} \tilde{\sigma}_{va}$ <span style="float: right;"><math>\triangleright v_k^a \sim \mathcal{N}(\underline{0}, \Delta_t \tilde{\sigma}_{va}^2 I \delta_{kl})</math></span>
9	$\sigma_{vg} = \sqrt{\Delta_t} \tilde{\sigma}_{vg}$ <span style="float: right;"><math>\triangleright v_k^a \sim \mathcal{N}(\underline{0}, \Delta_t \tilde{\sigma}_{vg}^2 I \delta_{kl})</math></span>
10	$R_c^b = R_3(\theta_c^b(3))R_2(\theta_c^b(2))R_1(\theta_c^b(1))$
11	
12	<b>Sann bane og kameramålinger</b>
13	
14	<b>for</b> $k = 0 : N_z$ <b>do</b>
15	$\theta = k\Delta_t\omega; \underline{\theta}_k = [0; 0; \theta];$
16	$\underline{\omega}_b^n n(k) = [0; 0; \omega];$
17	$R_n^b = (R_b^n)^{-1}$
18	$\underline{p}_k^n = [r\cos(\theta); r\sin(\theta); 1];$
19	$\underline{v}_k^n = [-r\omega\sin(\theta); r\omega\cos(\theta); 0];$
20	$\underline{a}_k^n = [-r\omega^2\cos(\theta); -r\omega^2\sin(\theta); 0];$
21	$\mathbf{X}(:, k+1) = [\underline{p}_k^n; \underline{v}_k^n; \underline{\theta}_k];$
22	$\underline{f}_k^b = R_n^b(\theta_k)(\underline{a}_k^n + \underline{g}^n)$
23	$F(:, k+1) = \underline{f}_k^b$
24	$\underline{\omega}_b^{nb}(k) = R_n^b(\theta_k)\underline{\omega}_b^{nn}(k)$
25	$\Omega(:, k+1) = \underline{\omega}_b^{nb}(k)$
26	<b>end</b>
27	
28	<b>Måling for estimatoren med støy</b>
29	
30	<b>if</b> $k > 0$ <b>then</b>
31	<b>for</b> $i=1, n_p$ <b>do</b>
32	$\underline{\rho}_i^c(k) = R_b^c R_n^b(\theta_k) \left( \underline{r}_i^n(k) - \underline{p}_k^n \right) - R_b^c \underline{p}_{bc}^b$
33	$\underline{s}_k^i = [\underline{\rho}_{i,1}^c(k); \underline{\rho}_{i,2}^c(k)] \frac{1}{\underline{\rho}_{i,3}^c(k)};$
34	$\mathbf{Z}(:, k) = [\underline{s}_k^1; \dots; \underline{s}_k^{n_p}] + \sigma_w I_{n_p \times n_p} \text{randn}(2n_p);$
35	<b>end</b>
36	<b>end</b>
37	$\triangleright$ fortsetter på neste side

```

1 Adderer støy til pådrag for estimatoren
2
3 trekk  $\underline{b}^a \sim \mathcal{N}(\underline{0}, \sigma_{b^a}^2 I)$ ;  $\underline{b}^g \sim \mathcal{N}(\underline{0}, \sigma_{b^g}^2 I)$ 
4
5 for  $k = 0 : N_z - 1$  do
6    $\underline{u}_k^a = \frac{F(:,k) + F(:,k+1)}{2} - \underline{b}^a - \sqrt{\Delta_t} \tilde{\sigma}_{v^a} \mathbf{I} \mathbf{randn}(3)$ 
7    $\underline{u}_k^g = \frac{\Omega(:,k) + \Omega(:,k+1)}{2} - \underline{b}^g - \sqrt{\Delta_t} \tilde{\sigma}_{v^g} \mathbf{I} \mathbf{randn}(3)$ 
8    $\mathbf{U}(:, k) = [\underline{u}_k^a; \underline{u}_k^g]$ ;
9 end

```

## 4.7 Initialverdier for støy til filtermodellen

I dette avsnittet beskrives verdiene til støymodellen som implementeres i Matlab. Disse er blitt hentet fra datasettet til MTi xx-serien som er et produkt til selskapet Xsense [24].

Prosessstøyen har et PSD for systemet som er definert som

$$\tilde{Q} = \begin{bmatrix} \tilde{Q}_a & \\ & \tilde{Q}_g \end{bmatrix} \approx \begin{bmatrix} 2.34 \cdot 10^{-4} I & \\ & 0.75 I \end{bmatrix}$$

hvor  $Q$  er kovariansmatrisen til støyen og hvor PSD for akselerometer og gyroskopet kan utledes på denne måten

$$\tilde{Q}_a = \left( \frac{1}{g} 10^{-6} D_a \right)^2$$

$$\tilde{Q}_g = \left( \frac{2\pi}{360} D_g \right)^2$$

$D_a = 150 \frac{g}{\sqrt{\text{Hz}}}$  og  $D_g = 0.05 \frac{\text{grader}}{\sqrt{\text{Hz}}}$  er verdiene for støytettheten for akselerometer og gyroskopet.

Standardavviket for diskret hvit støy kan ble regnet ut på denne måten

$$\underline{\sigma}_{v^a} = \sqrt{\Delta_t} \cdot \tilde{Q}_a \approx \sqrt{\Delta_t} \cdot 2.34 \cdot 10^{-4}$$

$$\underline{\sigma}_{v^g} = \sqrt{\Delta_t} \cdot \tilde{Q}_g \approx \sqrt{\Delta_t} \cdot 0.75$$



Standardavviket for bias er gitt som

$$\begin{aligned}\sigma_{b_a} &= 0.03 \\ \sigma_{b_g} &= 0.2 \frac{2\pi}{360}\end{aligned}$$

hvor  $\Delta_t$  er samplings intervallet.

## 4.8 Filtermodell på diskretform

Likningene til ikke-lineære systemer må over på standardform for at vi kan bruke estimeringsmetoden for tilstander.

$$\begin{aligned}\underline{x}_{k+1} &= \underline{f}(\underline{x}_k, \underline{u}_k, \underline{v}_k) \\ \underline{z}_k &= \underline{h}(\underline{x}_k, \underline{w}_k)\end{aligned}$$

### 4.8.1 Utledning av filtermodellen fra integrallikningene

Vi ser på integrallikning settet 151 og bruker det som utgangspunkt

$$\begin{aligned}\underline{p}^n(t) &= \underline{p}^n(t_0) + \int_{t_0}^t \underline{v}^n(\tau) d\tau \\ \underline{v}^n(t) &= \underline{v}^n(t_0) + \int_{t_0}^t R_b^n(\underline{\theta}(\tau)) \underline{f}^b(\tau) d\tau - (t - t_0) \underline{g}^n \\ \underline{\theta}(t) &= \underline{\theta}(t_0) + \int_{t_0}^t D_b^\theta(\underline{\theta}(\tau)) \underline{\omega}_b^{nb}(\tau) d\tau\end{aligned}\tag{165}$$

$\underline{p}^n(t_0)$ ,  $\underline{v}^n(t_0)$  og  $\underline{\theta}(t_0)$  er gitt

og videre ser på likningssettet 152 for å lage et sammenheng mellom sanne og målte akselerometer- og gyromålinger.

Vi ser på tidsintervallet  $[t_k, t_{k+1}] = [k\Delta_t, (k+1)\Delta_t]$ :

$$\begin{aligned}\underline{f}^b(t) &= \underline{\tilde{f}}^b(t) + \underline{b}^a + \underline{v}^a(t) \\ \underline{\omega}_b^{nb}(t) &= \underline{\tilde{\omega}}_b^{nb}(t) + \underline{b}^g + \underline{v}^g(t) \\ \underline{p}_{k+1}^n &= \underline{p}_k^n + \int_{t_k}^{t_{k+1}} \underline{v}^n(\tau) d\tau \\ \underline{v}_{k+1}^n &= \underline{v}_k^n + \int_{t_k}^{t_{k+1}} R_b^n(\underline{\theta}(\tau)) (\underline{\tilde{f}}^b(\tau) + \underline{b}^a + \underline{v}^a(\tau)) d\tau - \Delta_t \underline{g}^n \\ \underline{\theta}_{k+1}^n &= \underline{\theta}_k^n + \int_{t_k}^{t_{k+1}} D_b^\theta(\underline{\theta}(\tau)) (\underline{\tilde{\omega}}_b^{nb}(\tau) + \underline{b}^g + \underline{v}^g(\tau)) d\tau\end{aligned}\tag{166}$$

Hvis vi nå antar at alle vektorer og matriser er konstante i intervallet og ser bort fra hvitstøyen, vil vi forkorte intervallet og få:

$$\begin{aligned}
\underline{p}_{k+1}^n &= \underline{p}_k^n + \Delta_t \underline{v}_k^n \\
\underline{v}_{k+1}^n &= \underline{v}_k^n + \Delta_t R_b^n(\underline{\theta}_k)(\underline{f}_k^b + \underline{b}^a) - \Delta_t \underline{g}^n + R_b^n(\underline{\theta}_k) \int_{t_k}^{t_{k+1}} \underline{v}^a(\tau) d\tau \\
\underline{\theta}_{k+1}^n &= \underline{\theta}_k^n + \Delta_t D_b^\theta(\underline{\theta}_k)(\tilde{\omega}_b^{nb}(k) + \underline{b}^g) + D_b^\theta(\underline{\theta}_k) \int_{t_k}^{t_{k+1}} \underline{v}^g(\tau) d\tau \\
\int_{t_k}^{t_{k+1}} \underline{v}^a(\tau) d\tau &= \sqrt{\Delta_t} \tilde{\sigma}_{v^a} \underline{n}(3) \sim \mathcal{N}(\underline{0}, \Delta_t \tilde{\sigma}_{v^a}^2 \delta_{kl}) \\
\int_{t_k}^{t_{k+1}} \underline{v}^g(\tau) d\tau &= \sqrt{\Delta_t} \tilde{\sigma}_{v^g} \underline{n}(3) \sim \mathcal{N}(\underline{0}, \Delta_t \tilde{\sigma}_{v^g}^2 \delta_{kl})
\end{aligned} \tag{167}$$

Den diskrete filtermodellen vil da bli:

$$\begin{aligned}
\underline{p}_{k+1}^n &= \underline{p}_k^n + \Delta_t \underline{v}_k^n \\
\underline{v}_{k+1}^n &= \underline{v}_k^n + \Delta_t R_b^n(\underline{\theta}_k)(\underline{f}_k^b + \underline{b}_k^a) - \Delta_t \underline{g}^n + R_b^n(\underline{\theta}_k) \sqrt{\Delta_t} \tilde{\sigma}_{v^a} \underline{n}(3) \\
\underline{\theta}_{k+1} &= \underline{\theta}_k + \Delta_t D_b^\theta(\underline{\theta}_k)(\tilde{\omega}_b^{nb}(k) + \underline{b}_k^g) + D_b^\theta(\underline{\theta}_k) \sqrt{\Delta_t} \tilde{\sigma}_{v^g} \underline{n}(3) \\
\underline{b}_{k+1}^a &= \underline{b}_k^a \\
\underline{b}_{k+1}^g &= \underline{b}_k^g \\
\underline{x}_k &= [\underline{p}_k^n; \underline{v}_k^n; \underline{\theta}_k; \underline{b}_k^a; \underline{b}_k^g]; \underline{u}_k = [\underline{f}_k^b; \tilde{\omega}_b^{nb}(k)]; \underline{v}_k = [\underline{v}_k^a; \underline{v}_k^g] \\
\hat{P}_0 &= \mathbf{diag}([\hat{P}_0^p; \hat{P}_0^v; \hat{P}_0^\theta; \sigma_{b^a}^2 I; \sigma_{b^g}^2 I]) \\
\underline{x}_0 &\sim \mathcal{N}(\hat{x}_0, \hat{P}_0); \underline{n}(3) \sim \mathcal{N}(\underline{0}, I)
\end{aligned}$$

## 4.9 Diskretisering av ikke-lineær stokastiske prosess - del 2

I del 1 for dette kapitlet (se kapittel 3.2.6) forklarte vi hvordan det går an å diskretisere en stokastisk system som er ikke-lineær på den generelle formen. I dette kapitlet skal vi diskretisere vårt system.

$$\underline{f} \left( \begin{array}{c} \left[ \begin{array}{c} \underline{p}_k^n \\ \underline{v}_k^n \\ \underline{\theta}_k \\ \underline{b}_k^a \\ \underline{b}_k^g \end{array} \right], \left[ \begin{array}{c} \underline{\tilde{f}}_k^b \\ \underline{\tilde{\omega}}_b^{nb}(k) \end{array} \right], \underline{0} \end{array} \right) = \begin{array}{c} \underline{v}_k^n \\ R_b^n(\underline{\theta}_k)(\underline{\tilde{f}}_k^b + \underline{b}_k^a) - \underline{g}^n \\ D_b^\theta(\underline{\theta}_k)(\underline{\tilde{\omega}}_b^{nb}(k) + \underline{b}_k^g) \\ \underline{0} \\ \underline{0} \end{array} \quad (168)$$

$$G(\underline{x}^d, \underline{u}) = \frac{\partial}{\partial \underline{v}^T} \underline{f}(\underline{x}, \underline{u}, \underline{v}) \Big|_{\underline{x}^d, \underline{u}, \underline{0}} = \begin{array}{c} \left[ \begin{array}{ccc} \underline{0} & \underline{0} & \underline{0} \\ R_b^n(\underline{\theta}_k^d) & \underline{0} & \underline{0} \\ \underline{0} & D_b^\theta(\underline{\theta}_k^d) & \underline{0} \\ \underline{0} & \underline{0} & I \\ \underline{0} & \underline{0} & \underline{0} \\ \underline{0} & \underline{0} & \underline{0} \end{array} \right] \end{array} \quad (169)$$

og den diskrete modellen blir

$$\begin{aligned} \underline{p}_{k+1} &= \underline{p}_k + \Delta_t \underline{v}_k^n \\ \underline{v}_{k+1} &= \underline{v}_k + \Delta_t R_b^n(\underline{\theta}_k)(\underline{\tilde{f}}_k^b + \underline{b}_k^a) - \underline{g}^n + R_b^n(\underline{\theta}_k^d) \underline{v}_k^a \\ \underline{\theta}_{k+1} &= \underline{\theta}_k + \Delta_t \left( D_b^\theta(\underline{\theta}_k)(\underline{\tilde{\omega}}_b^{nb}(k) + \underline{b}_k^g) \right) + D_b^\theta(\underline{\theta}_k^d) \underline{v}_k^g \\ \underline{b}_{k+1}^a &= \underline{b}_k^a \\ \underline{b}_{k+1}^g &= \underline{b}_k^g \end{aligned} \quad (170)$$

## 5 Resultater og diskusjon

I dette kapitlet vil vi se på resultatene vi har kommet frem til. Alle resultatene er hentet fra Matlab koden som følger med rapporten som vedlegg A.

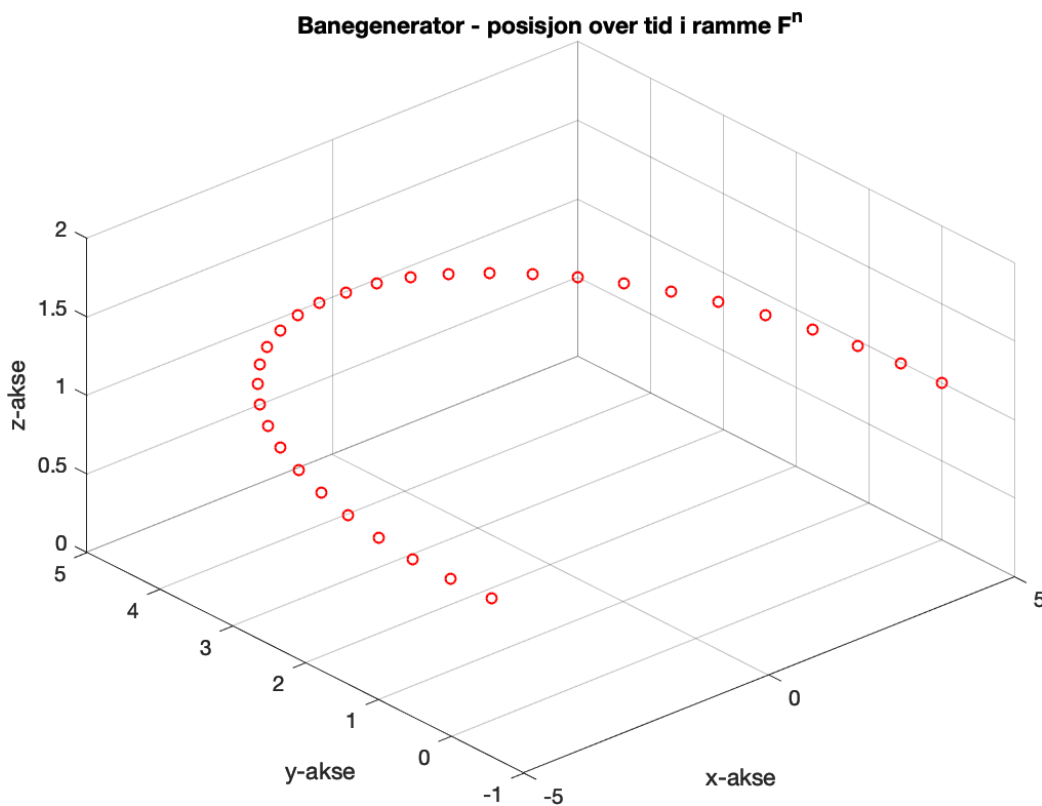
Siden oppgaven har hatt sine utfordringer, er det blitt diskutert på slutten av dette kapitlet hva som kunne blitt gjort annerledes og hva som ikke ble gjort, men var tiltenkt.

Figur 48 gir en fremstilling på hvordan mikrofonmatrisen beveger seg rundt en boogie. Med massesenteret til boogien i origo har vi valgt at høyden på matrisen i rommet er på en meter og banen har en radius på 5 meter.

### 5.1 De deterministiske testene

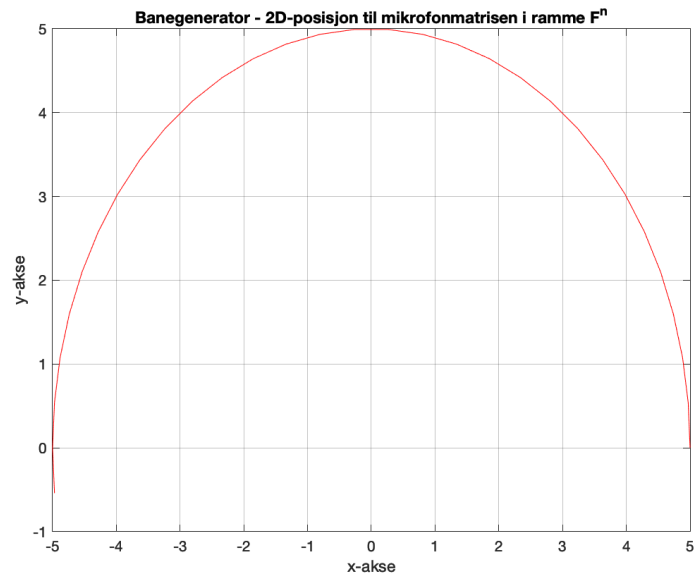
#### 5.1.1 Banegenerator:

Den analytiske posisjon for banegeneratoren presentert i 3D



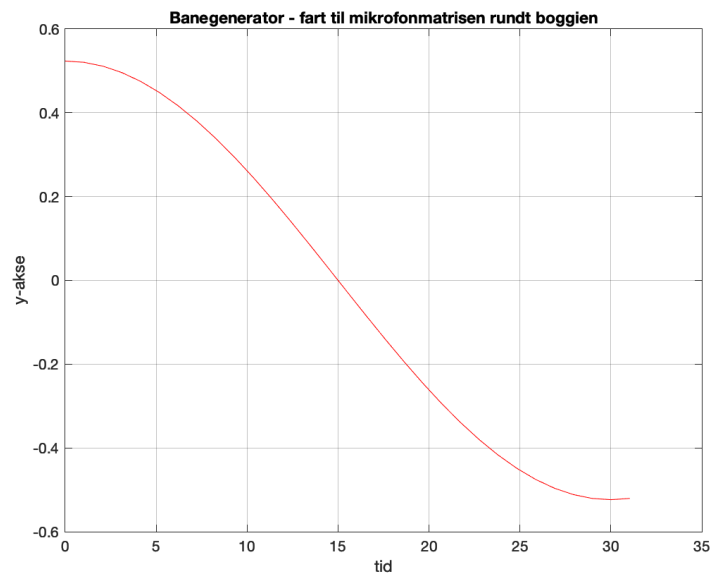
Figur 48: Posisjon i 3D for banegenerator

Den analytiske posisjon for banegeneratoren presentert i 2D



Figur 49: Posisjon i 2D for banegenerator

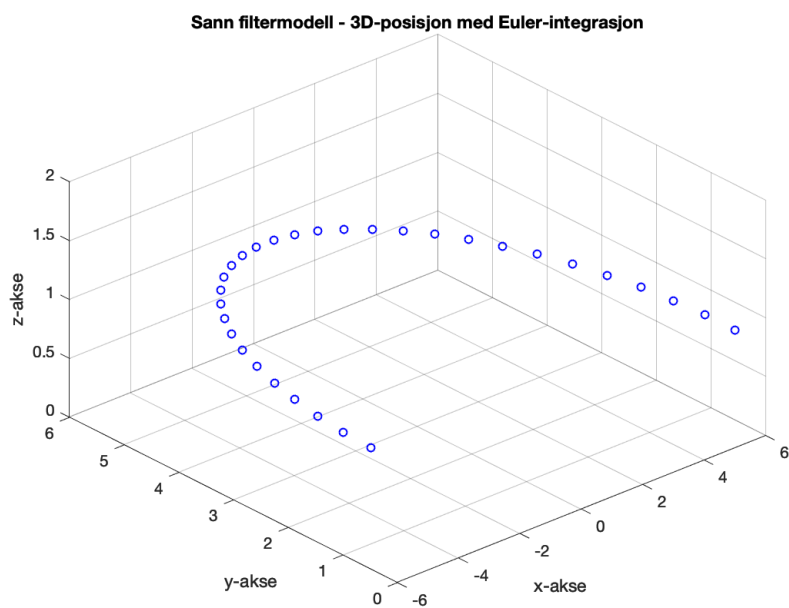
Den analytiske farten for mikrofonmatrisen rundt boogien over tid



Figur 50: Fart for mikrofonmatrise presentert for banegenerator

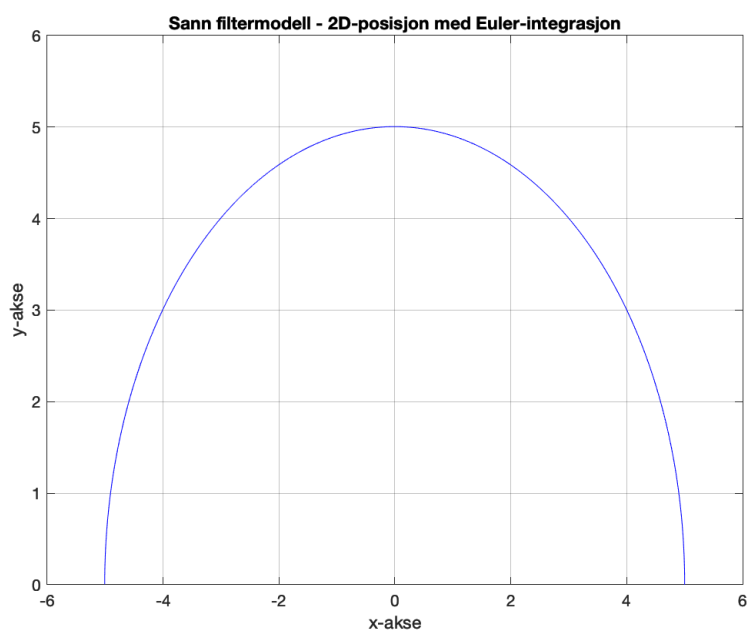
## 5.2 Sann filtermodell:

Sirkelbane i 3D for sann filtermodell



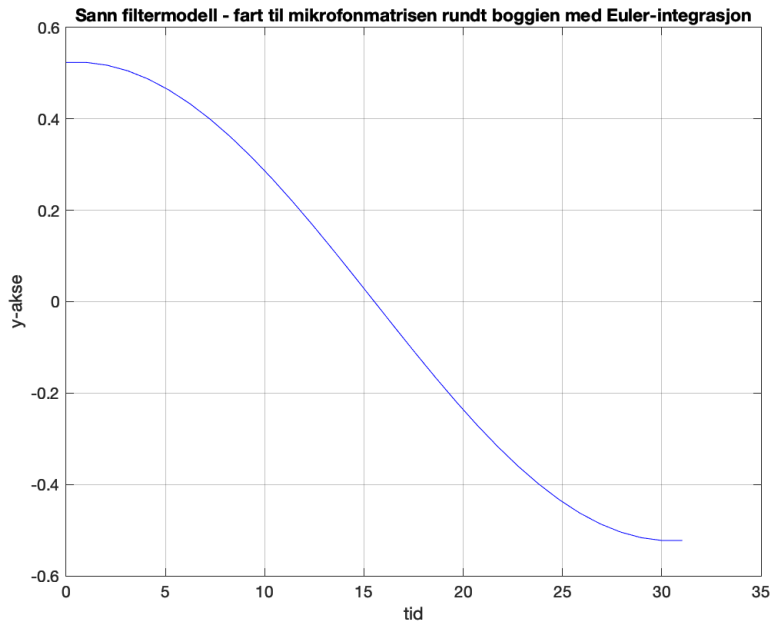
Figur 51: Posisjon i 3D for sann filtermodell

Sirkelbane i 2D for sann filtermodell



Figur 52: Posisjon i 2D for sann filtermodell

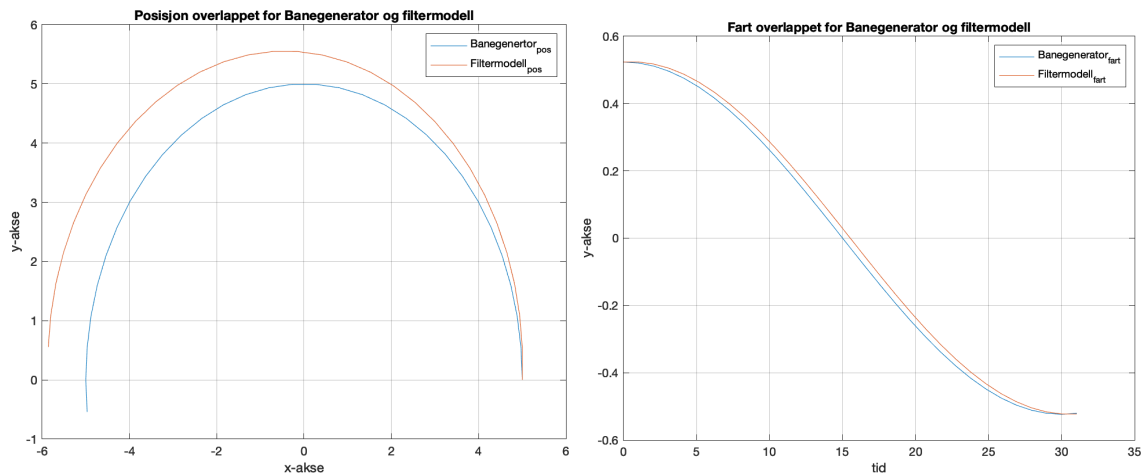
Farten for sanne filtermodellen.



Figur 53: Fart for mikrofonmatrise presentert for sann filtermodell

### 5.2.1 Sammenheng mellom banegenerator og sanne filtermodell:

Overlappet posisjon og fart av begge modellene for 30 iterasjoner med Euler-integrasjon

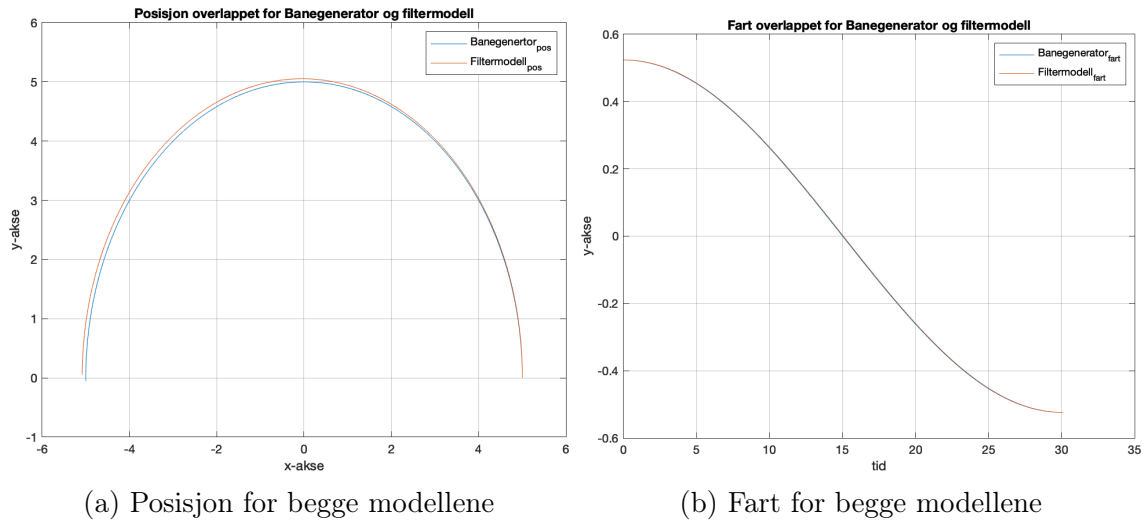


(a) Posisjon for begge modellene

(b) Fart for begge modellene

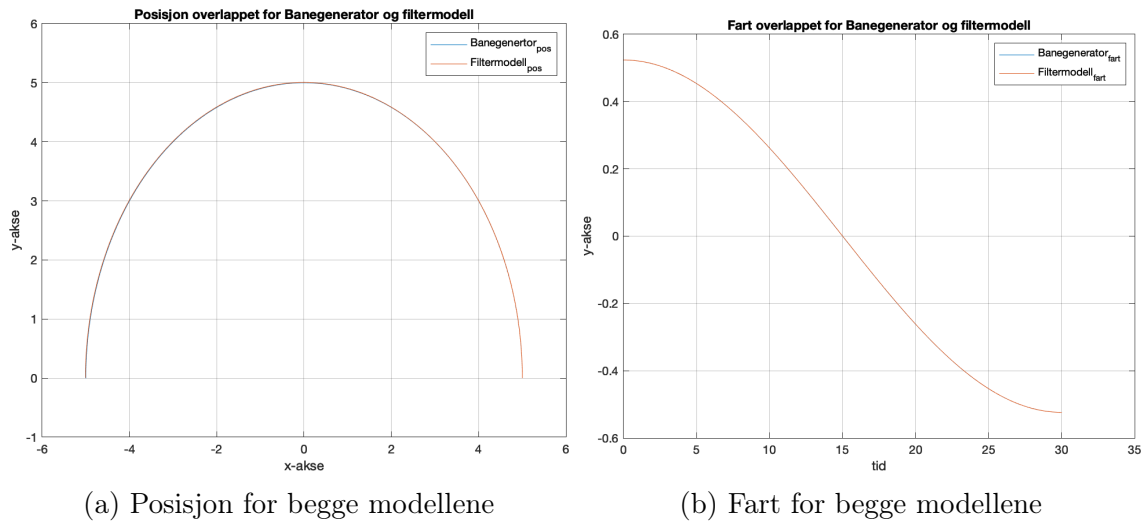
Figur 54: Begge banene overlappet for 30 sampler

Overlappet posisjon og fart av begge modellene for 300 iterasjoner med Euler-integrasjon



Figur 55: Begge banene overlappet for 300 sampler

Overlappet posisjon og fart av begge modellene for 3000 iterasjoner med Euler-integrasjon



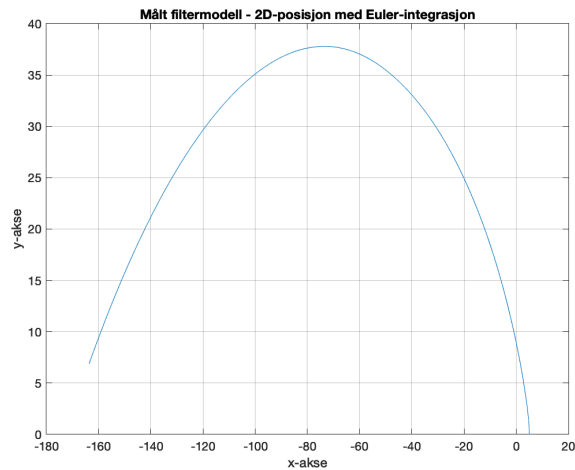
Figur 56: Begge banene overlappet for 3000 sampler



## 5.3 De stokastiske testene

### 5.3.1 Målte filtermodellen

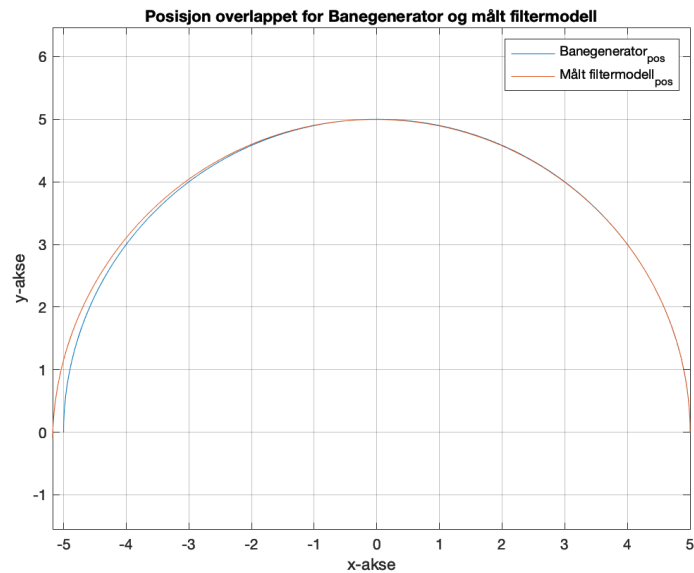
Sirkelbane for målt filtermodell med 3000 iterasjoner og omløpstid på 30 sek:



Figur 57: Posisjon for den målte filtermodellen for omløpstid med 30 sek

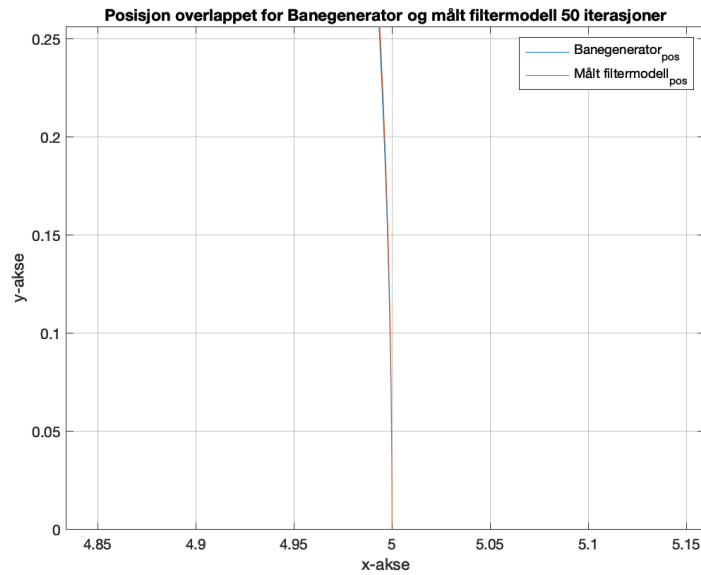
### 5.3.2 Sammenheng mellom banegenerator og den målte filtermodellen:

Banegenerator og den målte filtermodellen overlappet for 3000 iterasjoner og omløpstid på 3 sek:



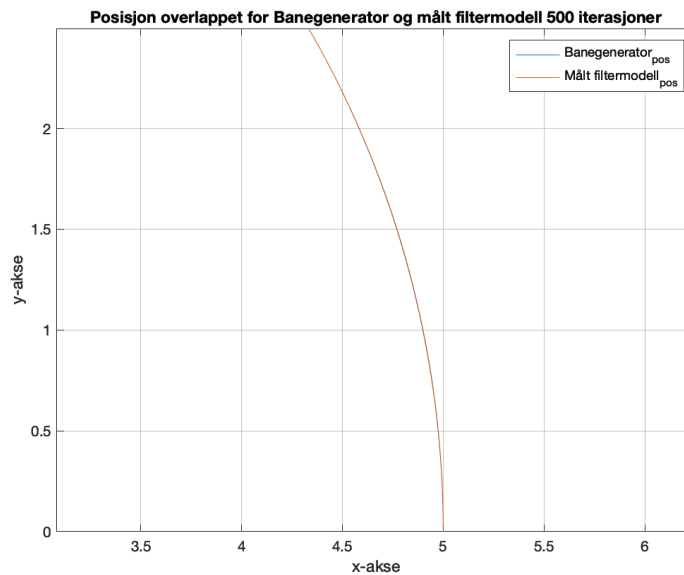
Figur 58: Banegenerator og målt filtermodell overlappet med omløpstid på 3 sek

Banegenerator og den målte filtermodellen overlappet for 50 første iterasjoner og omløpstid på 3 sek:



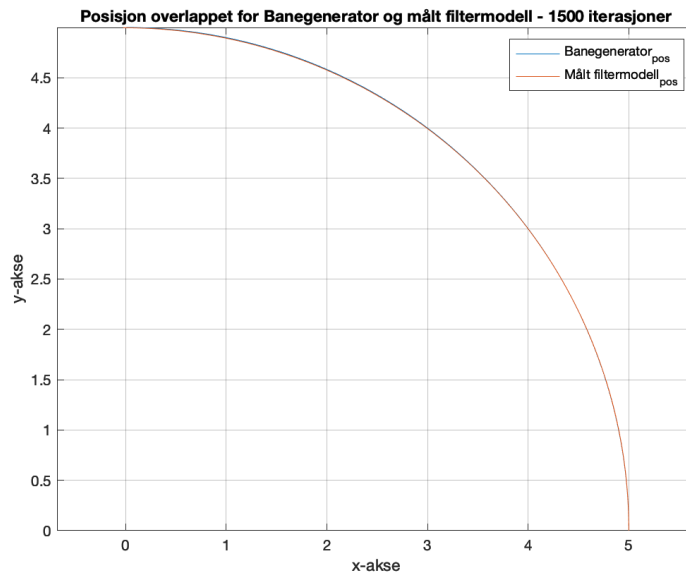
Figur 59: Banegenerator og målt filtermodell overlappet for 50 iterasjoner med omløpstid på 3 sek

Banegenerator og den målte filtermodellen overlappet for 500 første iterasjoner og omløpstid på 3 sek:



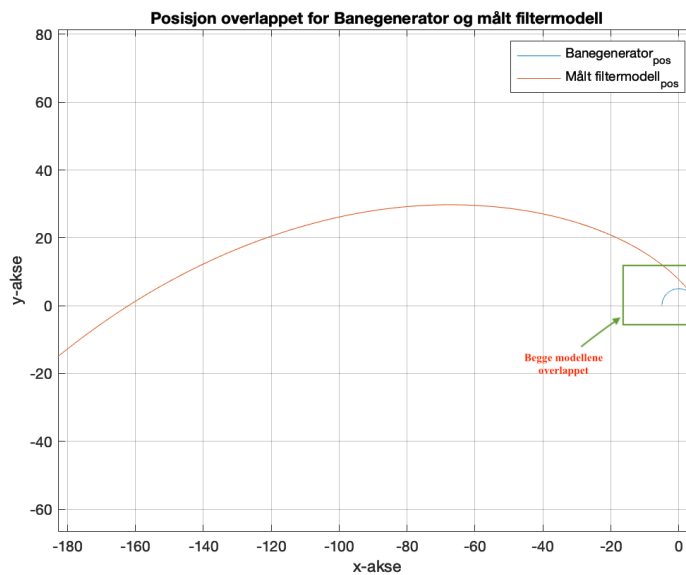
Figur 60: Banegenerator og målt filtermodell overlappet for 500 iterasjoner med omløpstid på 3 sek

Banegenerator og den målte filtermodellen overlappet for 1500 iterasjoner og omløpstid på 3 sek:



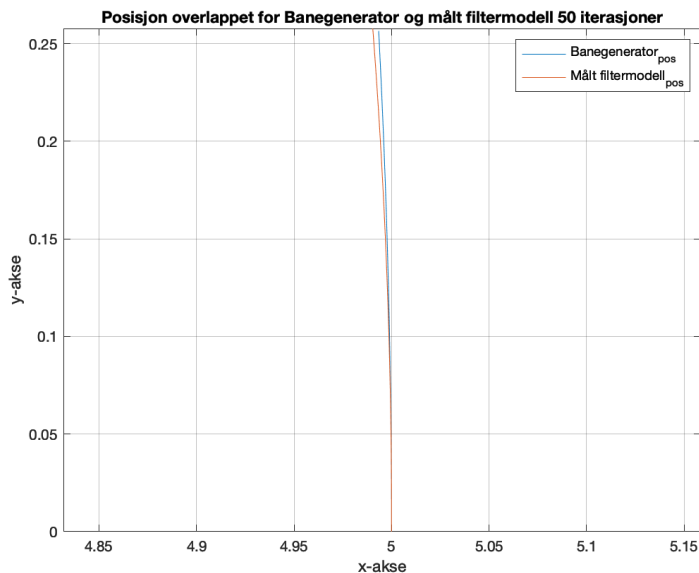
Figur 61: Banegenerator og målt filtermodell overlappet for 1500 iterasjoner med omløpstid på 3 sek

Banegenerator og den målte filtermodellen overlappet for 3000 iterasjoner og omløpstid på 30 sek:



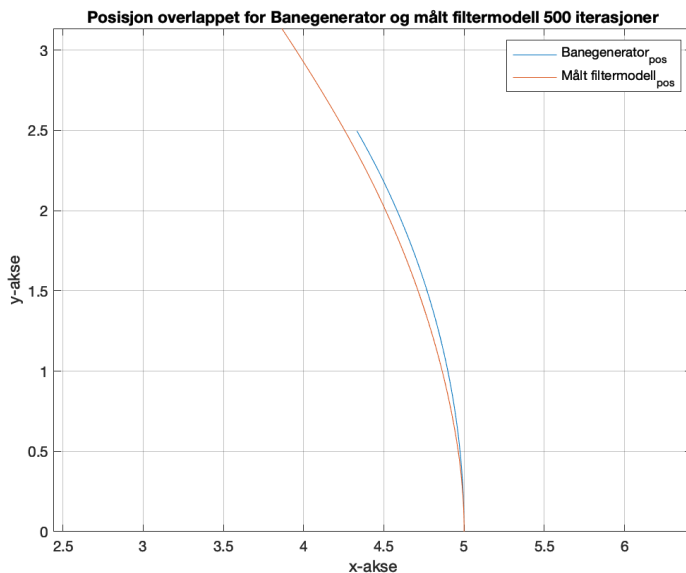
Figur 62: Banegenerator og målt filtermodell overlappet med omløpstid på 30 sek

Banegenerator og den målte filtermodellen overlappet for 50 første iterasjoner og omløpstid på 30 sek:



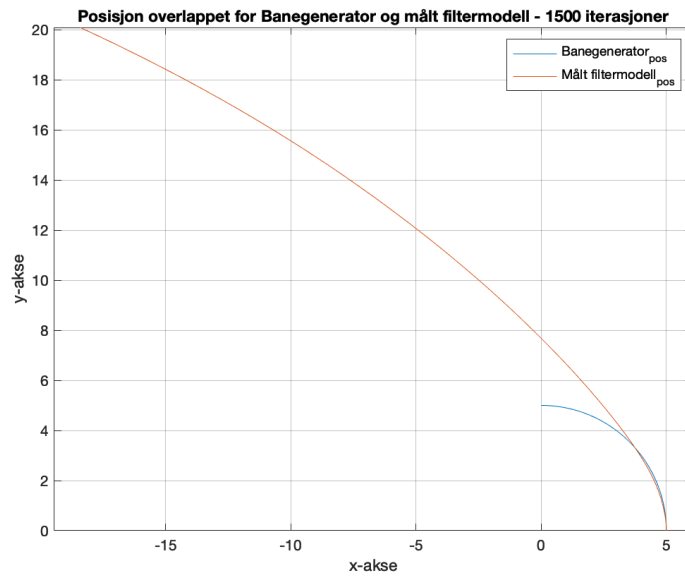
Figur 63: Banegenerator og målt filtermodell overlappet for 50 iterasjoner med omløpstid på 30 sek

Banegenerator og den målte filtermodellen overlappet for 500 første iterasjoner og omløpstid på 30 sek:



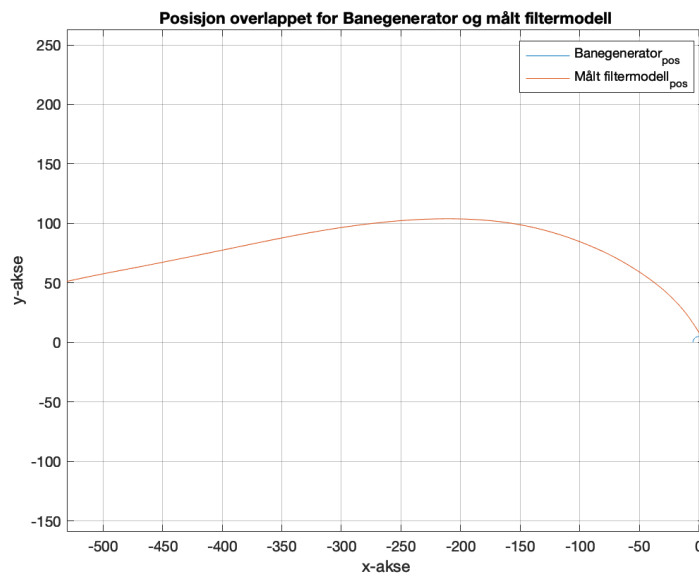
Figur 64: Banegenerator og målt filtermodell overlappet for 500 iterasjoner med omløpstid på 30 sek

Banegenerator og den målte filtermodellen overlappet for 1500 iterasjoner og omløpstid på 30 sek:



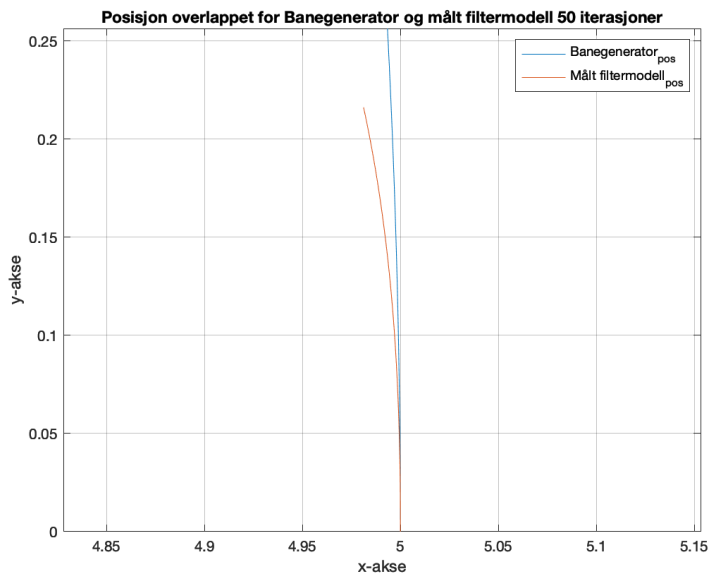
Figur 65: Banegenerator og målt filtermodell overlappet for 1500 iterasjoner med omløpstid på 30 sek

Banegenerator og den målte filtermodellen overlappet for 3000 iterasjoner og omløpstid på 60 sek:



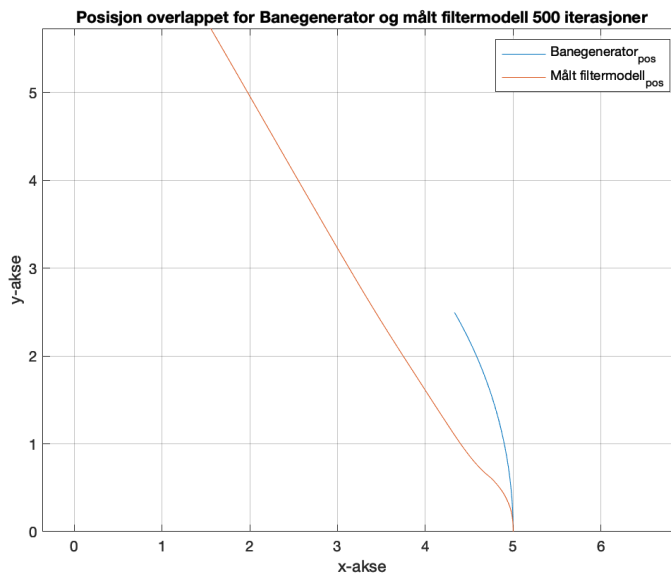
Figur 66: Banegenerator og målt filtermodell overlappet med omløpstid på 60 sek

Banegenerator og den målte filtermodellen overlappet for 50 første iterasjoner og omløpstid på 60 sek:



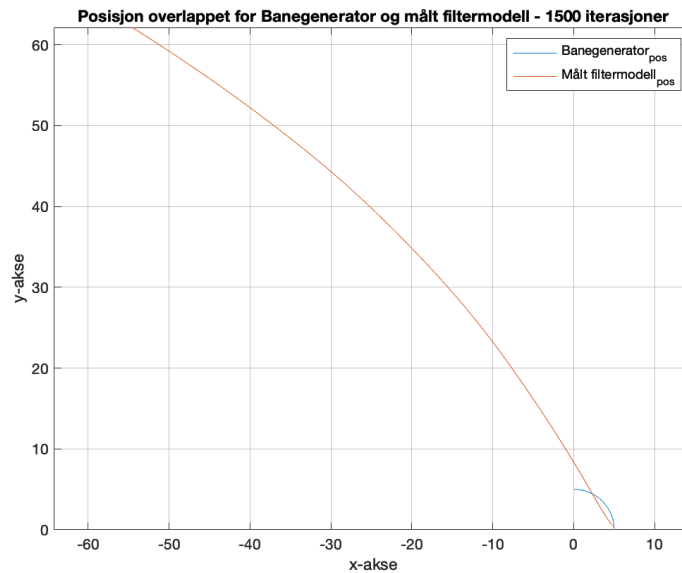
Figur 67: Banegenerator og målt filtermodell overlappet for 50 iterasjoner med omløpstid på 60 sek

Banegenerator og den målte filtermodellen overlappet for 500 første iterasjoner og omløpstid på 60 sek:



Figur 68: Banegenerator og målt filtermodell overlappet for 500 iterasjoner med omløpstid på 60 sek

Banegenerator og den målte filtermodellen overlappet for 1500 iterasjoner og omløpstid på 60 sek:



Figur 69: Banegenerator og målt filtermodell overlappet for 1500 iterasjoner med omløpstid på 60 sek

Nedenfor er det et utkast av Matlab-koden for punkter projisert i bildeplanet. Dette er data vi skal bruke som sensormålinger for vår estimator (resten av koden finnes i vedlegg A):

```

%% Sensorverdier(utgangsverdier) hentet fra punkter i verden
for k = 1 : n_s
    rho_1(:,k) = R_c_b'*R_b_n'*(r_1-P(:,k)) - R_c_b*p_bc_b;
    rho_2(:,k) = R_c_b'*R_b_n'*(r_2-P(:,k)) - R_c_b*p_bc_b;
    % x-koordinaten i bildeplaneet for punkt 1
    s_x_1 = rho_1(1,k);
    % y-koordinaten i bildeplaneet
    s_y_1 = rho_1(2,k);
    % z-koordinaten i bildeplaneet
    z_1 = rho_1(3,k);
    % x-koordinaten i bildeplaneet for punkt 2
    s_x_2 = rho_2(1,k);
    % y-koordinaten i bildeplaneet
    s_y_2 = rho_2(2,k);
    % z-koordinaten i bildeplaneet
    z_2 = rho_2(3,k);
    Z(1:3,k) = [s_x_1 s_y_1 z_1]/z_1'; % plasserer første punktet hentet fra bildeplanet i kamera
    Z(4:6,k) = [s_x_2 s_y_2 z_2]/z_2'; % plasserer andre punktet hentet fra bildeplanet i kamera
end

```

Figur 70: Matlab kode for sensormålinger

## 5.4 Drøfting og diskusjon av resultatene

Det første vi skal diskutere vedrørende dette kapittelet er resultatene vi har fått fra simuleringen av banegeneratoren og sann filtermodell. Deretter vil vi diskutere hvordan den målte filtermodellen fungerer. Det var ønskelig å ha et mye mer utvidet kapittel med resultater. I den utvidede delen ønsket vi å redegjøre for verdiene fra filtermodellen satt inn i en estimator som UKF. Den målte filtermodellen er definert med diverse støyer som gir oss et ikke-lineært system. Det skulle være vårt utgangspunkt for å skaffe en optimal bane med UKF som var tilnærmet lik den analytiske banen. Det lot seg ikke gjøre fordi tiden som var til rådighet ikke strakk til. Tidsbegrensningen var forårsaket av forandringen av oppgaven midtveis i masteroppgaven og en begrenset kunnskap om den nye oppgaven. Mye av den resterende tiden som var igjen gikk på å gjøre nye undersøkelser for den nye problemstillingen. Det vi har kommet frem til nå er tilstrekkelig for å bygge videre på en endelig løsningen av problemstillingen.

Vi kan se fra figur 48 og 49 at det er mulig å fremstille bevegelse av mikrofonmatrisen rundt en boogie på en analytisk måte. Denne banen og bevegelsen rundt denne banen er den ideelle fremstillingen. Det er slik vi ønsker at mikrofonmatrisen skal bevege seg rundt boogien. Vi kan også se fra figur 50 at farten som blir fremstilt stemmer med den matematiske modellen for banegeneratoren. Som nevnt tidligere i oppgaven vil vi få verdiene for posisjon, fart og akselerasjon fra banegeneratoren ved å derivere likninger for en sirkel. Fra disse verdiene får vi en spesifikk kraft og en vinkelhastighet. Vi bruker deretter spesifikk kraften og vinkelhastigheten til å lage en filtermodell. Til å begynne med er det ikke noe støy på filtermodellen og vi kaller den for sann filtermodell. Det er ønskelig at de verdiene vi får fra den sanne filtermodellen skal også samsvare med den analytiske løsningen. I motsetning til banegeneratoren integrer vi den spesifikke kraften for akselerometeret og vinkelhastigheten for gyroen slik at vi kan få posisjon og fart. Dette har vi vist at vi har oppnådd som vist med figur 51, 57 og 53. Det er viktig å påpeke at den sanne filtermodellen har et betraktelig avvik fra den analytiske løsningen når vi Euler-integrer med få iterasjoner. Dette er vist med en sammenligning av begge modellene som på figur 54 hvor vi har bare integrert farten og posisjonen med bare 30 iterasjoner. Hvis vi ser videre på filtermodellen i figur 55, ser vi at avviket er blitt mye mindre. Og som vist med figur 56 der det har blitt integrert 3000 ganger er begge modellene sammenfallende. Det forteller oss at for den simpleste banen og løsningen vil Euler-integrasjon være tilstrekkelig til å skaffe måle data for en filtermodell som skal brukes i en estimator. Det er også viktig å poengtere at disse målingene er for en sann filtermodell. Det vil si at dette er den ideelle løsningen. Den ideelle løsningen er alltid tilnærmet feilfri, derav navnet ideell.

Hvis vi nå skal bruke den målte filtermodellen vil det uten tvil være mye mer ulineære verdier. Dette er forårsaket av at vi har støy i form av bias og hvit støy som påvirker akselerometeret og gyroen. Gyroen vil i tillegg ha en farget støy som vi bevisst har utelatt i vår filtermodell. Figur 57 er en fremstilling av hvordan banen utfolder seg når prosessen er stokastisk og det påføres støy på filtermodellen. Hvis man ser på omløpstid for 3 sekunder, er det tydelig at forskjellen mellom den analytiske og den målte modellen som vist med 58 er liten. Det kommer også klart frem i figur 59, 60 og 61 at det er små avvik for hver iterasjon som utføres for en Euler-integrasjon.

Hvis man ser på omløpstid rundt banen ved et 30 sekunders intervall, kommer dette avviket tydeligere fram. Dette er vist i figur 62, og her er det mulig å se den analytiske løsningen og den målte filtermodellen i en og samme figur og avviket i mellom dem. Vi kan se at begge modellene starter likt, men for hver gang vi Euler-integrerer øker avviket mellom den målte og den analytiske pro-



sessen. Til slutt har støyen såpass stor påvirkning at den målte filtermodellen får et stort avvik fra den banen som er ønskelig. Den mer utvidede fremstillingen på hvordan differansen mellom begge modellene utvikler seg er vist i figur 63, 64 og 65. Det figur 63 og 64 viser oss er at støyen som påvirker gyroen og akselerometer ikke har mye påvirkning tidlig i prosessen. Derfor vil den målte filtermodellen følge den ideelle banen, men i figur 65 kan vi se at differansen øker. Desto lengre tid det tar rundt banen desto raskere vil avviket forekomme mellom disse to modellene. Dette er noe som vises tydelig i figur 67, 68 og 69 når omløpstiden settes til 60 sekunder.

Konklusjonen på dette for vår del er at for hvert sample over lengre tid som hentes fra Euler-integrasjon øker avviket fra gyroen og akselerometeret. Dette skjer mye raskere ved større omløpstid og vil føre til at systemet går fortere ut av den ideelle banen. Med disse beregningene kan vi fastslå at vår filtermodell fungerer som ønskelig, men siden vi ikke har mulighet til å bruke disse verdiene i en estimator, er det ønskelig at videre arbeid med denne problemstillingen innebærer å bruke verdiene fra filtermodellen funnet i denne masteroppgaven i et UKF.

Videre har vi sett på målinger fra sensoren. Sensoren i vårt tilfelle er kameraet. Målinger fra sensoren får vi fra faste punkter i rommet. Denne delen av oppgaven er blitt gjort ferdig i Matlab programmet som vist med figur 70 og vedlegg A. Det vil ikke være mulig å simulere et resultat som gir mening før vi kan ta i bruk en estimator. Det som er greit å nevne er at siden kameraet er plassert på toppen av matrisen har vi brukt punkter i taket som et referansepunkt. Dette er gjort fordi synsvinkelen til kameraet vil være mye større og faren for å bevege seg ut av bildeplanet være mindre. For å etterprøve denne påstand vil det være riktig å prøve dette ut med en estimator sammen med resten av tilstandene. Til slutt må det også tas høyde for farget støy for gyroen. Den er blitt utelatt for å holde filtermodellen simpel slik at det var mulig å vise til resultater.

Til slutt vil jeg ta med litt om den delen av oppgaven som var i utgangspunktet for denne masteroppgaven. Jeg hadde som formål å finne ut om en mikrofonmatrise ville gjøre den samme jobben som en akselerometer og mye mer effektivt. For å gi en konkret svar på dette var det viktig å ha målinger fra begge løsningene for å sammenligne dem med hverandre. Det første hindret som oppstod for oppgaven var tilgjengelighet av data. Det fantes data fra akselerometeret som var koblet på boogiene, men ikke fra en mikrofonmatrise. Siden vi ikke hadde en mikrofonmatrise tilgjengelig var det heller ikke mulig å oppdrive disse dataene. Hvis man skulle få en riktig tilnærming til problemstillingen var det helt nødvendig med en mikrofonmatrise. Denne mikrofonmatrisen måtte lages selv eller spesialbestilles fra eventuelt en av selskapene nevnt i kapittel 2. Det er også veldig essensielt for valget av mikrofonmatrisen at hvilken størrelsen det skal være på mikrofonmatrisen og hvor langt avstanden til lyd-kilden skulle være. Utenom det måtte det tas en avgjørelse på hvilken type mikrofonmatrise som skulle brukes. Med denne informasjonen lagt til grunn ble det veldig tydelig at oppgaven som var gitt som en kort masteroppgave burde i virkeligheten vært en lang masteroppgave. Det ble også nevnt underveis i oppgaven om å lage et rent simuleringsprogram som kunne fremstille en mikrofonmatrise. Dette ville ha gitt oss en ideell fremstilling av problemet, men det vil ikke gitt oss svar på problemstilling. Dette fordi det var mange andre faktorer som måtte tas stilling til. Ting som måtte tatt med i betraktning var eksempelvis, hvor det var ønskelig å plassere mikrofonmatrisen i den Dynamiske Testbenken. Skulle det være en stasjonær løsning eller skulle det være mulig å bevege matrisen rundt i rommet som igjen ville føre til at tilnærmingen til oppgaven måtte bli annerledes. Det finnes veldig mange forskjellige boogier og hver av dem har sine unike karakteristika. Det vil være helt umulig å lage et ideell simuleringsprogram og forvente at det vil

gå an å bruke den som referanse for situasjoner fra virkeligheten, når det er alt for mange ukjente faktorer som må tas med i betraktning. Derfor vil min anbefaling til Mantena AS være først å få laget en mikrofonmatrise. Det vil også være riktig å bruke en mikrofonmatrise som har en form som et plan og peker direkte på boogien (detaljer i kapittel 2.3.4.2 og 2.3.5.5). Dette vil føre til at en del av støyen som er rundt matrisen bli utelatt og vil gi mye bedre resultater.

Fra mine undersøkelser gjort for denne oppgaven vedrørende bruk av mikrofonmatrise vil jeg påstå at en løsning med mikrofonmatrise vil være mye mer effektivt i forhold til en akselerometer. Pre idag vil et akselerometer kanskje være mer robust, men utviklingen på feltet mikrofonmatriser skjer veldig raskt. Store mengder med data blir prosessert mye raskere enn tidligere og hastigheter for overførsel av data i sanntid er raskt. Dette fører til at feilsøking av støykilder vil være veldig nøyaktige med mikrofonmatriser. I tillegg til alt dette finnes det avanserte matematiske algoritmer som kan brukes for å fjerne støykomponenter med presisjon. Med dagens teknologi er det blitt mye enklere og raskere å prosessere disse algoritmene.

## 6 Konklusjon

Konklusjonen av hele oppgaven vil basere seg på begge problemstillingene som det er blitt jobbet med. Det første vil være å se på løsningen med mikrofonmatrisen og bruken av den for å effektivisere test tiden det tar på en boogie. Deretter skal vi ta for oss TNS integrert med mikrofonmatrise.

For å vite med sikkerhet om bruken av en mikrofonmatrise løsning ville være den mest effektive og produktive, ville det beste ha vært å ha data som kunne sammenligne akselerometermålinger med mikrofonmatrisemålinger. Da ville svaret ha vært basert på faktaopplysninger.

Etter å ha gjort en grundig undersøkelse og dokumentert utfyllende i denne oppgaven vil jeg påstå at jeg har nok grunnlag for å hevde at bruken av mikrofonmatrise er en løsning som vil både effektivisere testtiden av boogien og øke produktiviteten for feilsøking av boogien. Med dagens teknologi på feltet for mikrofonmatriser er det mulig å finne de vanskeligste støykildene som påvirker målingene. Dette er blitt redegjort for i kapittel 2.3.4 og 2.3.5. Med dagens teknologi vil det ikke bare være mulig å finne de riktige støykildene, men også fortelle med nøyaktighet hvor kilden vil befinne seg på boogien. En annen stor fordel med bruk av mikrofonmatrise vil være å jobbe med data i sanntid. En mer avansert løsning vil være å fremstille støyen visuelt i tillegg til å få den presentert med lyd. På denne måten kan restaureringen av feilen bli mer effektivt. Dette vil åpne muligheter for å oppdage feil eller slitasje på boogien i en tidlig fase. På denne måten vil det være mulig å jobbes med skaden før den blir verre. Dette vil også føre til en mye mer effektivt feilsøkingssprosess og spare Mantena AS for potensielle økonomisk tap. Programmer som blir brukt med mikrofonmatriser for å bearbeide og analysere dataen holder veldig høy standard. Det vil definitivt gjøre det enklere og mye mer effektivt for vedkommende som skal analysere dataen til å finne raske svar.

Derfor vil min anbefaling basert på undersøkelser gjort for denne oppgaven være at Mantena AS bytter ut dagens løsning med en mikrofonmatrise.

TNS integrert med mikrofonmatrise er den andre delen av masteroppgaven. Det var ønskelig å bruke en løsning der mikrofonmatrisen kunne bevege seg på den måten at den kunne detektere støykilden og rette seg etter den. Det vil si en bevegelse rundt boogien som var plassert inne i testbenken. Måten dette skulle løses på var å bruke en integrert løsning som hadde en mikrofonmatrise og en TNS koblet sammen. I tillegg til det var det et kamera koblet på toppen av mikrofonmatrisen som skulle hjelpe med orientering i rommet.

I kapittel for resultater kommer det tydelig frem at en analytisk fremstilling av denne bevegelsen er blitt redegjort for. Denne fremstillingen er blitt kalt for en banegenerator og er blitt brukt som en referanse for å definere en filtermodell. Verdier som er hentet ut fra banegenerator har vært utgangspunktet for å bruke som startverdier til filtermodellen. Vi kan se at ved å integrere en viss antall ganger så vil den sanne filtermodellen gi oss de samme data som banegeneratoren, men det er viktig å huske at med simpel bane som har form som en halvsirkel vil en Euler-integrasjon være tilstrekkelig, men for mer avanserte baner må det brukes en mer utvidet og robust løsning som f.eks Runge-Kuttas 2. ordens metode også kjent som Heuns metode. For vår oppgave er det mest relevant å bruke de målte verdiene for filtermodellen. De målte verdiene inneholder støy i form av bias og hvit støy. Dette fører til at det gir oss et ikke-lineært system. Det er blitt laget et Matlab program (vedlegg A) som simulerer banegenerator, sann filtermodell og den målte filtermodellen. For banegeneratoren og sanne filtermodellen stemmer våre antagelser med det som var et ønskelig

resultat, men for den målte filtermodellen har vi ikke hatt muligheten til å etterprøve resultatene som Matlab programmet (vedlegg A) produserer. Måten det skulle ha blitt gjort på var ved å se om de målte verdiene for filtermodellen implementert i en UKF ville gi oss tilnærmet lik den samme banen som banegeneratoren og sanne filtermodellen.

Avslutningsvis kan vi si at det totale resultatet til denne masteroppgaven ble ikke helt som tiltenkt. Vi fikk aldri prøvd ut en mikrofonmatrise i den Dynamiske Testbenken grunnet mangel på både data og mikrofonmatrise og vi fikk heller ikke brukt vår filtermodell til å se hvordan en UKF hadde virket for å gi oss posisjon og orientering for en TNS integrert med en mikrofonmatrise.

## 7 Veien videre

Veien videre for Mantena AS ville være enten å få laget en mikrofonmatrise som passer til formålet eller bestille en skreddersydd mikrofonmatrise fra en av selskapene nevnt i rapporten. Det er fullt mulig og bestille fra andre aktører, men jeg har funnet disse selskapene nevnt i rapporten som de beste på markedet innenfor sitt felt. Fordelen med å bestille en mikrofonmatrise vil være at du vil få et produkt som fungerer, har en garanti og er klar for bruk med engang. Ulempen vil være kostnaden. Det vil uten tvil være mye billigere å lage en mikrofonmatrise og kan faktisk være en oppgave som blir gitt til en bachelor gruppe eller en masterstudent. Neste steg etter dette vil være å få tatt målinger av forskjellige boogier både med en akselerometer og en mikrofonmatrise. Det må også bli tatt en avgjørelse om mikrofonmatrisen skal være stasjonært eller ha mulighet til å bevege seg i rommet. Hvis den kan bevege seg bør denne rapporten legges til grunn for å lage en integrert TNS med en mikrofonmatrise. For å gjøre det må eventuelt løsningen frem til nå i rapporten prøves først ut med en UKF. Det kan også være greit å ta med farget støy for gyroen for videre arbeid siden det vil ha en påvirkningen på filtermodellen i den virkelige verden. Den delen med perspektiv kameramodell vil også ha betydning. Vi har valgt å se på punkter i taket som referansepunkter for orientering og posisjon til matrisen, men en tanke i videre arbeid vil være å se på punkter i det horisontale planet for matrisen.

## Referanser

- [1] Andersen, P. B. [2018, 26 januar], ‘Kalmanfilter. i store norske leksikon’.  
**URL:** <https://snl.no/kalmanfilter>
- [2] *Baker Huges a GE company* [2019, 21 mars].  
**URL:** <https://www.gemeasurement.com/de/node/2221>
- [3] *Bruel og Kjør - The sound and vibration specialist, hentet 05 april 2019* [2019].  
**URL:** <https://www.bksv.com/en>
- [4] *ClearOne - Beamforming- and Ceiling Microphone Arrays* [2019, 11 mars].  
**URL:** <https://www.clearone.com>
- [5] Flatval, I. [2018], Vision aided ins - comparing the ukf and ekf, Master’s thesis, Universitetet i Oslo, Masteroppgave ved fysiske institutt-MatNat.
- [6] Fredrikstad, T. E. N. [2017], Vision aided inertial navigation, Master’s thesis, NTNU - Norwegian University of Science and Technology.
- [7] *Gatekeeper Intelligent Security* [2019, 05 mars].  
**URL:** <https://www.gatekeepersecurity.com>
- [8] *Guinness World Records* [2019, 07 mars].  
**URL:** <http://www.guinnessworldrecords.com/world-records/largest-microphone-array/>
- [9] Hallingstad, O. [2004], Tns - modeller.
- [10] Hallingstad, O. [2005], Standardmodeller og kalmanfilterlikninger. Notat 1.
- [11] Hallingstad, O. [2017], Ukf - sigmapunkt (unscented) kalmanfilter. Notat 9.
- [12] Hallingstad, O. [2019], Simulering av treghetsnavigasjonssystem.
- [13] Hallingstad, O. [Ukjent], Diskritisering av ulineære stokastisk prosesser. EKF og UKF - håndskrevne notater. Hentet 11. mai 2019.
- [14] Hervik, S. [2017, 16 januar], ‘Eulers metode. i store norske leksikon. hentet 13. april 2019’.  
**URL:** <https://snl.no/Eulersmetode>
- [15] Holden, H. [2017, 24. januar], ‘Taylors formel, store norske leksikon. hentet 25. mars 2019’.  
**URL:** <https://snl.no/Taylors.formel>
- [16] Horn, C. [2015], Modellering av mems- treghetssensorer i et navigasjonssystem, Master’s thesis, Universitetet i Oslo.
- [17] Hossain, A. [2014], Analyse av banegenerator, Master’s thesis, Universitetet i Oslo, Masteroppgave ved fysiske institutt-MatNat.

- [18] Kristensen, O. and Aanensen, S. [2019, 07 april], ‘Enhetssirkel’.  
**URL:** <https://ndla.no/subjects/subject:32/topic:1:165938/topic:1:165939/resource:1:116751>
- [19] *Lake sight technologies* [2019, 03 mars].  
**URL:** <https://www.lakesighttechnologies.com/story/3d-image-capturing-system-for-on-track-train-inspection/>
- [20] *Lake sight technologies* [2019, 26 februar].  
**URL:** <https://www.lakesighttechnologies.com>
- [21] *Learning about electronics - Microphone array* [2019, 06 mars].  
**URL:** <http://www.learningaboutelectronics.com/Articles/What-is-an-array-microphone.html>
- [22] Lied, F. [2009, 15. februar], ‘Treghetsnavigasjon. i store norske leksikon. hentet 31. mars 2019’.  
**URL:** <https://snl.no/treghetsnavigasjon>
- [23] Majidian, P. [2014], 3-d banegenerator, Master’s thesis, Universitetet i Oslo, Masteroppgave ved UNIK-MatNat.
- [24] *MTi User Manual, MTi 10-series and MTi 100-series* [2012].  
**URL:** <https://sensores-de-medida.es/wp-content/uploads/2017/08/mtiusermanual.pdf>
- [25] Noureldin, A., Karamat, T. B. and Georgy, J. [2013], *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*, Springer-Verlag Berlin Heidelberg.
- [26] Opsahl, T. [2018], Unik4690 - maskinsyn. Perspektiv kamera model, forelesningsnotater.
- [27] *Phased Array Ultrasonic Testing of Railroad Wheels - GE* [2019, 21 mars].  
**URL:** <https://www.youtube.com/watch?v=zcrpRv0qPqM>
- [28] Singh, H. [2018, 27 april], ‘The unscented kalman filter: Anything ekf can do i can do it better!’, hentet 26.01.2019’.  
**URL:** <https://towardsdatascience.com/the-unscented-kalman-filter-anything-ekf-can-do-i-can-do-it-better-ce7c773cf88d>
- [29] *Sorama* [2019, 22 mars].  
**URL:** <https://www.sorama.eu>
- [30] *Sorama - Cam1k* [2019, 22 mars].  
**URL:** <https://www.sorama.eu/cam1k>
- [31] *Sorama - Cam64* [2019, 22 mars].  
**URL:** <https://www.sorama.eu/cam64>
- [32] *Vocal - beamforming* [2019, 17 mars].  
**URL:** <https://www.vocal.com/beamforming/>
- [33] *Vocal - microphone array beamforming* [2019, 17 mars].  
**URL:** <https://www.vocal.com/beamforming/microphone-array/>

- [34] Wan, E. A. and Van Der Merwe, R. [2000], The unscented kalman filter for nonlinear estimation, *in* 'Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)', Ieee, pp. 153–158.
- [35] Woodman, O. J. [2007], An introduction to inertial navigation, Technical report, University of Cambridge.
- [36] *ZyLIA* [2019, 11 mars].  
**URL:** <https://www.zyLIA.co/white-paper.html>

## A Matlab koden

```
1 % ----- Masteroppgven til Zeeshan Aslam (2019) ----- %
2
3 % 1. Meste av Matlab-koden kommer fra psuedokoden som er i rapporten og fra
4 % notatet til Oddvar Hallingstad "Simulering av TNS".
5 % 2. Initial veridene og en del variabler som blir brukt er gjennomtenkt og
6 % skal vere
7 % paa den formen at programmet og simuleringen blir enkelt matematisk
8 % fremstilt
9 % 3. Alle forskjellige plotene av simuleringsmodellen er ogsaa laget i
10 % hovedprogrammet
11
12 %% ----- INITIAL- OG STARTVERDIER ----- %%
13
14 % ---- banegenerator ---- %
15 r = 5; % radiusen til banegeneratoren
16 n_s = 3001; % antall sampler
17 n_t = 15; % antall tilstander
18 T = 30; % en hel tidsomlop i sekunder
19 om = pi/T; % vinkelhastighet (konstant)
20 dt= T/(n_s-1); % samplings interevallet
21 g=[0;0;9.81]; % gravitasjons konstant
22 P = zeros(3,n_s+1); % tom matrise med nuller for aa lagre posisjon vektorer
23 til banegeneratoren
24 V = zeros(3,n_s+1); % tom matrise med nuller for aa lagre farts vektorer til
25 banegeneratoren
26 A = zeros(3,n_s+1); % tom matrise med nuller for aa lagre akselerasjon
27 vektorer til banegeneratoren
28 f_b = zeros(3,n_s+1); % tom matrise med nuller for aa lagre sann spesifikk
29 kraft til b-rammen
30 fb_b = zeros(3,n_s+1); % tom matrise med nuller for aa lagre maalt spesifikk
31 kraft til b-rammen
32 Omega_b = zeros(3,n_s+1); % tom matrise med nuller for aa lagre sann
33 vinkelhastighet til b-rammen
34 Omegab_b = zeros(3,n_s+1); % tom matrise med nuller for aa lagre maalt
35 vinkelhastighet til b-rammen
36 u_a = zeros(3,n_s+1); % tom matrise med nuller for aa lagre maaleverdier fra
37 Euler-integrasjon (spesifikk kraft bolge)
38 u_g = zeros(3,n_s+1); % tom matrise med nuller for aa lagre maaleverdier fra
39 Euler-integrasjon (vinkelhastighet bolge)
40
41
42 % ---- kamera delen ---- %
43 r_1 = [1;1;3]; % punkt 1 som er fast og kjent presentert i verden
44 r_2 = [1;-1;3]; % fpunkt 2 som er fast og kjent presentert i verden
45 rho_1 = zeros(3,n_s+1); % punkt 1 i verden projisert i bildeplanet
46 rho_2 = zeros(3,n_s+1); % punkt 2 i verden projisert i bildeplanet
```



```

36 p_bc_b = [0;0;0.1]; % translasjonsvektor som er konstant og kjent mellom
      kamera-rammen (f^c) og plattform-rammen (f^b)
37 R_c_b = [cos(pi/2), 0, sin(pi/2);
38           0, 1, 0;
39          -sin(pi/2), 0, cos(pi/2)]; % rotasjonsmatrise som konstant og kjent
      mellom kamera-rammen (f^c) og plattform-rammen (f^b)
40
41
42 % ---- filtermodellen ---- %
43 PE = zeros(3,n_s); % tom matrise med nuller for aa lagre posisjon vektorer til
      filtermodellen for Euler-integrasjon
44 VE = zeros(3,n_s); % tom matrise med nuller for aa lagre fart vektorer til
      filtermodellen for Euler-integrasjon
45 TevE = zeros(3,n_s); % tom matrise med nuller for aa lagre theta vektorer til
      filtermodellen for Euler-integrasjon
46 PEm = zeros(3,n_s); % tom matrise med nuller for aa lagre posisjon vektorer
      til filtermodellen for Euler-integrasjon med stoy
47 VEm = zeros(3,n_s); % tom matrise med nuller for aa lagre fart vektorer til
      filtermodellen for Euler-integrasjon med stoy
48 TevEm = zeros(3,n_s); % tom matrise med nuller for aa lagre theta vektorer til
      filtermodellen for Euler-integrasjon med stoy
49 TA = 0:n_s; %
50 TA = dt*TA; %
51
52
53 % ---- bias for akselerometer og gyro ----- %
54 % ND_a = 150*[1;1;1]; % Stoytetthet for akselerometer (micro g/sqrt(Hz))
55 % ND_g = 0.05*[1;1;1]; % Stoytetthet for gyro (grader/sqrt(Hz))
56 % b_a = 0.03*[1;1;1]; % Standardavvik for bias-akselerometer
57 % b_g_1 = 0.2*[1;1;1]; % Standardavvik for bias-gyro (grader/s)
58 % b_g_1 = deg2rad(b_g); % Bias-gyro konvertert til (rad/s)
59 b_a = zeros(3,n_s+1); % Tom array for bias-akselerometer
60 b_g = zeros(3,n_s+1); % Tom array for bias-gyro
61 sigma_b_a = 0.03*[1;1;1]; % Bias akselerometer
62 sigma_b_g = 0.2*(2*pi/360)*[1;1;1]; % Bias gyro
63 b_a(:, :) = sigma_b_a(3, :); % Bias til akselerometer
64 b_g(:, :) = sigma_b_g(3, :); % Bias til gyro
65
66
67 % ---- hvit stoy for akselerometer og gyro ---- %
68 sigmab_va = 2.34*1e-4; % Standardavvik for hvitstoy diskritisert med Euler-
      metode for akselerometer, for k ? {1,...,N}
69 sigmab_vg = 0.75; % Standardavvik for hvitstoy diskritisert med Euler-metode
      for gyro, for k ? {1,...,N}
70 sigma_va = sigmab_va*randn(3,n_s+1); % Hvit stoy til akselerometer
71 sigma_vg = sigmab_vg*randn(3,n_s+1); % Hvit stoy til gyro
72
73
74 % ---- estimator ---- %
75 X = zeros(n_t,n_s+1); % tom matrise med nuller for aa lagre tilstander fra
      banegeneratoren (p^n,v^n,theta,b^a,b^g) -- Fx

```

```

76 U = zeros(6,n_s+1);          % tom matrise med nuller for aa lagre maalte verdier
    for filtermodell (fb^b,omegab_b^nb)      -- Lu
77 v_stoy = zeros(6,n_s+1); % tom matrise med nuller for aa lagre hvit stoy
                                -- Gv
78 Z = zeros(6,n_s+1);          % tom matrise med nuller for aa lagre utgangsverdier
    for filtermodell (s_x,s_y)              -- Hx
79
80
81 %% ---- Banegenerator for analytisk-losning (derivasjon) ---- %%
82 for k = 0 : n_s
83     te = k*dt*om; % Regner ut theta for banegenerator
84     P(:,k+1) = [r*cos(te),; r*sin(te);1]; % posisjon banegenerator
85     V(:,k+1) = r*om*[-sin(te);cos(te);0]; % fart banegenerator
86     A(:,k+1) = r*om^2*[-cos(te);-sin(te);0]; % akcelerasjon banegenerator
87     f_b(:,k+1) = [cos(te), sin(te),0;
88                 -sin(te), cos(te),0;
89                 0, 0, 1]*(A(:,k+1)+g); % spesifikk kraft regnet ut fra banegenerator
90     Omega_b(:,k+1) = [0;0;om]; % vinkelhastighet regnet ut fra banegenerator
91 end
92
93
94 %% ----- MaaLTE VERIDER FOR FILTERMODELL ----- %%
95 for k = 1 : n_s
96     fb_b(:,k+1) = f_b(:,k) - b_a(:,k) - sqrt(dt)*sigma_va(:,k); % maalt
    spesifikk kraft med bias og hvitstoy
97     Omegab_b(:,k+1) = Omega_b(:,k) - b_g(:,k) - sqrt(dt)*sigma_vg(:,k); %
    maalt vinkelhastighet med bias og hvitstoy
98 end
99
100 for k = 1 : n_s-1
101
102     u_a(:,k+1) = (f_b(:,k)+f_b(:,k+1))/2 - b_a(:,k) - sqrt(dt)*sigma_va(:,k);
103     u_g(:,k+1) = (Omega_b(:,k)+Omega_b(:,k+1))/2 - b_g(:,k) -sqrt(dt)*sigma_vg
    (:,k);
104 end
105
106
107 %% ----- INITIALVERDIER FOR SANN FILTERMODELL (EULER-INTEGRASJON) ----- %%
108 p = [5;0;1]; % initialverdi for posisjon til Euler-integrasjon
109 v = [0;om*r;0]; % initialverdi for fart til Euler-integrasjon
110 tev = [0;0;0]; % % initialverdi for theta til Euler-integrasjon
111
112 PE(:,1) = p; % setter initialverdi for posisjon inn i matrisen for startverdi
113 VE(:,1) = v; % setter initialverdi for fart inn i matrisen for startverdi
114 TevE(:,1) = tev; % setter initialverdi for theta inn i matrisen for startverdi
115
116
117 %% ---- Sann filtermodell for banegeneratoren med Euler-integrasjon ---- %%
118 for k = 1 : n_s
119     p = p + dt*v; % Euler-integrer for posisjon
120     R_b_n = R_3(tev(3))*R_2(tev(2))*R_1(tev(1)); % Euler-integrer for fart

```

```

121     v=v+dt*R_b_n*f_b(:,k)-dt*g; %
122     D_b_te = D_te_b(tev)'; % Euler-integrerer for theta
123     tev = tev + dt*D_b_te*Omega_b(:,k);
124     PE(:,k+1) = p; % lager verdiene inn i den tomme matrisen for posisjon
125     VE(:,k+1) = v; % lager verdiene inn i den tomme matrisen for fart
126     TevE(:,k+1) = tev; % lager verdiene inn i den tomme matrisen for theta
127 end
128
129
130 %% ----- INITIALVERDIER FOR MaaLT FILTERMODELL (EULER-INTEGRASJON) ----- %%
131 pm =[5;0;1]; % initialverdi for posisjon til Euler-integrasjon med stoy
132 vm = [0;om*r;0]; % initialverdi for fart til Euler-integrasjon med stoy
133 tevm =[0;0;0]; % % initialverdi for theta til Euler-integrasjon med stoy
134
135 PEm(:,1) = pm; % setter initialverdi for posisjon inn i matrisen for
    startverdi med stoy
136 VEm(:,1) = vm; % setter initialverdi for fart inn i matrisen for startverdi
    med stoy
137 TevEm(:,1) = tevm; % setter initialverdi for theta inn i matrisen for
    startverdi med stoy
138
139
140 %% ---- Maalt filtermodell for banegeneratoren med Euler-integrasjon ---- %%
141 for k = 1 : n_s
142     pm = pm + dt*vm; % Euler-integrerer for posisjon med stoy
143     R_b_n = R_3(tevm(3))*R_3(tevm(2))*R_1(tevm(1)); % Euler-integrerer for fart
        med stoy
144     vm=vm+dt*R_b_n*fb_b(:,k)-dt*g; %
145     D_b_te = D_te_b(tevm)'; % Euler-integrerer for theta
146     tevm = tevm + dt*D_b_te*Omegab_b(:,k);
147     PEm(:,k+1) = pm; % lager verdiene inn i den tomme matrisen for maalt
        posisjon med stoy
148     VEm(:,k+1) = vm; % lager verdiene inn i den tomme matrisen for maalt fart
        med stoy
149     TevEm(:,k+1) = tevm; % lager verdiene inn i den tomme matrisen for theta
150 end
151
152
153 %% Sensorverdier(utgangsverider) hentet fra punkter i verden
154 for k = 1 : n_s
155     rho_1(:,k) = R_c_b'*R_b_n'*(r_1-P(:,k)) - R_c_b*p_bc_b;
156     rho_2(:,k) = R_c_b'*R_b_n'*(r_2-P(:,k)) - R_c_b*p_bc_b;
157     % x-koordinaten i bildeplaneet for punkt 1
158     s_x_1 = rho_1(1,k);
159     % y-koordinaten i bildeplaneet
160     s_y_1 = rho_1(2,k);
161     % z-koordinaten i bildeplaneet
162     z_1 = rho_1(3,k);
163     % x-koordinaten i bildeplaneet for punkt 2
164     s_x_2 = rho_2(1,k);
165     % y-koordinaten i bildeplaneet

```

```

166     s_y_2 = rho_2(2,k);
167     % z-koordinaten i bildeplaneet
168     z_2    = rho_2(3,k);
169     Z(1:3,k) = [s_x_1 s_y_1 z_1]/z_1'; % plasserer forste punktet hentet fra
        bildeplanet i kamera
170     Z(4:6,k) = [s_x_2 s_y_2 z_2]/z_2'; % plasserer andre punktet hentet fra
        bildeplanet i kamera
171 end
172
173
174 %% ---- TILSTANDS MATRISE FOR ESTIMATOR ---- %%
175 X(1:3,:) = P(:, :); % plasserer posisjons vektorer i tilstands matrisen
176 X(4:6,:) = V(:, :); % plasserer farts vektorer i tilstands matrisen
177 X(7:9,:) = TevE(:, :); % plasserer theta vektorer i tilstands matrisen
178 X(10:12,:) = b_a(:, :); % plasserer bias for akselerometer i tilstands matrisen
179 X(13:15,:) = b_g(:, :); % plasserer bias for gyro i tilstands matrisen
180
181
182 %% --- PaaDRAG MATRISE FOR ESTIMATOR --- %%
183 U(1:3,:) = u_a(:, :); % plasserer paadraget maalte spesifikk kraft i matrisen
184 U(4:6,:) = u_g(:, :); % plasserer paadraget maalte vinkelhastigheten i matrisen
185
186
187 %% --- HVIT SToY MATRISE FOR ESTIMATOR --- %%
188 v_stoy(1:3,:) = sigma_va(:, :); % hvit stoy for akselerometer
189 v_stoy(4:6,:) = sigma_vg(:, :); % hvit stoy for gyro
190
191
192 %% --- UTGANGSVERDIER FOR ESTIMATOR ---- %%
193 Z(1:3, :);
194 Z(4:6, :);
195
196
197 %%% ----- ALLE PLOTTER FOR SIMULERINGER ----- %%
198
199 % 3D-plot av banegeneratoren analytisk (derivasjon)
200 figure
201 plot3(P(1,:),P(2,:),P(3,:), 'o', 'Color', 'r', 'MarkerSize', 5)
202 grid on
203 title('Banegenerator - 3D-posisjon til mikrofonmatrisen i ramme F^n')
204 xlabel('x-akse')
205 ylabel('y-akse')
206 zlabel('z-akse')
207 view([-41.70 43.22])
208 hold off
209
210 % 2D-plot av banegeneratoren analytisk (derivasjon)
211 figure
212 plot(P(1,:),P(2,:), 'r')
213 grid on
214 title('Banegenerator - 2D-posisjon til mikrofonmatrisen i ramme F^n')

```

```

215 xlabel('x-akse')
216 ylabel('y-akse')
217 hold off
218
219 % Fart fra banegeneratoren
220 figure
221 plot(TA,V(2,:), 'r')
222 grid on
223 title('Banegenerator - fart til mikrofonmatrisen rundt boggien')
224 xlabel('tid')
225 ylabel('y-akse')
226 hold off
227
228 % 3D-plot av sann filtermodell (Euler-integrasjon)
229 figure
230 plot3(PE(1,:),PE(2,:),PE(3,:), 'o', 'Color', 'b', 'MarkerSize', 5)
231 grid on
232 title('Sann filtermodell - 3D-posisjon med Euler-integrasjon')
233 xlabel('x-akse')
234 ylabel('y-akse')
235 zlabel('z-akse')
236 view([-41.70 43.22])
237 hold off
238
239 % 2D-plot av sann filtermodell (Euler-integrasjon)
240 figure
241 plot(PE(1,:),PE(2,:), 'b')
242 grid on
243 title('Sann filtermodell - 2D-posisjon med Euler-integrasjon')
244 xlabel('x-akse')
245 ylabel('y-akse')
246 hold off
247
248 % Fart fra filtermodellen
249 figure
250 plot(TA,VE(2,:), 'b')
251 grid on
252 title('Sann filtermodell - fart til mikrofonmatrisen rundt boggien med Euler-
      integrasjon')
253 xlabel('tid')
254 ylabel('y-akse')
255 hold off
256
257 % Banegenerator og filtermodell posisjon overlappet (forandr n_z til
      30,300,3000 osv for aa se forskjellen)
258 plot(P(1,:),P(2,:),PE(1,:),PE(2,:));
259 grid on;
260 title('Posisjon overlappet for Banegenerator og filtermodell')
261 xlabel('x-akse')
262 ylabel('y-akse')
263 legend('Banegenerator_{pos}', 'Filtermodell_{pos}')

```

```

264
265
266 % Banegenerator og filtermodell fart overlappet (forandr n_z til 30,300,3000
      osv for aa se forskjellen)
267 plot(TA,V(2,:),TA,VE(2,:));
268 grid on;
269 title('Fart overlappet for Banegenerator og filtermodell')
270 xlabel('tid')
271 ylabel('y-akse')
272 legend('Banegenerator_{fart}','Filtermodell_{fart}')
273
274
275
276
277 % 2D-plot av den maalte filtermodellen
278 plot(PEm(1,:),PEm(2,:))
279 grid on
280 title('Maalt filtermodell - 2D-posisjon med Euler-integrasjon')
281 xlabel('x-akse')
282 ylabel('y-akse')
283 hold off
284
285
286 % Banegenerator og maalt filtermodell posisjon overlappet
287 plot(P(1,:),P(2,:),PEm(1,:),PEm(2,:))
288 axis equal
289 grid on;
290 title('Posisjon overlappet for Banegenerator og maalt filtermodell')
291 xlabel('x-akse')
292 ylabel('y-akse')
293 legend('Banegenerator_{pos}','Maalt filtermodell_{pos}')
294
295 % Banegenerator og maalt filtermodell posisjon overlappet med forste 50
      iterasjoner
296 plot(P(1,1:50),P(2,1:50),PEm(1,1:50),PEm(2,1:50))
297 axis equal
298 grid on;
299 title('Posisjon overlappet for Banegenerator og maalt filtermodell 50
      iterasjoner')
300 xlabel('x-akse')
301 ylabel('y-akse')
302 legend('Banegenerator_{pos}','Maalt filtermodell_{pos}')
303
304 % Banegenerator og maalt filtermodell - posisjon overlappet med ved 500
      iterasjoner
305 plot(P(1,1:500),P(2,1:500),PEm(1,1:500),PEm(2,1:500))
306 axis equal
307 grid on;
308 title('Posisjon overlappet for Banegenerator og maalt filtermodell 500
      iterasjoner')
309 xlabel('x-akse')

```

```

310 ylabel('y-akse')
311 legend('Banegenerator_{pos}','Maalt filtermodell_{pos}')
312
313 % Banegenerator og maalt filtermodell - posisjon overlappet med ved 1500
    iterasjoner
314 plot(P(1,1:1500),P(2,1:1500),PEm(1,1:1500),PEm(2,1:1500))
315 axis equal
316 grid on;
317 title('Posisjon overlappet for Banegenerator og maalt filtermodell - 1500
    iterasjoner')
318 xlabel('x-akse')
319 ylabel('y-akse')
320 legend('Banegenerator_{pos}','Maalt filtermodell_{pos}')
321
322
323
324
325
326
327
328
329 %%
330 function R = R_1(te)
331 R = [1, 0, 0; 0, cos(te), -sin(te); 0, sin(te), cos(te)];
332 end
333 function R = R_2(te)
334 R = [cos(te), 0, sin(te); 0, 1, 0; -sin(te), 0, cos(te)];
335 end
336 function R = R_3(te)
337 R = [cos(te), -sin(te), 0; sin(te), cos(te), 0; 0, 0, 1];
338 end
339
340 function D = D_te_b(tev)
341 D = [1,0,-sin(tev(2)); 0, cos(tev(1)), sin(tev(1))* cos(tev(2)); ...
342     0, -sin(tev(1)), cos(tev(1))*cos(tev(2))];
343 end

```