

# Modellbasert klassifisering av spedbarns bevegelser

**Andreas Berg**

Master i teknisk kybernetikk  
Oppgaven levert: Juni 2008  
Hovedveileder: Geir Mathisen, ITK



# Oppgavetekst

Rett etter fødselen har friske barn helt karakteristiske bevegelser kalt Fidgety General Movements, mens barn som senere utvikler Cerebral Parese mangler disse. Det eksisterer i dag en klinisk metode kalt General Movement Assessment (GMA) med høy deteksjonsrate, men av praktiske grunner er den lite utbredt i klinikkene. Som et ledd i arbeidet med å skape et teknisk måleutstyr som kan assistere klinikerne i diagnosearbeidet, skal du i denne oppgaven studere bruk av dynamisk systemteori og modelltilpasninger for å klassifisere bevegelser som hhv. normale eller avvikende.

1. Gjør et litteratursøk og lag en oversikt over hvordan dynamiske modeller i kombinasjon med ulike klassifiseringsmetoder har vært brukt i analyse og klassifisering av biologisk bevegelse (stikkord: normal vs. unormal/patologisk bevegelse hos mennesker og dyr, optimal vs. suboptimal bevegelse i for eksempel idrett).
2. Velg en eller to metoder fra litteraturen eller foreslå nye metoder for bruk av dynamiske modeller, tilstandsestimering og/eller systemidentifikasjon med potensiell relevans til den foreliggende problemstillingen. Vis konkret hvordan metoden(e) kan benytte foreliggende datamateriale for å diskriminere mellom normale og avvikende spedbarnsbevegelser.
3. Gjør et begrunnet valg av metoder basert på funnene i 1. og 2., og implementer de valgte metodene. I forbindelse med utvalg av egenskaper for klassifisering skal det legges vekt på at de ulike bevegelsessensorene kanskje gir svært forskjellig bidrag til metodens sensitivitet og spesifisitet.
4. I den grad tiden tillater det, evaluer resultatene opp mot GMA og andre "konkurrerende" metoder.

Oppgaven gitt: 07. januar 2008  
Hovedveileder: Geir Mathisen, ITK



# Forord

Denne rapporten er et resultat av masteroppgaven avlagt ved NTNU, institutt for teknisk kybernetikk. Arbeidet har bestått i et litteratursøk, samt implementering av metoder i `MATLAB`. Det har vært interessant å få bruke dynamisk systemteori på måledata som kommer fra spedbarn, og ikke et teknisk system. Videre har det vært nyttig å sette seg inn i metoder innenfor fagfeltet klassifisering, som før oppgaven var ukjent for meg. Det har vært motiverende å bruke virkelige data, og å få metodene til å klassifisere best mulig.

Jeg vil takke veileder Øyvind Stavdahl for å komme med mange gode og verdifulle innspill.

Andreas Berg  
Trondheim, 9. juni 2008



# Sammendrag

Hensikten med denne oppgaven er å bruke dynamiske modeller til å klassifisere spedbarnsbevegelser som normale eller avvikende. Bevegelser er registrert med 6 sensorer, festet til ankler, håndledd, bryst og panne. Hver sensor måler posisjon i rommet og rotasjon, og i denne oppgaven er det blitt fokusert på posisjon.

Hvert spedbarn har en eller flere segmenter med måledata, kalt Region of Interest (ROI). Ut i fra hver ROI er det blitt identifisert en dynamisk modell på grunnlag av utgangsvARIABLE. I dette arbeidet er det brukt SISO og MIMO AR- og tilstandsrommodeller. UtgangsvARIABLENE representerer utgangen på den dynamiske modellen, og mange forskjellige alternativer har blitt prøvd ut.

Klassifisering er blitt foretatt på bakgrunn av de identifiserte systemparameterne. To strategier er fulgt: Den første baserer seg på å bruke systemparameterne direkte som egenskpsvektor sammen med lineær diskriminantanalyse. Den andre tar utgangspunkt i metrikker som måler en distanse mellom to dynamiske modeller, og som bruker k-nærmeste-nabo som klassifikator.

Resultatene viser at bevegelsene til armene og hodet er viktigst for å skille mellom den friske og avvikende gruppen. Ved å kun betrakte hodet i  $y$ -retning, på tvers av spedbarnet, oppnås en spesifisitet på 78.26% og en sensitivitet på 78.64% på ROI-nivå. Ved å se på avstanden mellom armene i  $x$ -retning, langs spedbarnet, oppnås en spesifisitet på 79.61% og en sensitivitet på 82.61%. I begge tilfellene er det brukt 4.ordens SISO AR-modeller og LDA som klassifikator. Ved å kombinere egenskpsvektorene fra disse to tilfellene sammen i en ny egenskpsvektor oppnås en spesifisitet på 86.96% og en sensitivitet på 85.44%. Klassifisering på pasient-nivå foregår ved å betrakte ROI'ene tilhørende hver pasient. Ved å vekte friske og avvikende ROI'er forskjellig, kan ulike verdier for spesifisitet og sensitivitet oppnås. Resultatet nærmest det optimale punktet i ROC-plottet har en spesifisitet på 90.91% og sensitivitet på 85.71%.

Verdiene på spesifisitet og sensitivitet gitt ovenfor viser at tilpasning av dynamiske modeller er en nyttig metode for klassifisering av spedbarnsbevegelser. Metoden er ikke like nøyaktig som den kliniske metoden GMA, men det er mange muligheter for å forbedre resultatene oppnådd i dette arbeidet.





# Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
<b>2</b>	<b>Bakgrunn</b>	<b>3</b>
<b>3</b>	<b>Metode</b>	<b>13</b>
3.1	Måledata . . . . .	13
3.2	Segmentering og valg av utgangsvARIABLE . . . . .	15
3.2.1	PCA for reduksjon i dimensjon på måledata . . . . .	18
3.3	Modell . . . . .	19
3.3.1	Modellstruktur . . . . .	19
3.3.2	Analyse av modell . . . . .	21
3.3.3	Orden på AR-modell . . . . .	22
3.4	Systemidentifikasjon . . . . .	22
3.4.1	Minste kvadraters metode . . . . .	23
3.4.2	Yule-walker . . . . .	23
3.4.3	Minste kvadraters metode for MIMO modeller . . . . .	24
3.4.4	Subspace identification . . . . .	24
3.5	Klassifisering . . . . .	25
3.5.1	Lineær diskriminantanalyse(LDA) . . . . .	25
3.5.2	K-Nærmeste Nabo(KNN) . . . . .	26
3.5.3	Metrikker for dynamiske modeller . . . . .	27
3.5.4	Mål på separabilitet . . . . .	30
3.5.5	Kombinering av egenskapsvektorer . . . . .	32
3.6	Oversikt over algoritmer . . . . .	33
<b>4</b>	<b>Resultater</b>	<b>35</b>
4.1	Orden på AR-modell . . . . .	35
4.2	Definisjon av sensitivitet og spesifisitet . . . . .	36
4.3	PCA og varians . . . . .	38
4.4	Vekting av ROI'er . . . . .	41
4.5	Resultater fra separabilitetsanalyse og klassifisering . . . . .	41
4.5.1	SISO-modell . . . . .	42
4.5.2	MIMO-modell . . . . .	49

4.6	Endring av orden og metode . . . . .	59
4.7	Analyse av modell . . . . .	62
<b>5</b>	<b>Diskusjon</b>	<b>69</b>
5.1	Videre arbeid . . . . .	70
5.2	Evaluering opp mot GMA . . . . .	72
<b>6</b>	<b>Konklusjon</b>	<b>73</b>
	<b>Bibliografi</b>	<b>77</b>
<b>A</b>	<b>Seperabilitet</b>	<b>79</b>
<b>B</b>	<b>PCA med tidsserie som observasjonsvektor</b>	<b>81</b>
<b>C</b>	<b>Implementering i MATLAB</b>	<b>83</b>

# Kapittel 1

## Innledning

I denne oppgaven studeres bevegelser fra to grupper spedbarn, friske spedbarn og barn som senere utvikler Cerebral Parese. Friske spedbarn har karakteristiske bevegelser kalt Fidgety General Movements, mens barn som senere utvikler cerebral parese mangler disse. Målet med oppgaven er å undersøke hvor godt man klarer å identifisere bevegelser som friske eller avvikende, ved å ta i bruk metoder fra dynamisk systemteori.

I Kapittel 2 gis det en oversikt over metoder som har vært brukt i analyse og klassifisering av biologisk bevegelse. Kapitlet beskriver metoder som tar i bruk dynamiske modeller, eller som kan brukes i sammenheng med slike ved klassifisering.

I Kapittel 3 presenteres det hvordan bevegelsesdata fra spedbarn kan modelleres ved bruk av dynamiske modeller. Det blir vist hvordan systemparameterne i disse modellene kan identifiseres ved bruk av metoder innenfor systemidentifikasjon, og det presenteres hvordan de resulterende parameterne kan brukes som utgangspunkt for klassifisering. Kapitlet tar også for seg sentrale systemteoretiske egenskaper, som kan brukes til å sammenligne modeller fra de to gruppene.

I Kapittel 4 presenteres resultater fra modelltilpasning og klassifisering. Kapittel 5 inneholder en overordnet diskusjon av resultatene, og de vurderes opp mot den kliniske GMA-metoden.



## Kapittel 2

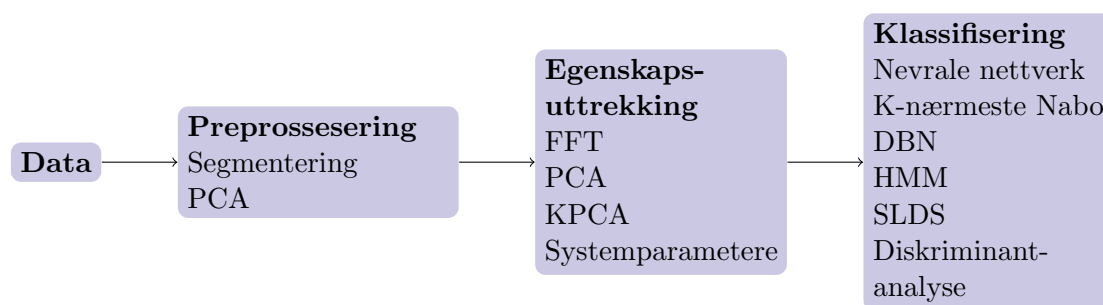
# Bakgrunn

Det meste av litteraturen som presenteres i dette kapitlet dreier seg om menneskelig bevegelse. Det er lagt vekt på å finne litteratur som tar i bruk dynamiske modeller i en eller annen form. Siden ARMA-modeller har vært mye brukt i klassifisering av myoelektriske signaler er noen eksempler kort diskutert, selv om de ikke innebærer direkte klassifisering av menneskelig bevegelse. Denne oppgaven er ikke i noen stor grad fokusert på hva som finnes av forskning på spedbarnsbevegelser og Fidgety-bevegelser. Bruk av dynamiske modeller for klassifisering er en generell innfallsvinkel som kan brukes for mange forskjellige typer signaler.

Da dette kapitlet skal presentere relevant litteratur for oppgaven, er det verdt å se litt nærmere på hvilke delalgoritmer oppgaven kan deles inn i. Figur 2.1 viser hvilke algoritmer som kan være del av et system som klassifiserer bevegelse.

Utgangspunktet er at vi har en mengde innsamlet bevegelsesdata. Ofte er det nødvendig med en form for preprossesering av dataene. Dette kan blant annet innebære lavpassfiltrering av støyete måledata eller utfylling av manglende data. Kanskje er det også ønskelig å redusere dimensjonen til dataene, og da kan prinsippal komponentanalyse (PCA) være aktuelt. Siden menneskelig bevegelse er kompleks, kan det være interessant å se om det er mulig å dele opp en sammensatt bevegelse i mindre segmenter. Hvis totalbevegelsen består av flere deler, der hver del har sin egne karakteristiske dynamikk eller mønster, kan det bli lettere å beskrive totalbevegelsen ved å se på delene hver for seg. Dette kan for eksempel være tilfellet hvis man betrakter mange etterfølgende forskjellige mer eller mindre målrettede bevegelser. Hvis bevegelsene er mer tilfeldige, som i denne oppgaven, er det langt fra åpenbart hvordan man skal segmentere bevegelser, og om det har noen hensikt.

Etter preprosseseringen vil man gjerne prøve å representere dataene ved bruk av noen få parametere. Dette kalles egenskapsuttrekking og de utvalgte parameterne samles i en egenskapsvektor. Målet er å finne frem til parametere som diskriminerer godt mellom



Figur 2.1: Algoritmer for klassifisering av bevegelse

forskjellig bevegelse. Legg merke til at PCA også er ført opp under egenskapsuttrekking i tillegg til preprossesering, og dette vil bli klargjort senere i kapitlet. I denne oppgaven er målet å tilpasse dynamiske modeller til bevegelsesdataene. Dette blir en form for egenskapsuttrekking, der systemparameterne til de dynamiske modellene utgjør egenskapsvektorene som ligger til grunn for klassifisering.

Klassifiseringen av egenskapsvektorene kan gjøres på mange måter ved bruk av kjente klassifikatorer. Noen de mest brukte når det gjelder klassifisering av bevegelse er nevrale nettverk, hidden Markov-modeller og diskriminantanalyse. Litteratur presentert i det etterfølgende kan stort sett relateres til en eller flere av delalgoritmene i Figur 2.1.

I dag finnes en klinisk metode for å avgjøre om et spedbarn senere vil utvikle Cerebral Parese, kalt General Movements Assessment (GMA). General movements består av et bevegelsesmønster der alle kroppsdeler er involvert, uten at det er noen spesiell rekkefølge mellom bevegelsene til ulike deler. Kvaliteten på General movements kan beskrives av variasjon og kompleksitet. Kompleksitet betyr blant annet forandring i retning til bevegelsene, og variasjon vil si at barnet hele tiden lager nye bevegelsesmønstre (Hadders-Algra 2004). I et stadium etter fødselen vil friske barn foreta små elegante bevegelser, såkalte Fidgety General Movements. Barn som senere utvikler cerebral parese har lavere kvalitet på GM enn friske barn, og mangler Fidgety General Movements. I GMA brukes dette til å fastslå om et spedbarn er friskt, eller om det senere vil utvikle Cerebral Parese. I Prechtl et al. (1997) vises det at GMA klassifiserer 67 av 70 friske korrekt og 57 av 60 av spedbarn som enten har Cerebral Parese eller mindre neurologiske abnormaliteter.

I L.Meinecke et al. (2006) presenteres en undersøkelse av 22 spedbarn, derav 15 er friske. Et 3D bevegelsesanalysesystem brukes for å hente inn data. Parametere som brukes til å beskrive bevegelsene består av fysiske variable hentet ut i fra analysesystemet, samt variable som er ment til å reflektere spesielle bevegelsesmønstre en kliniker observerer. Totalt blir det definert 53 variable og ved hjelp av cluster-analyse blir de 8 parametrene som skiller best mellom friske og syke bestemt. Cluster-analysen foretas på 8 barn der halvparten er friske. Klassifisering blir så utført med de resterende 14 barna, der 11 er friske. Klassifiseringen består av å foreta en sammenligning med barna brukt i

clusteranalysen basert på kvadratisk diskriminantanalyse.

I (Robertson 1982) foretas en analyse av rytmiske bevegelser hos spedbarn 2. eller 3. dag etter fødsel. Armbevegelser blir registrert ved bruk av et kamera 1 til 4 ganger i sekundet. Dataene blir undersøkt ved bruk av effekt-tetthetsspektre, noe som vi også vil gjøre for spedbarnbevegelsene i denne oppgaven.

Selv om menneskelig bevegelse er kompleks, kan det hende at bevegelsene er basert på noen grunnleggende underliggende prinsipper. I Gibet et al. (2004) gis en oversikt over kinematiske lover som er utviklet for hånd- og armbevegelser, som er ment for å beskrive slike grunnleggende sammenhenger. Isochrony-prinsippet sier at hastigheten til en bevegelse er avhengig av lengden til bevegelsen. For bevegelser i et plan uttrykkes denne sammenhengen gjennom Fitt's lov. Two-third power law gir en sammenheng mellom vinkelhastighet og kurvaturen til en bevegelsestrajektor. I P.Viviania & C.Terzuolo (1982) brukes denne sammenhengen til å segmentere kompleks bevegelse opp i mindre deler. For punkt-til-punkt bevegelse har det blitt argumentert for at den resulterende trajektoren vil være den som gir minst rykking (eng: jerk) Gibet et al. (2004).

Klassifisering og gjenkjenning av menneskelig bevegelse er mye studert i sammenheng med videobasert tracking. Tracking og egenskapsuttrekking fra videodata er ikke et fokus i dette arbeidet, men egenskapsvektorer som blir funnet ut i fra videodata kan i noen tilfeller være på samme form som et sensorbasert målesystem. Derfor kan metoder som er brukt innenfor selve klassifiseringen være av interesse. I Moeslund et al. (2006) finnes en oversikt over metoder for tracking og klassifisering basert på videodata.

Dynamiske modeller har vært brukt til å gjenkjenne eller klassifisere forskjellige typer gange. I Bissacco et al. (2001) studeres tre typer gange: normal gange, løping, og gange opp en trapp. Hver type gange representeres av en dynamisk modell som er identifisert fra en tidsserie av leddvinkler. Den dynamiske modellen er på formen:

$$x_{k+1} = Ax_k + v_k \quad (2.1)$$

$$y_k = Cx_k + w_k \quad (2.2)$$

$$v_k \sim N(0, Q), \quad w_k \sim N(0, R) \quad (2.3)$$

der  $A \in \mathbb{R}^{n \times n}$  and  $C \in \mathbb{R}^{m \times n}$ .

I denne modellen representerer  $y_k$  forskjellige leddvinkler. Ved å måle en serie med leddvinkler,  $y_k, k = 1 \dots \tau$ , fås estimater  $\hat{A}$  og  $\hat{C}$  ved å bruke en metode fra systemidentifikasjon kalt n4sid, se seksjon 3.4.4.  $\hat{A}$  og  $\hat{C}$  er systemparameterne som beskriver en bestemt type gange, og det er ønskelig å kunne skille systemparameterne for ulike type gange fra hverandre. For å måle avstanden mellom to modeller presenteres det flere metrikker, som vi vil se nærmere på i seksjon 3.5.3. K-Nærmeste Nabo(KNN) med en av metrikkene som avstandsmål blir foreslått som klassifikator. De samme metrikkene brukes i Jin & Mokhtarian (2005) og Veeraraghavan et al. (2004) for gjenkjenning av menneskelig bevegelse basert på silhuetter.

Tilpasning av dynamiske modeller til en tidsserie av leddvinkler er også diskutert i Lu & Ferrier (2003). Her består tilstandsvektoren  $X$  av vrist-, albue- og skuldervinkler, henholdsvis  $\alpha_w$ ,  $\alpha_e$  og  $\alpha_s$ :

$$[\alpha_w \quad \alpha_e \quad \alpha_s]^T \quad (2.4)$$

Først deles tidsserien opp i mindre segmenter, og så beskrives hvert segment med en andre ordens AR-modell for hver leddvinkel. Segmenteringsalgoritmen bruker en andre ordens AR-modell med flere utganger, der alle leddvinklene behandles sammen:

$$X_k - \bar{X} = A_2(X_{k-2} - \bar{X}) + A_1(X_{k-1} - \bar{X}) + B_0 e_k \quad (2.5)$$

$$e_k \sim N(0, \sigma_k^2) \quad (2.6)$$

$$(2.7)$$

der  $\bar{X}$  er gjennomsnittet av  $X$ .

Segmenteringsalgoritmen begynner med høy-dimensjonal minste kvadraters metode (HRLS) for å estimere systemmatrisen  $A$ . Dette gir et tidsvarierende estimat  $\hat{A}_k$ . For å måle forandringen i dynamikk beregnes avstanden mellom de to siste systemmatrisene:

$$A_{d_k} = A_k - A_{k-1} \quad (2.8)$$

En skalar tidsserie,  $a_k$ , regnes ut ved å ta Mean Square Error(MSE) av  $A_{d_k}$ .  $a_k$  modeleres ved hjelp av AR-modellen:

$$a_k = c_1 a_{k-1} + e_k \quad (2.9)$$

der  $c_1$  estimeres ved bruk av rekursiv minste kvadraters metode. Segmenteringspunktene finnes ved å bruke toppunkter i tidsserien  $a_k$ , som blir funnet ved høypassfiltrering.

Etter segmentering betraktes leddvinklene hver for seg. En andre ordens AR-modell tilpasses tidsseriene for hver leddvinkel. Løsningstrajektoren for hver leddvinkel vil være beskrevet av samplingspunkter på ligningen:

$$x(t) = \alpha e^{-\beta t} \cos(2\pi f t + \phi) + T \quad (2.10)$$

der  $\alpha, \beta, f, \phi$  og  $T$  henholdsvis beskriver amplitude, dempningsrate, frekvens, fase, og bias. Dette blir brukt for å beskrive dataene på en kompakt måte:

$$[\alpha_{ij} \quad \beta_{ij} \quad f_{ij} \quad T_{ij}]^T \quad (2.11)$$

der  $1 \leq i \leq n, 1 \leq j \leq s$ , og  $n$  er antall leddvinkler og  $s$  er antall segmenter. Parameterne  $f$  og  $\beta$  regnes ut i fra den andre ordens AR-modellen, mens de resterende parameterne blir funnet ved å minimere avviket mellom ligning 2.10 og måledataene. For å sammenligne forskjellige bevegelser introduseres en metrikk basert på kinetisk energi, som regnes ut i fra systemparameterne.



I J.G.Barton & A.Lees (1995) læres et nevralt nettverk opp til å skille mellom tre forskjellige type gange. De tre typene består av normal gange, en simulering av forskjell i benlengde, og en simulering av en forskjell i benvekt. Målte parametere er hofte- og knevinkler. Fast Fourier Transform (FFT) blir brukt som egenskapsuttrekking, og bare de lavere frekvensene blir brukt i klassifiseringen. Totalt brukes 30 FFT-koeffisienter, 15 for hofte-vinkler og 15 for kne-vinkler. Hver koeffisient representerer et nevron i inputlaget til det nevralt nettverket. Nettverket består av fire lag der de 2 skjulte lagene består av henholdsvis 5 og 4 nevroner. Vektene til synapsene i nettverket bestemmes ved back-propagation-metoden. Totalt 24 mønstre brukes til trening og testing, og ved å dele mønstrene inn i 18 for trening og 6 for testing, fås 4 kombinasjoner. Resultatene som presenteres viser at nettverket klassifiserer 20 av 24 mønstre korrekt.

PCA (se seksjon 3.2.1) kan være en nyttig metode for å beskrive menneskelig bevegelse med få parametere, og er således en metode for egenskapsuttrekking. PCA har vært brukt for å beskrive gripebevegelser, skriving på tastatur, leppebevegelser og gange (Sanger 2000). I alle tilfellene kan den totale bevegelsen bli tilnærmet beskrevet av et lite antall prinsipale komponenter. I Sanger (2000) brukes PCA for å beskrive armbevegelser. Prøvepersoner beveger enden av en to-ledds manipulator, og prøver å kopiere en gitt måltrajektor. Etter andre forsøk blir måltrajektoren satt til gjennomsnittet av de to forrige. I løpet av en slik bevegelse blir det registrert 50 x- og y-koordinater med konstant tid mellom hvert punkt. Koordinatene samles i en vektor  $m_f$ :

$$m_f = [m_x(1), \dots, m_x(50), m_y(1), \dots, m_y(50)] \quad (2.12)$$

der  $m_f$  betyr målevektoren for forsøk nummer  $f$ . Det blir så brukt PCA med  $m_f$  som observasjonsvektor. Det vises at trajektorene konvergerer mot en lineær sammensetning av de første prinsipale komponentene.

En lignende analyse blir gjort i Fod et al. (2002). Her er armbevegelserne i 3 dimensjoner, og beskrives av fire leddvinkler: the shoulder flex extend, shoulder adduction abduction, humeral rotation og elbow rotation. Det presenteres to forskjellige algoritmer for å segmentere bevegelsen. Begge algoritmene baserer seg på vinkelhastighetene til alle leddvinklene. Den første algoritmen registrerer når vinklene skifter retning, og et segmenteringspunkt defineres der 2 leddvinkler skifter retning i løpet av  $300ms$ . Den andre algoritmen betrakter summen av vinkelhastigheten til alle leddvinkler:

$$z = \dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2 + \dot{\theta}_4^2 \quad (2.13)$$

Et segmenteringspunkt defineres der  $z$  er under en nedre grense. Etter segmentering blir det foretatt en PCA-analyse på dataene. Hvert segment blir redusert til en lengde på 400 (4 vinkler ganger 100 tidspunkter), og alle segmentene blir brukt som observasjonsvektor i en PCA-analyse. Ved å først projisere dataene ned på et gitt antall prinsipale komponenter, for å så projisere de tilbake, fås en 400-dimensjonal vektor slik at bevegelsen blir rekonstruert. Simuleringer blir foretatt med en humanoid robot.

I Wu et al. (2007) presenteres klassifisering av gange mellom unge og eldre. På grunnlag av data fra et målesystem blir 36 forskjellige variabler definert, samlet i en vektor

$x_k \in \mathbb{R}^{36}$ , der  $k$  representerer ulike skritt.  $x_k$  representerer her kinematiske egenskaper, eller egenskaper knyttet til posisjon i rommet til en gitt tid, for skritt nummer  $k$ . For å redusere dimensjonen på dataene blir det brukt kjernebasert prinsippal komponentanalyse (KPCA), med  $x_k$  som observasjonsvektor. KPCA er en ulineær egenskapsuttrekningsmetode, hvor ulineæriteten består i at det opprinnelige datamaterialet blir mappet til et rom med høyere dimensjon. Vanlig lineær PCA blir så brukt på den transformerte datamengden. Det blir argumentert for at KPCA er godt egnet til å beskrive den ulineære strukturen til menneskelig gange. Den ulineære mappingen kan skrives som:

$$\Phi : \mathbb{R}^N \rightarrow F \quad (2.14)$$

Som vi vil se i seksjon 3.2.1 er prinsippalkomponentene egenvektorene til kovariansmatrisen til observasjonsvektorene. I beregningen av kovariansmatrisen i  $F$  brukes prikkprodukt, og måten det beregnes på avgjøres av kjernefunksjonen  $K$ :

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.15)$$

Kjernefunksjoner som brukes i Wu et al. (2007) er:

$$K(x_i, x_j) = ((x_i \cdot x_j) + 1)^d \quad (2.16)$$

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (2.17)$$

Selve beregningen av prinsippale komponenter blir nå litt mer komplisert, se Wu et al. (2007). Klassifiseringen blir gjort med et nevralt nettverk som blir trent opp med support vector machines(SVM)-algoritmen. Testen består av 24 friske unge personer og 24 friske eldre. Det vises at KPCA i kombinasjon med SVM, gir bedre resultater enn lineær PCA og SVM. Best resultat ble oppnådd når alle variablene ble brukt, både kinematiske og de knyttet til romlig posisjon. Dette indikerer at forskjellige typer data-variabler kan bedre klassifiseringen av menneskelig bevegelse.

PCA kan også brukes til segmentering. I Barbic et al. (2004) presenteres det en metode som baserer seg på projeksjonsfeil. Ved å betrakte et gitt antall prinsippale komponenter, defineres et segment der projeksjonsfeilen stiger mye.

Det er verdt å studere i litt mer detalj hvordan PCA brukes i artiklene nevnt ovenfor. I PCA kan man skille mellom observasjoner og variabler. Hver observasjon vil inneholde en eller flere variabler, se seksjon 3.2.1. Målet med PCA er å redusere dimensjonen på dataene, slik at alle observasjonsvektorene kan beskrives med en lavere dimensjon.

I Barbic et al. (2004) inneholder hver observasjon kun data fra et tidspunkt. Hver observasjon inneholder variabler som representerer kinematiske egenskaper på et gitt tidspunkt. Ønsket er å finne ut hvilke variable som er viktige for å kunne beskrive de aktuelle bevegelsene.

I Sanger (2000) består hver observasjon av 100 variabler, der variablene representerer  $x$ - og  $y$ -verdier på 50 forskjellige tidspunkter. I Fod et al. (2002) blir PCA brukt på samme

måte, der hver observasjon har dimensjon 400 og består av verdien på 4 forskjellige variable i 100 forskjellige tidspunkter. Ved PCA-analyse får man like mange egenvektorer som man har variable. I tilfellene ovenfor får man dermed henholdsvis 100 og 400 egenvektorer med tilhørende egenverdier. En observasjon svarer her til et gitt forsøk. Merk at i Sanger (2000) og Fod et al. (2002) er tiden en varierende parameter i observasjonsvektoren. I praksis betyr dette at en vektor som representerer målinger i mange forskjellige etterfølgende tidspunkter approksimeres med en mye lavere dimensjon. At det går an å gjøre en reduksjon i dimensjon på denne måte kan blant annet forklares med at to nærliggende målevariable på tidsaksen vil være svært korrelerte.

Det er ikke noe i veien for å bruke begge metodene ovenfor på et gitt problem, og en slik fremgangsmåte presenteres i R.Das et al. (2006). Målet er her å gjenkjenne ulike type gange: gåing, jogging, løping og halting. PCA foretas i 2 trinn der det første trinnet består i å redusere dimensjonen på et 6-dimensjonalt datasett, bestående av tidsserier av 6 forskjellige variable som representerer vinkler og vinkelhastigheter tilhørende lår og legger. Hver observasjon består dermed av 6 variable, og hver observasjon projiseres ned på to prinsipale komponenter. Dette resulterer i 2 tidsserier, en for hver prinsipale komponent. Dette første steget er en måte for å redusere dimensjonen på dataene på, som tar vare på de kinematiske variablene som står for mest av variansen i datasettet. Hver av de to tidsseriene blir så delt opp i enkelte “gait cycles”, da hver tidsserie opprinnelig består av mange slike. For hver av de to tidsseriene, blir hver cycle betraktet som observasjoner i det andre trinnet i PCA analysen. Ved å så projisere hver enkelt gait cycle ned på de tre prinsipale komponentene, fås en egenskapsvektor i  $\mathbb{R}^3$  som brukes til klassifisering. Det oppnåes en klassifiseringsrate på over 90%. Den siste PCA-analysen, foretatt med en observasjonsvektor som består av en parameter endret over tid, trekker ut viktig informasjon fra en tidsserie. Det kan sammenlignes med å foreta FFT-analyse eller tilpasse en dynamisk modell. Sistnevnte metode er hva vi vil gjøre i dette arbeidet.

Slike dynamiske modeller, i form av AR-modeller, er blitt brukt i klassifisering av EMG-signaler. Ved å modellere signalet som en AR-prosess, representeres hele signalet med noen få parametere, som lett kan bli behandlet av en klassifikator. I F.Kelly et al. (1990) klassifiseres fire forskjellige armbevegelser ved hjelp av EMG-signaler. Signalene tilpasses en fjerde ordens AR-modell. Parameterne i denne modellen finnes ved bruk av Hopfield-nettverk, som vises å være raskere enn sekvensiell minste kvadraters metode. Den første AR-koeffesienten, sammen med signaleffekten, blir brukt som innganger til et feed-forward nevralt nettverk. Nettverket som blir brukt har 3 lag og 4 noder i det skjulte laget og blir trent ved bruk av back-propagation-algoritmen.

En lignende analyse er gjort i de Oliveira Andrade & Soares (2007). Også her brukes en AR-modell til å modelere signalene, og parameterne i modellen blir funnet ved hjelp av minste kvadraters metode. AR-parameterne blir så brukt som innganger til et kunstig nevralt nettverk, basert på back-propagation.

Hidden Markov Models(HMM) er den enkleste formen for dynamiske baysianske nettverk(DBN),

og har vært mye brukt til å klassifisere menneskelig bevegelse. I Biswas & Quwaider (2008) brukes HMM til å skille mellom positurene sitte og stå. Data samles inn av fire radiosendere/sensorer som er festet til kroppen, to på overarmene og to på lårene. Hovedprinsippet er at sensorene er nærmere hverandre når man sitter. Ved å summere målestyrken i de ulike sensorene, fåes et mål på hvor nærme sensorene er hverandre. Dette signalet utgjør observasjonen i hidden Markov-modellen, og målet er å finne ut hvilke av de to tilstandene, sitte eller stå, nettverket er i. Resultatene som blir presentert viser at HMM er mer effektivt enn en enkel metode basert på terskelverdier.

I Nam & Wohn (1997) brukes HMM til å segmentere og gjenkjenne armbevegelser som en del av et system for å gjenkjenne håndbevegelser. Vinkel på fingre og absolutt posisjon i rommet måles, og brukes som observasjonsvektor til hidden Markov-modellen. En Hidden Markov model blir trent for hver type håndbevegelse, og gjenkjenning blir foretatt med Viterbi søkealgoritmen.

Hidden Markov-modeller har også vært brukt i klassifisering av gange. I Meyer (1997) trenes hidden Markov-modeller opp til å skille mellom vanlig gange, løping, hopping og halting. Det blir presentert to forskjellige måter for å hente ut en egenskapsvektor fra video-data. Den ene baserer seg på trajektorer og den andre fra optiske strømningsfelt. I det førstnevnte tilfellet består egenskapsvektoren av forflytning i x og y- retning for hodet, høyre ben og høyre fot. Egenskapsvektoren brukes videre som observasjonsvektor i hidden Markov-modellen.

Klassifisering av gange trenger ikke skje via videodata. I Chen et al. (2007) samles data inn ved hjelp av et akselerometer, et gyroskop, og kraftmålere som er festet til en sko. Hensikten med målingene er å skille mellom vanlig gange, gange med føttene innover, og gange med føttene utover. Dataene fra sensorene behandles først ved bruk av FFT, og den endelige egenskapsvektoren blir konstruert ved bruk av PCA. Hver type gange blir så modelert som en HMM med 6 noder etter hverandre, der hver node har overgang til seg selv, noden foran eller noden 2 hakk frem. Baum-Welch algoritmen brukes til å trene nettverkene og gjenkjenning/klassifisering skjer ved forward-backward-algoritmen.

HMM har også vært brukt i bevegelsesanalyse hos dyr. I Pfau et al. (2008) brukes hidden Markov-modeller til å klassifisere bevegelse for veddeløpshester ved bruk av en 6-DOF sensor inn i de to klassene galopp og ikke-galopp.

En annen type DBN er switching linear dynamic systems (SLDS). I Pavlovic & Rehg (2000) brukes SLDS til å tracke og klassifisere vanlig gange og jogging. Prinsippet bak SLDS er at en ulineær prosess beskrives ved å skifte mellom ulike lineære dynamiske systemer over tid. Totalsystemet kan beskrives av to delsystem:

$$x_{k+1} = A(s_{k+1})x_k + v_{k+1}(s_{k+1}) \quad (2.18)$$

$$y_t = Cx_k + w_k \quad (2.19)$$

$$x_0 = v_0(s_0) \quad (2.20)$$

er den fysiske modellen og

$$P(s_{k+1} = i | s_k = j) = \Pi(i, j) \quad (2.21)$$

$$P(s_0 = i) = \pi_0(i) \quad (2.22)$$

er modellen for svitsjing mellom de lineære systemene.  $x_k$  er den skjulte tilstanden til systemet, og  $y_k$  er observasjonsvektoren.  $v_t$  og  $w_t$  er henholdsvis normalfordelt prosess- og målestøy med varians  $Q$  og  $R$ .  $s_t$  er tilstanden for svitsjemodellen og  $\Pi(i, j)$  er en matrise som angir sannsynligheter fra å gå fra en svitsjetilstand til en annen.  $\pi_0(i)$  angir initialfordelingen til  $s_0$ . Nettverket må trenes for å finne parameterne  $A$ ,  $C$ ,  $Q$ ,  $R$ ,  $\Pi$  og  $\pi_0$ . Observasjonsvektoren består av rotsjonsvinkler for legger, kropp (torso) og hode. Det presenteres 3 forskjellige metoder for å finne parameterne, og disse metodene blir brukt for å trene opp nettverk for vanlig gange og løping. I tillegg trenes også en HMM. Det blir vist at SLDS fungerer bedre for klassifisering enn hidden Markov-modellen.



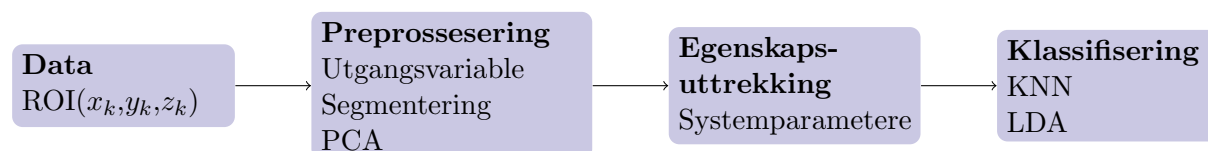
# Kapittel 3

## Metode

I dette kapitlet vises det hvordan dynamisk systemteori og modelltilpasning kan brukes for å klassifisere bevegelser som normale eller avvikende. Vi bruker måledata fra sensorene til å identifisere dynamiske modeller. En slik modelltilpasning kan sees på som en form for egenskapsuttrekking, der systemparameterne tilsvarer en egenskapsvektor. Figur 3.1 illustrerer fremgangsmåten brukt i denne oppgaven for å klassifisere spedbarnsbevegelser. Vi starter med en samling av måledata. Ut i fra måledata, konstrueres utgangsvARIABLE, som er direkte basert på sensordata eller PCA på sensordata. Utgangsvareblene blir så brukt til å tilpasse dynamiske modeller ved bruk av systemidentifikasjon. Til slutt blir systemparameterne brukt til klassifisering.

### 3.1 Måledata

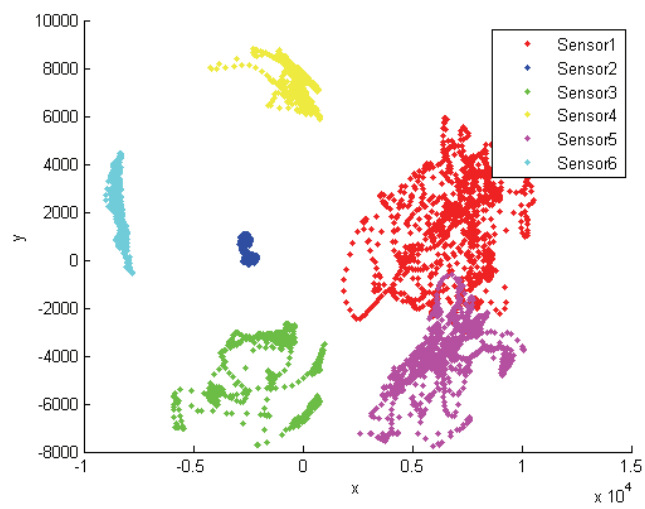
I denne oppgaven brukes bevegelsesdata fra 81 spedbarn som på forhånd har vært innsamlet. Bevegelserne til spedbarna ble registrert ved hjelp av 6 sensorer. Sensorene var plassert på håndledd, ankler, bryst og hode, se Tabell 3.1. Hver sensor måler plassering i rommet med koordinatene  $x, y$  og  $z$ . I tillegg måler hver sensor rotasjon, representert i måledataene som et kvaternion, det vil si 4 parametere. Data ble samlet inn med en samplingsrate på 25Hz. I Figur 3.3 vises sensorplassering og definisjon av akser og i Figur 3.4 illustreres det at z-aksen er positiv oppover.



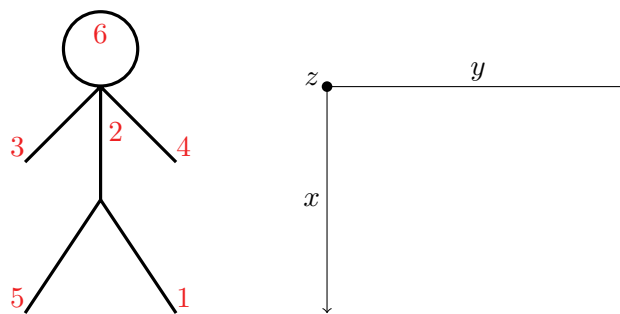
Figur 3.1: Metode for klassifisering av spedbarnsbevegelser

Sensor nr.	Plassering
Sensor 1	Venstre ankel
Sensor 2	Bryst
Sensor 3	Høyre håndledd
Sensor 4	Venstre håndledd
Sensor 5	Høyre ankel
Sensor 6	Hode(panne)

Tabell 3.1: Sensorplassering

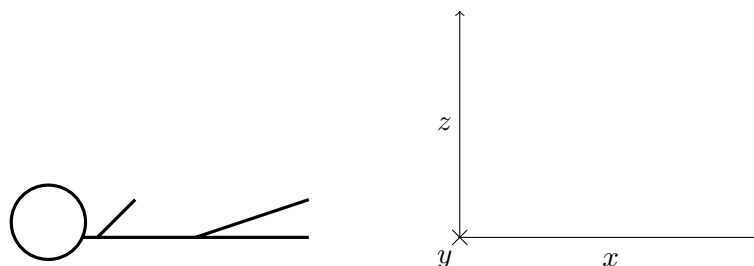


Figur 3.2: Måleverdi på forskjellige sensorer



Figur 3.3: Koordinatsystem og sensorplassering





Figur 3.4: Koordinatsystem

For hvert spedbarn er det blitt samlet inn en eller flere opptak med bevegelsesdata. En fysioterapeut har så plukket ut interessante tidsperioder i disse opptakene, kalt Region Of Interest (ROI). Hvert opptak vil dermed bestå av et eller flere ROI'er. I Figur 3.2 vises måleverdier for alle 6 sensorer i  $xy$ -planet for en enkelt ROI. All måledata som er blitt brukt i denne oppgaven er i form av ROI'er fra spedbarn med normale og avvikende bevegelser. En ROI inneholder tidsserier med data fra alle 6 sensorer, bestående av måleverdier i  $x$ ,  $y$  og  $z$ -rening, samt rotasjonsdata. ROI'ene er av ulik lengde, og varierer fra 25 sekunder til over 15 minutter. Som tidligere nevnt er samplingsraten  $25Hz$ , så en ROI på 25 sekunder, vil inneholde 625 observasjoner, der hver observasjon representerer måleverdier for alle akser og rotasjoner for de 6 sensorene.

## 3.2 Segmentering og valg av utgangsvARIABLE

En dynamisk modell skal beskrive et system ved å gi en sammenheng mellom inngangen (pådraget) og utgangen til systemet. For å kunne tilpasse en dynamisk modell til måledataene må vi velge hva som skal være utgangsvARIABLE for systemet, og hvor lange tidsserier med data som skal betraktes om gangen. UtgangsvARIABLE representerer tidsserier som beregnes ut i fra måledata, som vi har sett er gitt i form av ROI'er. I dette arbeidet har det blir fokusert på å bruke  $x$ ,  $y$  og  $z$ -koordinatene i ROI'ene. Vi vet ikke på forhånd hvilke utgangsvARIABLE som best representerer den ønskede dynamikken, det vil si den dynamikken som skiller normale bevegelser fra avvikende bevegelser. Valg av utgangsvARIABLE er i denne oppgaven ikke gjort ut i fra kunnskap om spedbarnsbevegelser, men ønsket om å:

- Se på viktigheten av hver enkelt sensor for seg
- Se om noen akseretninger er viktigere enn andre. Dette er for eksempel relevant hvis måledata skal komme fra videodata, der spedbarnet filmes ovenifra og man kun betrakter bevegelse i horisontalplanet.

- Kombinere flere sensorer. Vi kan velge å ha en modell med flere utganger, eller så kan vi beregne en utgangsvariabel basert på flere sensorer (som for eksempel avstanden mellom to sensorer).

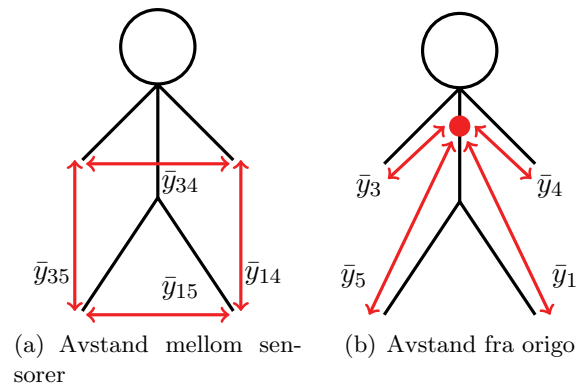
I denne oppgaven er utgangsvariable beregnet ut i fra måledata fra sensorene, men noe eksplisitt pådrag finnes naturligvis ikke. Vi vil derfor anta at utgangsvariablene representerer stokastiske systemer drevet av hvit støy.

Et naturlig valg er å velge  $x$ ,  $y$  og  $z$ -koordinatene fra sensorene som utgangsvariable. Siden vi har 3 akser for hver av de 6 sensorene gir dette 18 mulige variable. En annen mulighet er å betrakte avstander mellom forskjellige sensorer. Dette kan også gjøres på mange forskjellige måter, f.eks kan man velge å se på avstand i rommet mellom to sensorer, eller avstand kun langs en av aksene. På denne måten kan data fra flere sensorer kombineres samtidig som vi fortsatt bare har en utgang på systemet. Tabell 3.2 viser notasjon for utgangsvariable som har blitt brukt i denne oppgaven, og Figur 3.5 illustrerer hva de forskjellige variablene måler. Utgangsvariable i fet skrift betyr en vektor med flere utganger. Med  $\mathbf{y}_{s_i}$  som utgang betrakter vi to akser for en sensor, mens i tilfellet med  $\mathbf{y}_{s_i s_j}$  som utgang betrakter vi 2 forskjellige sensorer langs en felles akse ( $x, y$  eller  $z$ ). I denne oppgaven er alle utgangsvariable blitt normalisert og skalert slik at de varierer rundt 0.

Måleverdiene som er registrert av sensorene vil være gitt ut i fra et punkt i rommet som ikke er et bestemt punkt på spedbarnet. En mulighet er å betrakte brystet som origo for alle måledata der dette er relevant (betraktes avstand mellom sensorer spiller ikke origo noen rolle).

Som nevnt i forrige avsnitt er all data gitt i form av ROI'er. En mulighet er å bruke hver enkelt ROI som datamateriale for beregning av utgangsvariable og påfølgende systemidentifikasjon. Dette medfører at tidsseriene som sendes inn til systemidentifikasjonsalgoritmen vil være av ulik lengde, og vi vil få en dynamisk modell per ROI. Siden bevegelsene fra spedbarn er komplekse og varierte, vil noe av denne informasjonen kunne gå tapt hvis vi betrakter en hel ROI om gangen. Det kan derfor være interessant å dele opp hver ROI i segmenter med fast lengde, for eksempel i 30 sekunders intervaller. Hver ROI vil da bli delt opp i et antall mindre deler, heretter kalt mini-ROIer, og vi vil få en dynamisk modell per mini-ROI.

Når det gjelder preprossesering av data, hadde det vært mulig å behandle dataene med et lavpassfilter, for å filtrere ut målestøy. Dette er ikke gjort i dette arbeidet.



Figur 3.5: Forskjellige valg av utgangsvARIABLE

Variabel	Betydning
$y_{s_a}$	Sensor $s$ , akse $a$
$\bar{y}_{s_i}$	Avstand fra sensor $s_i$ til origo
$\bar{y}_{s_i k}$	Avstand fra sensor $s_i$ til origo langs plan $k$ ( $xy$ , $xz$ , $yz$ -planet)
$\bar{y}_{s_i s_j}$	Avstand mellom sensor $s_i$ og sensor $s_j$
$\bar{y}_{s_i s_j k}$	Avstand mellom sensor $s_i$ og sensor $s_j$ i retningen $k$
$\mathbf{y}_{s_i k}$	Sensor $s_i$ i $k$ -planet ( $xy$ , $xz$ , $yz$ -planet)
$\mathbf{y}_{s_i s_j k}$	Sensor $s_i$ og sensor $s_j$ langs aksene $k$

Tabell 3.2: UtgangsvARIABLE

### 3.2.1 PCA for reduksjon i dimensjon på måledata

PCA er en måte å redusere dimensjonen i dataene på. Utgangspunktet er at vi observerer en rekke variabler i flere observasjoner, se uttrykk 3.1.

$$X = \left. \begin{array}{c} \overbrace{\left[ \begin{array}{ccc} x_{11} & \cdots & x_{1j} \\ \vdots & \cdots & \vdots \\ x_{N1} & \cdots & x_{ij} \end{array} \right]}^{j \text{ variable}} \\ \end{array} \right\} N \text{ observasjoner} \quad (3.1)$$

Hver rad tilsvarer en observasjon og hver kolonne representerer en variabel. Kovariansmatrisen til de  $N$  observasjonene  $x_i$  er gitt av

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T \quad (3.2)$$

der  $x_i = [x_{i1} \dots x_{ij}]^T$  og  $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$

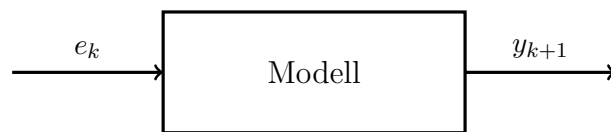
Hvis vi ser på egenvektor- og egenverdiene til  $C$ , så vil egenverdien representere variansen vi har i retningen til egenvektoren. Vi har også at egenvektoren til den største egenverdien er den retningen med størst varians. Denne egenvektoren kalles den prinsipale komponenten. Et antall  $k$  med prinsipalkomponenter tilsvarer de  $k$  egenvektorene som tilhører de  $k$  høyeste egenverdiene. Prinsipalkomponentene står alltid vinkelrett på hverandre.

Dimensjonen på dataene reduseres ved å projisere dataene ned på hyperplanet spent ut av de  $k$  største egenvektorene ( $k$  prinsipale komponenter):

$$X_p = (E^T \cdot \hat{X}^T)^T \quad (3.3)$$

der  $E = [v_1 \dots v_k]$  er en matrise med  $k$  prinsipale komponenter som kolonner,  $\hat{X}$  tilsvarer  $X$  der hver observasjonsvektor er justert for gjennomsnittet, og  $X_p \in \mathbb{R}^{N \times k}$ .

I denne oppgaven er PCA brukt for å beregne utgangsvARIABLE langs dimensjonen/dimensjonene med mest varians. Prinsipalkomponenter er blitt beregnet ved å betrakte alle ROI'er fra både den normale og den avvikende gruppen som observasjonsvektorer. ROI'ene er i dette tilfellet hentet fra treningssettet, se seksjon 3.5. I PCA-beregningene består hver observasjon av variablene  $x$ ,  $y$  og  $z$ , så PCA-analysen i dette tilfellet tilsvarer projisering ned på en linje eller et plan, avhengig om vi bruker 1 eller 2 prinsipalkomponenter. Prinsipalkomponenter kunne også ha vært beregnet for hver ROI, eller for hver pasient. Hvis spedbarna viser seg å ligge skjevt i forhold til hverandre, kunne dette være en aktuell metode. Det kan også være fornuftig ut i fra tanken at det er dynamikken langs en retning i rommet som er viktig, og ikke retningen i seg selv. I dette arbeidet er det antatt at spedbarna ligger forholdsvis i samme retning, og at det derfor gir mening



Figur 3.6: Modellen har inngang  $e_k$  og utgang  $y_k$

å se på prinsipalkomponenter beregnet fra samtlige ROI'er. Der  $\mathbf{y}_{\text{sis}_{j_k}}$  er utgangsvektor, beregnes prinsipale komponenter separat for hver sensor.

### 3.3 Modell

I denne seksjonen presenteres forskjellige typer dynamiske modeller som kan brukes for å beskrive spedbarnsbevegelser. Vi vil anta en lineær modell, siden vi ikke har noen fysiske modeller å gå ut i fra som skulle tilsi noe annet. Det har blitt fokusert på kjente lineære modeller som er mye brukt sammen med systemidentifikasjon. Som nevnt i seksjon 3.2, vil vi betrakte en dynamisk modell drevet av hvit støy, se Figur 3.6.

#### 3.3.1 Modellstruktur

En dynamisk modell kan ha en eller flere utganger, kalt henholdsvis Single Input Single Output(SISO) og Multiple Input Multiple Output(MIMO). I vårt tilfelle vil dimensjonen på støyvektoren tilsvare dimensjonen på utgangen. La oss i første omgang se på SISO-tilfellet. En enkel løsning er å modellere systemet som en Autoregressive Moving Average(ARMA)(p,q)-prosess:

$$y_n = \sum_{k=1}^p a_k y_{n-k} + \sum_{k=0}^q b_k e_{n-k} \quad (3.4)$$

Her er  $y_n$  en utgangsverdi i tidspunkt  $n$ , mens  $e_n$  er verdien på støyen. Ved å sette  $b_0 = 1$ , og  $b_k = 0$  for  $k > 0$  fåes en AR( $p$ )-modell.

$$y_n = \sum_{k=1}^p a_k y_{n-k} + e_n \quad (3.5)$$

Ved å sette  $a_k = 0$  for  $k \geq 1$  fåes en MA( $q$ )-modell

$$y_n = \sum_{k=1}^q b_k e_{n-k} \quad (3.6)$$

Det er naturlig å anta at en utgangsverdi i tidspunkt  $k$  er avhengig av tidligere utgangsverdier, og derfor er det lite trolig at en MA( $q$ )-modell gir en fornuftig modellering.

Vi kan også anta en modell som tar hensyn til flere utgangsverdier samtidig. En mulighet er en AR-modell med flere utganger:

$$y_n = \sum_{k=1}^p A_k y_{n-k} + e_n \quad (3.7)$$

der  $y_n \in \mathbb{R}^i$ ,  $A_k \in \mathbb{R}^{i \times i}$  og  $e_n \in \mathbb{R}^i$ .

En tredje mulighet er å bruke en tilstandsrommodell:

$$x_{k+1} = Ax_k + v_k \quad (3.8)$$

$$y_k = Cx_k + w_k \quad (3.9)$$

$$v_k \sim N(0, Q), \quad w_k \sim N(0, R) \quad (3.10)$$

der  $A \in \mathbb{R}^{n \times n}$  and  $C \in \mathbb{R}^{m \times n}$ .

Tilstandsrommodellen kan brukes både for SISO- og MIMO-modeller. Hvis denne type modell brukes som utgangspunkt i systemidentifikasjon, vil ikke tilstanden  $x$  ha noen fysisk tolkning.

At en ARMA-modell kan uttrykkes som en tilstandsrommodell og omvendt kommer frem hvis vi betrakter modellene i  $z$ -domenet. ARMA-modellen transformeres enkelt til  $z$ -domenet ved å betrakte  $z$  som en skifteoperator i tid:

$$y(z) = \frac{\sum_{j=0}^q b_j z^{-j}}{\sum_{j=0}^p a_j z^{-j}} e(z) \quad (3.11)$$

der  $a_0 = b_0 = 1$ .

Fra systemteori vet vi at tilstandsrommodellen i 3.10 kan uttrykkes i  $z$ -domenet som:

$$H(z) = C(zI - A)^{-1} \quad (3.12)$$

der den vanlige  $B$ -matrisen nå er identitetsmatrisen. En gitt  $z$ -funksjon vil kunne ha mange tilstandsrom-representasjoner og en ARMA-representasjon. Nøyaktig hvordan man går frem og tilbake mellom disse representasjonene er forklart i Henriksen (1998). ARMA- og tilstandsrom-modellen kan sies å være ekvivalente siden vi kan gå fra den ene representasjonen til den andre. Valg av modellstruktur vil uansett påvirke den endelige klassifiseringen, da det finnes ulike systemidentifikasjonsalgoritmer for de ulike modellstrukturene.

### 3.3.2 Analyse av modell

ARMA-modellen i ligning 3.11 kan skrives på formen:

$$H(z) = c \frac{\prod_{i=1}^q (1 - \beta_i z^{-1})}{\prod_{i=1}^q (1 - \alpha_i z^{-1})} \quad (3.13)$$

der  $\alpha_i$  og  $\beta_i$  er henholdsvis poler og nullpunkter til systemfunksjonen  $H(z)$ . For rene AR-modeller vil vi kun få poler, i tillegg til nullpunkter i origo. Formen i ligning 3.13 er interessant for å analysere systemfunksjonen i frekvensdomenet. I frekvensdomenet betrakter vi systemet som et filter, som filtrerer hvit støy. Fra lineær systemteori vet vi at poler og nullpunkter godt beskriver filteregenskapene til  $H(z)$ . Det vil derfor være interessant å gjøre en analyse av polenes plassering for modeller tilpasset både normale og avvikende bevegelser.

I tillegg til å studere polene til systemet kan vi se på effekt-tetthetsspekteret, som sier hvordan effekten i signalet er fordelt på de ulike frekvensene. Effekt-tetthetsspekteret til en stokastisk prosess kan uttrykkes ved systemfunksjonen  $H(z)$  og effekt-tetthetsspekteret til inngangssignalet:

$$\Gamma_{yy}(\omega) = |H(\omega)|^2 \Gamma_{xx}(\omega) \quad (3.14)$$

Inngangen er i vårt tilfelle hvit støy med varians  $\sigma^2$ , det vil si  $\Gamma_{xx}(\omega) = \sigma^2$ . Effekt-tetthetsspekteret til utgangen blir dermed:

$$\Gamma_{yy}(\omega) = |H(\omega)|^2 \sigma^2 \quad (3.15)$$

Effekt-tetthetsspekteret kan også brukes til å validere modellene vi finner ved systemidentifikasjon. Ligning 3.15 representerer en parametrisk metode for å regne ut spekteret på, det vil si at man først tilpasser en dynamisk modell til dataene, for så å regne ut spekteret ved bruk av systemparameterne. Metoden finnes i **MATLAB** som funksjonen `ar-cov`. Spekteret kan også beregnes uten å gå veien om en modell, og da snakker vi om ikke-parametriske metoder for spekterestimering. Periodogrammet er et slikt estimat, og kan regnes ut som følger:

$$P_{xx}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-j2\pi f n} \right|^2 = \frac{1}{N} |X(f)|^2 \quad (3.16)$$

Effekt-tetthetsspekteret for stokastiske prosesser er Fourier-transformen til autokorrelasjonen. Periodogrammet utledes ved å ta Fouriertransformen av et estimat av autokorrelasjonen, se Proakis & Manolakis (1996) side 905. Periodogrammet kan beregnes i **MATLAB** ved hjelp av funksjonen `periodogram`.

### 3.3.3 Orden på AR-modell

#### Autokorrelasjon-kriterium

Etter å ha tilpasset en modell til dataene, er det interessant å se hvor godt modellen passer. For AR-modeller kan dette gjøres ved å se på residualene til modellen. Residualet i et tidspunkt  $k$  regnes ut i fra:

$$\omega_k = y_k - \sum_{i=1}^p a_i y_{k-i} \quad (3.17)$$

der  $p$  er ordenen og  $a_i$  er AR-parameterne.

I følge modellen skal residualene representere den underliggende hvite støyen som driver AR-prosessen, og de skal dermed være ukorrelerte. For å undersøke om residualene representerer hvit støy kan autokorrelasjonen til residualene studeres. I følge Brockwell & A.Davis (2001) skal en slik sekvens approksimativt være normalfordelt med  $N(0, 1/n)$ . Hvis  $e_n$  er hvit støy bør dermed 95% av verdiene falle innenfor  $\pm 1.96/\sqrt{n}$ .

#### AIC og FPE

To kriterier for å velge orden på en AR-modell er presentert i Proakis & Manolakis (1996). Det ene heter *final prediction error criterion* (FPE) og ordren som velges minimerer kriteriet:

$$FPE(p) = \hat{\sigma}_{wp}^2 \left( \frac{N + p + 1}{N - p - 1} \right) \quad (3.18)$$

der  $\sigma_{wp}^2$  er variansen til prediksjonsfeilen,  $p$  er orden og  $N$  er antall datapunkter. I det andre kriteriet, *Akaike information criterion* (AIC) minimeres funksjonen:

$$AIC(p) = \ln(\hat{\sigma}_{wp} + 2p/N) \quad (3.19)$$

Vi ser at begge kriteriene straffer høy orden. Verdien for de to kriteriene for en gitt orden kan i MATLAB beregnes via funksjonene `fpe` og `aic`.

## 3.4 Systemidentifikasjon

Denne seksjonen presenterer algoritmer for systemidentifikasjon for ARMA og tilstand-srommodeller, som er brukt i denne oppgaven. Systemidentifikasjon er i hovedsak blitt implementert med funksjoner fra MATLAB sin `System Identification Toolbox`.



### 3.4.1 Minste kvadraters metode

Minste kvadraters metode brukes for AR-modeller og beregnes i `MATLAB` ved hjelp av `ar` med `ls` eller `fb` som tilleggsparemer. Metodene `'ls'` og `'fb'` er begge varianter av minste kvadraters metode. Den førstnevnte er den vanlige minste kvadraters metode der kriteriet som minimeres er kvadratet av prediksjonsfeilen for forovermodellen:

$$\epsilon = \sum_{n=p}^{N-1} |x(n) + \sum_{k=1}^p a_p(k)x(n-k)|^2 \quad (3.20)$$

Metoden `'fb'` tar hensyn til prediksjonsfeil til både bakover og forovermodellen:

$$\epsilon = \sum_{n=p}^{N-1} \left( |x(n) + \sum_{k=1}^p a_p(k)x(n-k)|^2 + |x(n-p) + \sum_{k=1}^p a_p^*(k)x(n+k-p)|^2 \right) \quad (3.21)$$

Begge kriteriene fører til et sett med lineære ligningsett som må løses.

### 3.4.2 Yule-walker

Yule-walker baserer seg på sammenhengen mellom systemparameterne i en AR-modell og autokorrelasjonen (Proakis & Manolakis 1996):

$$\gamma_{xx}(m) = - \sum_{k=1}^p a_k \gamma_{xx}(m-k) \quad m > q \quad (3.22)$$

$$- \sum_{k=1}^p a_k \gamma_{xx}(m-k) + \sigma_w^2 \quad m = 0 \quad (3.23)$$

$$\gamma_{xx}^*(-m) \quad m < 0 \quad (3.24)$$

Disse ligningene, kalt Yule-Walker-ligningene, uttrykker en lineær sammenheng mellom autokorrelasjonen og systemparameterne. De kan uttrykkes på matrisform på følgende måte:

$$\begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \dots & \gamma_{xx}(-p) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \dots & \gamma_{xx}(-p+1) \\ \vdots & \vdots & \vdots & & \vdots \\ \gamma_{xx}(p) & \gamma_{xx}(p-1) & \gamma_{xx}(p-2) & \dots & \gamma_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3.25)$$

For å finne AR-parameterne må korrelasjonsmatrisen inverteres. I Yule-walker utnyttes den spesielle symmetrien i matrisen (Toeplitz-matrise) ved at invertering foregår ved

Levinson-Durbin-algoritmen (Proakis & Manolakis 1996). Siden  $\gamma_{xx}$  ikke er kjent benyttes estimatet:

$$r_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} x(n)x(n+m) \quad m \geq 0 \quad (3.26)$$

I tillegg til minste-kvadraters metode og yule-walker er Burg-algoritmen også blitt testet ut. Den baserer seg på minimalisering av prediksjonsfeil slik at Levinson-Durbin-rekursjonen er tilfredstilt, se Proakis & Manolakis (1996). For ARMA-modeller er det blitt brukt en prediksjonsfeilmethode i MATLAB kalt `armax`. Både yule-walker- og burg-algoritmen finnes i MATLAB, via funksjonen `ar` med henholdsvis `'yw'` og `'burg'` som tilleggsparametere.

### 3.4.3 Minste kvadraters metode for MIMO modeller

Når man har modeller av i typen ligning 3.7, kan MATLAB-funksjonen `arx` brukes. Som i vanlig minste kvadraters metode, er metoden basert på minimering av prediksjonsfeilen. Siden det nå er flere utganger minimeres følgende matrise:

$$\sum_{k=1}^N \epsilon_k^T \epsilon_k \quad (3.27)$$

der  $\epsilon_k$  er prediksjonsfeilen for fremovermodellen definert i ligning 3.20.

### 3.4.4 Subspace identification

For en tilstandsrommodell med en eller flere utganger, kan subspace-algoritmer brukes. I denne oppgaven er det blitt brukt MATLAB-funksjonen `n4sid`. Denne algoritmen returnerer en modell på innovasjonsform:

$$x_{k+1} = Ax_k + Ke_k \quad (3.28)$$

$$y_k = Cx_k + e_k \quad (3.29)$$

Subspace-metoder er basert på at man lager et tilstandsrom for en utgangsvariabel  $y$  ved at man på et gitt tidspunkt betrakter vektorrom bestående av tidligere og fremtidige utgangsverdier. Tilstandsrommet lages ved en geometrisk projeksjon, f.eks av fremtidige verdier ned på tidligere verdier (Picci 1996). Algoritmene er relativt kompliserte og det henvises derfor til Overschee & Moor (1996) og Overschee & Moor (1994).

## 3.5 Klassifisering

Systemparameterne funnet med systemidentifikasjons-metodene beskrevet i forrige kapittel, skal brukes til å skille mellom normale og avvikende spedbarnsbevegelser. En mulig strategi er å bruke systemparameterne direkte som egenskapsvektor i en klassifikator. Denne metoden kan brukes for SISO og MIMO AR-modeller. For dette formålet er det blitt brukt lineær diskriminantanalyse(LDA) som klassifikator.

En annen mulighet er å bruke en metrikk for å måle avstanden mellom to forskjellige dynamiske modeller. I denne sammenheng vil K-Nærmeste-Nabo(KNN) brukes som klassifikator.

Ved klassifisering er det vanlig å dele inn data inn i treningssett og testsett. I dette arbeidet er det blitt delt inn i to omtrent like store sett. Fordelingen av ROI'er på de to settene vises i Tabell 3.3.

Type sett	Normale ROI'er	Avvikende ROI'er
Treningssett	99	25
Testsett	103	23

Tabell 3.3: Antall ROI'er i treningssett og testsett

I inndelingen i trening- og testsett er det tatt forbehold om at alle ROI'er tilhørende en enkelt pasient-id enten befinner seg i treningssettet eller testsettet.

### 3.5.1 Lineær diskriminantanalyse(LDA)

I diskriminantanalyse antar man at egenskapsvektoren  $x \in \mathbb{R}^l$  gitt at den tilhører klassen  $\omega_i$  er normalfordelt:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{\frac{l}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right) \quad (3.30)$$

Her er  $\mu_i = E(x)$  gjennomsnittsverdien til egenskapsvektoren  $x$  for klasse  $i$ , og  $\Sigma_i \in \mathbb{R}^{l \times l}$  er kovariansmatrisen definert som:

$$\Sigma_i = E[(x - \mu_i)(x - \mu_i)^T] \quad (3.31)$$

For å avgjøre hvilken klasse en gitt observasjon hører til brukes "Bayes classification rule":

- $P(\omega_1|x) > P(\omega_2|x)$  -  $x$  hører til  $\omega_1$

- $P(\omega_1|x) < P(\omega_2|x)$  -  $x$  hører til  $\omega_2$

Ved å brukes “Bayes rule” kan disse kriteriene skrives om til (Theodoridis 2006):

- $P(x|\omega_1)P(\omega_1) > P(x|\omega_2)P(\omega_2)$  -  $x$  hører til  $\omega_1$
- $P(x|\omega_1)P(\omega_1) < P(x|\omega_2)P(\omega_2)$  -  $x$  hører til  $\omega_2$

I diskriminantanalyse er disse ulikhetene uttrykt ved en funksjon  $g(x)$  der

$$g_i(x) = \ln(p(x|\omega_i)P(\omega_i)) \quad (3.32)$$

Klassifiseringskriteriet kan nå uttrykkes som:

- $g_i(x) > g_j(x) \forall j \neq i$  -  $x$  hører til  $\omega_i$

Strukturen på kovariansen  $\Sigma_i$  vil ha mye å si for klassifiseringen. Ved å anta en felles kovariansmatrise for de to klassene, vil diskriminantfunksjonen kunne uttrykkes som:

$$g_i(x) = \omega_i^T x + \omega_{i0} \quad (3.33)$$

der  $w_i = \Sigma^{-1}\mu_i$  og  $w_{i0} = \ln(P(\omega_i)) - \frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i$

$g(x)$  er her en lineær funksjon av  $x$ , og de to klassene skilles via et hyperplan. I tilfellet der  $x \in \mathbb{R}^2$  blir hyperplanet redusert til en linje. Gjennomsnittet  $\mu_i$  og kovariansmatrisen  $\Sigma$  blir i LDA vanligvis funnet med maximum likelihood-algoritmen som i dette tilfellet ikke er noe annet en de vanlige estimatene:

$$\hat{\mu} = \frac{1}{N} \sum_{k=1}^N x_k \quad (3.34)$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{k=1}^N (x_k - \hat{\mu})(x_k - \hat{\mu})^T \quad (3.35)$$

LDA finnes i MATLAB som funksjonen `classify` med `linear` som tilleggsparameter.

### 3.5.2 K-Nærmeste Nabo(KNN)

K-Nærmeste-Nabo(KNN) er en veldig enkel klassifikator: Gitt et sett med egenskapsvektorer  $C_1, C_2, \dots$ , der hver vektor hører til en av de  $c$  klassene  $\{\lambda_1, \lambda_2, \dots, \lambda_c\}$ , og treningsettet har like mange tilfeller av de to klassene. For hver ny vektor vi ønsker å klassifisere

undersøkes de  $k$  nærmeste naboene. Hvis flesteparten av naboene hører til klassen  $\lambda_i$ , vil denne nye vektoren bli gruppert i denne klassen. Naboer beregnes ut i fra treningsettet.

Når det er et forskjellig antall vektorer av de to klassene i treningsettet, må den klassen det er minst av tillegges større vekt. Dette vil i praksis si at vi innfører en vektparameter som multipliseres med hver nabo som er av denne klassen. I denne oppgaven, med ROI'er tilhørende den friske og avvikende klassen, vil et naturlig valg for en slik vektparameter være:

$$w = \frac{\text{antall normale ROI'er}}{\text{antall avvikende ROI'er}} \quad (3.36)$$

For å bruke KNN trengs en metrikk for å si hvor nærme to egenskapsvektorer er hverandre. Vi vil i neste seksjon se på metrikker som kan brukes for dynamiske modeller. KNN er en ulineær klassifikator, som kan takle at egenskapsvektorer fra en og samme klasse posisjonerer seg i mange grupper.

### 3.5.3 Metrikker for dynamiske modeller

Vi ønsker å finne et avstandsmål, en metrikk, mellom to dynamiske modeller. Den kanskje mest åpenbare metoden er å konstruere en metrikk som beregner denne avstanden direkte ut i fra modellenes systemparametere, her for en AR-modell:

$$d(M, M') = \left( \sum_{j=1}^p w_j |a_j - a'_j|^2 \right)^{\frac{1}{2}} \quad (3.37)$$

der  $M$  og  $M'$  er to forskjellige AR-modeller og  $w_j$  vektor avstanden mellom systemparametere  $a_j$  og  $a'_j$ . I Martin (2000) kritiseres denne metrikken på flere områder:

1. Det er ingen åpenbar måte å bestemme vektfunksjonen  $w_j$
2. Metrikken har ingen nyttige system-teoretiske egenskaper
3. Metrikken har ingen nyttige matematiske egenskaper

Skal man se på en modell sine egenskaper i frekvensdomenet, kan man med fordel studere modellens poler, som nevnt i seksjon 3.3.2. Punkt 2 begrunnes med at metrikken ikke tar eksplisitt hensyn til polenes plassering, og da spesielt ikke poler utenfor enhetssirkelen som indikerer ustabile systemer.

Punkt 3 går ut på at ARMA modeller har visse egenskaper som gjør at de kan karakteriseres som en gruppe i  $z$ -domenet. Metrikken bør derfor være en enkel funksjon av polene til systemet.

I Martin (2000) foreslås en metrikk basert på sammenligning av cepstrumene til to modeller. Metrikken er definert som følger:

$$d_m(M, M') = \left[ \sum_{n=1}^{\infty} w(n) |c_n - c'_n|^2 \right]^{\frac{1}{2}} \quad (3.38)$$

$$= \left[ \sum_{n=1}^{\infty} n |c_n - c'_n|^2 \right]^{\frac{1}{2}} \quad (3.39)$$

$c_n$  og  $c'_n$  er cepstrum-koeffesienter til to forskjellige modeller,  $M$  og  $M'$ . Metrikken har sitt utspring i at man ønsker å sammenligne frekvensinnholdet i modellene, og et naturlig utgangspunkt er derfor å studere effekt-tetthetsspekteret, gitt av Fouriertransformen til autokorrelasjonen:

$$\Gamma_{xx}(\omega) = \sum_{n=-\infty}^{\infty} \gamma_{xx}(n) e^{-j\omega n} \quad (3.40)$$

Vi vil nå betrakte logaritmen av effekt-tetthetsspekteret,  $\log(\Gamma_{xx}(\omega))$ . Dette er en periodisk funksjon med periode  $2\pi$ , noe som betyr at den kan uttrykkes som en Fourier-serie:

$$\log(\Gamma_{xx}(\omega)) = \sum_{n=-\infty}^{\infty} c(n) e^{-j\omega n} \quad (3.41)$$

Siden  $c(n)$  er invers Fourier-transform av  $\log \Gamma_{xx}(\omega)$  kan vi nå skrive:

$$c(n) = F^{-1}(\log(F(\gamma_{xx}(\omega)))) = F^{-1}(\log(\Gamma_{xx}(\omega))) \quad (3.42)$$

der  $F$  og  $F^{-1}$  er henholdsvis Fouriertransform og invers Fouriertransform.  $c(n)$  er definert som cepstrumet til  $\gamma_{xx}(n)$ . Hvis vi ser bort fra logaritme-funksjonen, reduseres cepstrumet til autokorrelasjonen. Årsaken til å se på studere cepstrumet fremfor autokorrelasjonen kommer frem hvis vi ved hjelp av ligning 3.41 skriver:

$$\Gamma_{xx}(\omega) = \exp\left( \sum_{n=-\infty}^{\infty} c(n) e^{-j\omega n} \right) = \sigma_w^2 |H(\omega)|^2 \quad (3.43)$$

der vi har brukt sammenhengen mellom systemfunksjonen og effekt-tetthetsspekteret, som diskutert i seksjon 3.3.2. Av ligning 3.43 kan vi utlede følgende sammenheng:

$$H(\omega) = \exp\left( \sum_{n=1}^{\infty} c(n) e^{-j\omega n} \right) \quad (3.44)$$

$$(3.45)$$

der vi har definert  $\sigma_w^2 = e^{c(0)}$ .  $c(0)$  er dermed ikke knyttet til ARMA-koeffesientene til modellen. Vi ser at systemfunksjonen  $H(\omega)$  elegant kan uttrykkes ved cepstrum-koeffesientene  $c(n)$ . Cepstrum-koeffesientene er direkte knyttet til ARMA-modellens poler og nullpunkter Martin (1997):

$$c_n = \frac{1}{n} \left( \sum_{j=1}^p \alpha_j^n - \sum_{j=1}^q \beta_j^n \right) \quad n > 0 \quad (3.46)$$

$$\ln \sigma_w^2 \quad n = 0 \quad (3.47)$$

$$c_n = \frac{1}{n} \left( \sum_{j=1}^p (\alpha_j^*)^{-n} - \sum_{j=1}^q (\beta_j^*)^{-n} \right) \quad n < 0 \quad (3.48)$$

Hvis vektparameteren  $w(n)$  i metrikken settes som  $w(n) = n$ , kan den uendelige summen i ligning 3.39 uttrykkes som et endelig produkt mellom modellenes poler og nullpunkter (Martin 2000):

$$d(M, M') = \left( \ln \frac{\prod_{i=1}^p \prod_{j=1}^{p'} (1 - \alpha_i \alpha_j'^*) \prod_{i=1}^{p'} \prod_{j=1}^p (1 - \alpha_i' \alpha_j^*)}{\prod_{i=1}^p \prod_{j=1}^p (1 - \alpha_i \alpha_j^*) \prod_{i=1}^{p'} \prod_{j=1}^{p'} (1 - \alpha_i' \alpha_j'^*)} \right)^{\frac{1}{2}} \quad (3.49)$$

der  $\alpha$  og  $\alpha'$  er polene til modellene  $M$  og  $M'$ . Ligning 3.49 gjelder for stabile AR-modeller, men som vi senere skal se kan ARMA-modeller transformeres slik at 3.49 likevel kan brukes. I motsetning til en metrikk direkte basert på systemparameterne (3.37) har vi nå sett at metrikken  $d_m$  er en enkel funksjon av modellens poler og nullpunkter, og er direkte knyttet til effekt-tetthetsspekteret til modellen. Metrikken er sensitiv til polenenes plassering innenfor enhetssirkelen, og metrikken er singularær hvis en pol ligger akkurat på enhetssirkelen Martin (2000).

I cepstrum-domenet er konvolusjon mellom to signaler det samme som å addere cepstrumene. Dette medfører at avstanden mellom to modeller  $M$  og  $M'$ ,  $d_m(M, M')$ , ikke vil endre seg hvis de to modellene kjøres gjennom et lineært filter  $M''$ , det vil si  $d(MM'', M'M'') = d(M, M')$  (Martin 2000). Vi kan bruke denne egenskapen til å bruke ligning 3.49 til å regne ut  $d(M, M')$  for ARMA modeller. Har vi to ARMA-modeller  $M = \frac{B}{A}$  og  $M' = \frac{B'}{A'}$ , vil  $d_m(M, M') = d_m(N, N')$  der  $N$  og  $N'$  er to AR-modeller representert ved  $N = \frac{1}{AB'}$  og  $N' = \frac{1}{BA'}$ . Dette kommer frem ved å sette  $M'' = \frac{1}{BB'}$  (Martin 2000).

I Cock & de Moor (2000) presenteres 3 forskjellige metrikker for SISO tilstandsrommodeller, hvor av en er metrikken diskutert ovenfor. Utgangspunktet er at man har modeller på formen:

$$x_{k+1} = Ax_k + w_k \quad (3.50)$$

$$y_k = Cx_k + v_k \quad (3.51)$$

Denne modellen kan representere både AR- og ARMA-modeller, avhengig om modellen har kun poler, eller poler og nullpunkter. At tilstandsrommodellen og ARMA-modellen er ekvivalente ble diskutert i 3.3.

For AR-modeller baserer metrikkene seg på prinsipalvinkler mellom de uendelige observerbarhetsmatrisene til modellene, og de resulterende vinklene blir kalt for underromsvinkler. Den uendelige observerbarhetsmatrisen til en modell er gitt av:

$$\mathcal{O}_\infty(M_i) = \begin{bmatrix} C_i \\ C_i A_i \\ C_i A_i^2 \\ \vdots \end{bmatrix} \quad (3.52)$$

der  $M_i$  er en modell og  $i \in 1, 2$ . Underromsvinkelen mellom  $M_1$  og  $M_2$  er definert som prinsipalvinklene mellom  $\text{range}(\mathcal{O}_\infty(M_1))$  og  $\text{range}(\mathcal{O}_\infty(M_2))$ . Se (Cock & de Moor 2000) for definisjon av prinsipalvinkler. For ARMA-modeller er underromsvinklene definert slik at de ivaretar egenskapen fra  $d_m$ -metrikken, nemlig at underromsvinklene  $M = \frac{B}{A}$  og  $M = \frac{B'}{A'}$  tilsvarer underromsvinklene mellom  $N = \frac{1}{AB'}$  og  $N = \frac{1}{B'A'}$ . Dette fører til at underromsvinklene defineres som prinsipalvinklene mellom  $\text{range}(\mathcal{O}_\infty(M), \mathcal{O}_\infty(M'^{-1}))$  og  $\text{range}(\mathcal{O}_\infty(M), \mathcal{O}_\infty(M'^{-1}))$ . I Cock & de Moor (2000) presenteres det to forskjellige måter å regne ut underromsvinklene på, ved bruk av “the observability gramian”, eller via modellens poler.

For både AR og ARMA-modeller defineres følgende metrikker, der  $d_m$  er metrikken diskutert tidligere.

$$d_m(M, M')^2 = \ln \prod_{i=1}^n \frac{1}{\cos^2 \theta_i} \quad (3.53)$$

$$d_g(M, M') = \sin(\theta_{max}) \quad (3.54)$$

$$d_f(M, M')^2 = 2 \sum_{i=1}^{2n} \sin^2 \theta_i \quad (3.55)$$

der  $\theta_1, \theta_2, \dots, \theta_n$  er underromsvinkler mellom  $M$  og  $M'$ .

I (Bissacco et al. 2001) presenteres en metode for å regne ut underromsvinkler mellom MIMO tilstandsrommodeller, som er brukt i denne oppgaven. Metrikkene ovenfor kan deretter brukes på underromsvinklene, for å regne ut distansen mellom MIMO-modeller.

### 3.5.4 Mål på separabilitet

I tilfellet der vi bruker systemparametere direkte som egenskapsvektor, er det ønskelig å undersøke hvor godt disse skiller mellom de to klassene. Dette er en analyse som kan gjøres før klassifisering, og som dermed ikke er avhengig av en gitt klassifikator. I Theodoridis (2006) presenteres det to forskjellige metoder for å måle separabilitet.



Det første separabilitetsmålet kan utledes fra Bayes klassifiseringsregel. Uttrykkene

$$\begin{aligned} D_{12}(x) &= \ln \frac{p(\omega_1|x)}{p(\omega_2|x)} \\ D_{21}(x) &= \ln \frac{p(\omega_2|x)}{p(\omega_1|x)} \\ d_{12}(x) &= D_{12}(x) + D_{21}(x) \end{aligned} \quad (3.56)$$

måler separabilitet med hensyn på klassene  $\omega_1$  og  $\omega_2$  for en gitt egenskapsvektor  $x$ . Ved å så betrakte 3.56 over alle  $x$  fåes et endelig mål på separabiliteten, se Theodoridis (2006) for formel.

Vi antar nå at egenskapsvektorene  $x$  er normalfordelt og at de to klassene har lik varians. I dette tilfellet kan  $d_{12}$  skrives som:

$$d_{12} = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) \quad (3.57)$$

der  $\mu_1$  og  $\mu_2$  er gjennomsnittet til egenskapsvektoren til henholdsvis klasse 1 og 2, og  $\Sigma$  er kovariansmatrisen.

Det andre separabilitetsmålet går ikke veien om sannsynlighetsfordelinger, men bruker i stedet scatter-matriser, som inneholder informasjon om hvordan egenskapsvektorene,  $x \in \mathbb{R}^n$ , er spredd i et  $n$ -dimensjonalt rom. Følgende Scatter-matriser defineres:

Scatter-matrise innenfor klasser(eng:Within-class scatter matrix):

$$S_w = \sum_{i=1}^M P_i S_i \quad (3.58)$$

der  $S_i$  er kovariansmatrisen til klasse  $\omega_i$ :

$$S_i = E[(x - \mu_i)(x - \mu_i)^T] \quad (3.59)$$

og  $P_i$  er sannsynligheten for klasse  $\omega_i$ . I beregninger kan  $P_i$  settes som:

$$P_i \approx \frac{n_i}{N} \quad (3.60)$$

$S_w$  er et mål på gjennomsnittet til variansen og kovariansen til elementene i egenskapsvektoren tatt over alle klassene.

Scatter-matrise mellom klasser(eng:Between-class scatter matrix):

$$S_b = \sum_{i=1}^M P_i (\mu_i - \mu_0)(\mu_i - \mu_0)^T \quad (3.61)$$

der  $\mu_0$  er det globale gjennomsnittet:

$$\mu_0 = \sum_{i=1}^M P_i \mu_i \quad (3.62)$$

$S_b$  måler gjennomsnittsavstanden som klassene har fra klassenes gjennomsnitt til det globale gjennomsnitt.

Blandet scatter-matrise (eng: Mixture scatter matrix)

$$S_m = E[(x - \mu_0)(x - \mu_0)^T] \quad (3.63)$$

Det kan vises at  $S_m$  kan skrives som:

$$S_m = S_w + S_b \quad (3.64)$$

$S_m$  er kovariansmatrisen til egenskapsvektoren med hensyn på det globale gjennomsnittet. For å måle hvor bra klassene er separert innføres kriteriet:

$$J = \text{trace}(S_w^{-1} S_m) \quad (3.65)$$

Et annet mulig kriterie er:

$$J_2 = \frac{\text{trace}(S_m)}{\text{trace}(S_w)} \quad (3.66)$$

Det er kriteriet  $D_{12}$  og  $J$  som har blitt brukt som mål på separabilitet i denne oppgaven.

### 3.5.5 Kombinering av egenskapsvektorer

Hvis systemparametere brukes direkte som egenskapsvektorer i klassifisering, kan systemparametere fra flere ulike modeller identifisert fra forskjellige utgangsvARIABLE kombineres for hver ROI. Betrakter vi to modeller,  $M_1$  og  $M_2$  identifisert fra to forskjellige utgangsvARIABLE for samme ROI, kan vi kombinere systemparametere i en egenskapsvektor:  $v = [v_1^T v_2^T]^T$  der  $v_1$  og  $v_2$  er systemparametere for henholdsvis  $M_1$  og  $M_2$ .

I tilfellet der vi ikke bruker systemparametere direkte, men bruker en metrikk for å regne ut avstanden mellom modeller, kan også modeller funnet fra forskjellige utgangsvARIABLE kombineres. Hvis  $d(M_1, M_1')$  er avstanden mellom to ROI'er identifisert fra utgangsvARIABLE 1, og  $d(M_2, M_2')$  er avstanden mellom de samme ROI'ene identifisert fra utgangsvARIABLE 2, kan vi regne ut avstanden mellom de to ROI'ene som:

$$d(ROI, ROI') = d(M_1, M_1') + d_2(M_2, M_2') \quad (3.67)$$

### 3.6 Oversikt over algoritmer

Hittil i dette kapitlet er det presentert modeller, systemidentifikasjonsmetoder og klassifikatorer. Tabell 3.4 oppsummerer de forskjellige algoritmene, og hvordan de kombineres. Følgende forkortelser er brukt:

MK: Minste kvadraters metode

YW: Yule Walker-metoden

FB: Forward-Backward-metoden

PF: Prediksjonsfeilmeter

LDA: Lineær Diskriminant Analyse

KNN: K-Nærmeste-Nabo

Modell	Systemidentifikasjon	Klassifikator
AR (SISO)	MK, YW, Burg, FB	LDA, KNN(metrikk: $d_m$ )
ARMA(SISO)	PF	LDA, KNN (metrikk: $d_m$ )
Tilstandsrom(SISO)	subspace (n4sid)	KNN (metrikk: $d_m, d_f$ eller $d_g$ )
AR(MIMO)	MK	LDA
Tilstandsrom(MIMO)	subspace(n4sid)	KNN(metrikk: $d_m, d_f$ eller $d_g$ )

Tabell 3.4: Oversikt over metoder i kapittel 3

Legg merke til at  $d_m$  metrikken både brukes for AR-modeller og tilstandsrommodeller, som vist i seksjon 3.5.3.



## Kapittel 4

# Resultater

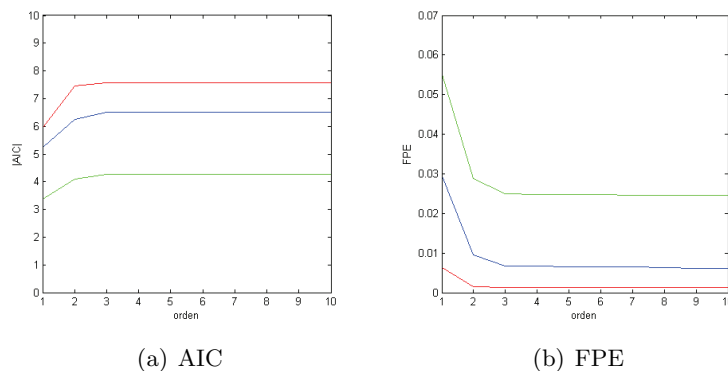
Dette kapitlet presenterer resultater fra modelltilpasning og klassifisering. Resultater blir diskutert fortløpende, men en overordnet diskusjon kommer i Kapittel 5. Vi vil først se på hva som er et fornuftig valg av orden på AR-modellene. Seksjon 4.2 tar for seg begrepene spesifisitet og sensitivitet, som er sentrale for å avgjøre hvor god klassifiseringen er. Seksjon 4.3 viser resultatet av en PCA-analyse på sensordata, samt en analyse av varians for de forskjellige sensorene. Fordi vi er interessert i resultater på pasient-nivå tar seksjon 4.4 for seg vektning av ROI'er. I seksjon 4.5 undersøkes det hvordan modeller identifisert fra ulike utgangsvARIABLE skiller mellom de to gruppene, med hensyn på spesifisitet og sensitivitet. De beste resultatene fra denne seksjonen blir så analysert videre: I seksjon 4.6 ser vi blant annet på hvordan endring i modellorden og systemidentifikasjonsmetode påvirker resultatene. Til slutt, i seksjon 4.7, analyseres de samme modellene med hensyn på systemteoretiske egenskaper som effekt-tetthetsspektre, impulsrespons og poler. Metoder er implementert i MATLAB, og det henvises til tillegg C.

### 4.1 Orden på AR-modell

Valg av orden på modell vil i stor grad avhenge av hvilke utgangsvARIABLE modellen er blitt identifisert fra. For enkelhets skyld, er det i denne oppgaven blitt brukt samme orden på alle AR-modeller. Den samme ordenen er også brukt på tilstandsrommodeller og MIMO AR-modeller. Valget av orden ble valgt ut i fra analysen som følger.

Figur 4.1 viser verdien på AIC- og FPE-kriteriet for modeller identifisert fra 3 forskjellige utgangsvARIABLE. Verdiene er basert på ROI'er fra den normale gruppen, og grafene representerer gjennomsnittet av alle ROI'ene. Vi ser verdiene på kriteriene flater ut etter 3. orden. Merk at det er absoluttverdien av AIC-kriteriet som er plottet, og en høyere verdi på grafen betyr dermed en bedre modelltilpasning. I følge AIC- og FPE-kriteriet

skal man velge den ordenen som gir lavest verdi. Da AIC-kriteriet har en tendens til å overestimere ordenen (Brockwell & A.Davis 2001), tas det mest hensyn til der grafene flater ut. En orden på 3 eller 4 virker fornuftig, og i denne oppgaven er orden 4 valgt.



Figur 4.1: AIC- og FPE-kriterium: Blå:  $y_{4_1}$ , Rød:  $y_{6_2}$ , Grønn:  $y_{34_x}$

Vi kan validere valget av orden med autokorrelasjons-kriteriet. Figur 4.2 illustrerer kriteriet for orden 2 og 4, for en modell funnet fra en ROI fra et barn i den normale gruppen, basert på utgangsvariabel  $y_{4_1}$ . Vi ser at en orden på 4 gir mer fornuftig modellering enn en orden på 2, som har mange verdier utenfor grensen  $\pm \frac{1.96}{\sqrt{n}}$ . Vi ser og at en økning i orden til 6 ikke i stor grad bedrer situasjonen.

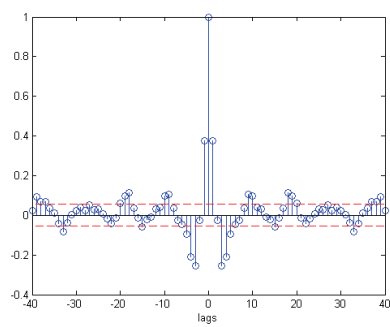
## 4.2 Definisjon av sensitivitet og spesifisitet

Sensitivitet er et mål på hvor bra en binær klassifiseringstest identifiserer tilstanden som det testes for. Sensitivitet er definert som sannsynligheten for at en test gir positivt utslag, gitt at tilstanden faktisk er positiv. En positiv tilstand vil si at testpersonen har tilstanden det testes for, i dette tilfellet Cerebral Parese.

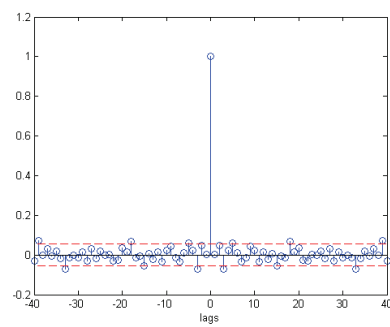
Spesifisitet er et mål på hvor bra en binær klassifiseringstest identifiserer en negativ tilstand. Spesifisiteten er definert som sannsynligheten for at en test gir negativt utslag, gitt at tilstanden faktisk er negativ.

Ved klassifisering kan 4 situasjoner oppstå:

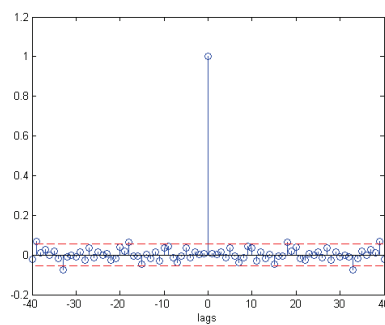
1. True Positive(TP):Tilstanden er positiv og testen er positiv
2. False Negative(FN):Tilstanden er positiv og testen er negativ
3. False Positive(FP):Tilstanden er negativ og testen er positiv
4. True Negative(TN):Tilstanden er negativ og testen er negativ



(a) 2.ordens



(b) 4.ordens



(c) 6.ordens

Figur 4.2: Autokorrelasjonskriteriet på modell identifisert fra  $y_{41}$

Faktisk \ Test	P	N
P	TP	FN
N	FP	TN

Tabell 4.1: Mulige utfall ved klassifisering

De 4 tilfellene er oppsummert i Tabell 4.1.

Sensitivitet og spesifisitet til en test kan uttrykkes matematisk som:

$$\text{Sensitivitet} = \frac{TP}{TP + FN} \quad (4.1)$$

$$\text{Spesifisitet} = \frac{TN}{TN + FP} \quad (4.2)$$

Ofte vil man kunne variere en test sin sensitivitet og spesifisitet ved å endre egenskapene til klassifikatoren. En vanlig måte å fremstille endring av sensitivitet og spesifisitet i et slikt tilfelle er ved hjelp av et ROC-plot, en grafisk fremsilling av sensitivitet mot (1-spesifisitet).

### 4.3 PCA og varians

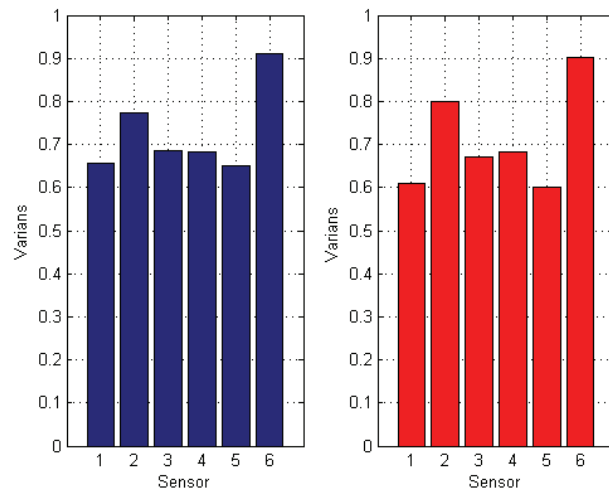
I seksjon 3.2.1 ble det diskutert hvordan PCA kan brukes for å redusere dimensjonene på data. Vi vil i denne seksjonen studere hvor mye av variansen som forklares ved å betrakte en prinsipalkomponent for de 6 sensorene, der observasjonsvektoren består av målinger i  $x$ ,  $y$  og  $z$ -retning. Dette vil resultere i 3 egenvektorer med tilhørende egenverdier. Vi vil i tillegg studere variansen langs aksene  $x$ ,  $y$  og  $z$ , samt summen av disse.

Figur 4.3 viser i gjennomsnitt hvor mye av variansen i dataene som forekommer langs den prinsipale komponenten, for Sensor 1 til 6. Gjennomsnittet er tatt over alle ROI'er til henholdsvis barn med normale (blå) og avvikende (rød) bevegelser. Figuren viser ingen store forskjeller mellom de to gruppene. Det vi derimot kan se er at sensor 2 og 6 skiller seg ut med å ha størst del av variansen langs prinsipalkomponenten. Dette er forventet da bevegelsene til hodet og brystet til spedbarna i de fleste tilfeller er mindre komplekse enn bevegelser til armer og ben.

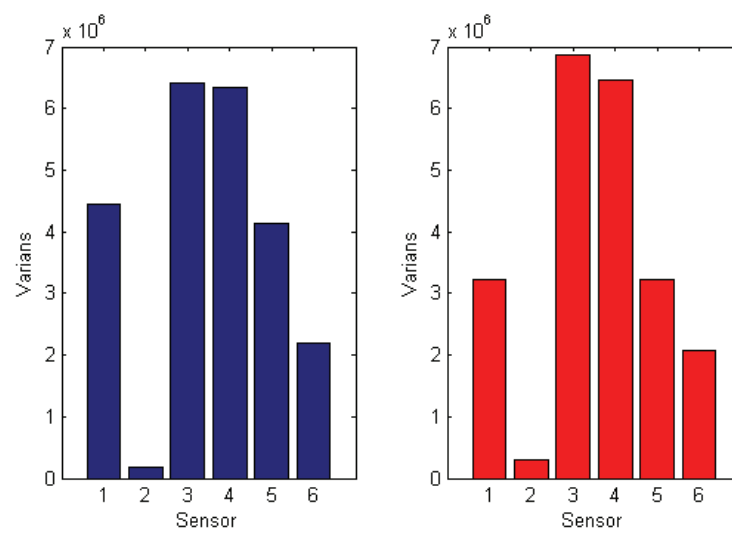
Vi kan også studere den totale variansen til forskjellige sensorer, som vist i Figur 4.4. Figuren viser at de normale barna har betydelig større varians på sensor 1 og 5. Figur 4.5 viser variansen langs aksene  $x$ ,  $y$  og  $z$ . Vi ser at det er en stor skilnad i  $z$ -retning for sensor 1 og 5.

Figur 4.6 viser resultatet av en PCA-beregning for en ROI. Figuren viser kun  $xy$ -planet, prinsipalkomponenten og dataene har også en komponent i  $z$ -retning.

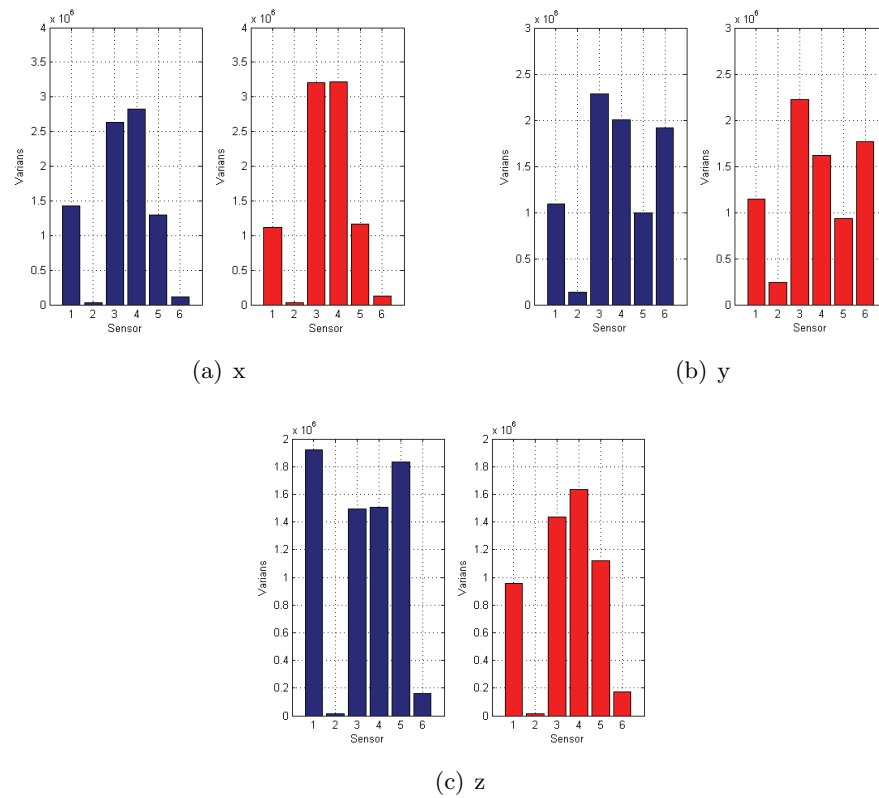
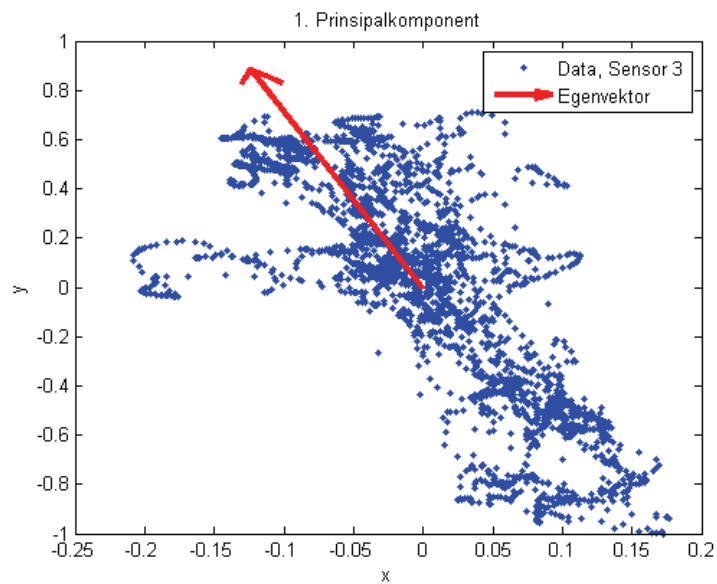




Figur 4.3: Delmengde av varians langs prinsipalkomponent for de 6 sensorene



Figur 4.4: Varians for sensorer over alle 3 akser

Figur 4.5: Varians langs  $x$ ,  $y$  og  $z$  for de 6 sensorene

Figur 4.6: XY-plot av prinsippkomponent og normalisert data for Sensor 3

## 4.4 Vekting av ROI'er

Ved klassifisering vil hver ROI eller mini-ROI klassifiseres som normal eller avvikende. Ettersom vi ønsker å klassifisere hver pasient inn i en av de to klassene, må ROI'er tilhørende en pasient sammenlignes for å avgjøre hvilken gruppe pasienten tilhører. Det kanskje mest åpenbare er å si at pasienten er frisk hvis flertallet av ROI'ene blir klassifisert som normale. Dette behøver derimot ikke være den beste løsningen. Kanskje er det slik, i hvert fall sett ut i fra et systemteoretisk perspektiv, at bare noen av ROI'ene inneholder de nødvendige karakteristiske bevegelsene som viser om spedbarnet tilhører den normale eller avvikende gruppen. Vi kan dermed vekte ROI'ene til en pasient etter hvilken klasse de tilhører:

$$\begin{aligned}
 ROI_n &: \text{antall normale ROI'er} \\
 ROI_a &: \text{antall avvikende ROI'er} \\
 ROI_n &\geq w \cdot ROI_a \rightarrow \text{normal} \\
 ROI_n &< w \cdot ROI_a \rightarrow \text{avvikende}
 \end{aligned} \tag{4.3}$$

der  $w$  er vektparameteren. De mest ekstreme tilfellene er å anta at pasienten er frisk/syk hvis en av ROI'ene klassifiseres som normal/avvikende. Disse tilfellene tas med ved å la vektparameteren være stor/liten nok. I denne oppgaven er følgende verdier blitt brukt på vektparameteren:

$$w = 10^a \tag{4.4}$$

der  $a$  tar verdiene:  $\{-1, -0.9, \dots, -0.1, 0, 0.1, \dots, 0.9, 1\}$

## 4.5 Resultater fra separabilitetsanalyse og klassifisering

I denne seksjonen presenteres resultater fra separabilitetsanalyse og klassifisering. Det undersøkes hvordan dynamiske modeller identifisert fra forskjellige utgangvariable skiller mellom de to gruppene. For SISO-tilfellet er AR-modeller blitt brukt, identifisert med minste kvadraters metode. FOR MIMO-modeller er både AR-modeller og tilstandsrommodeller blitt brukt. De er blitt identifisert med henholdsvis minste kvadraters metode og subspace-algoritmen n4SID. LDA og KNN er brukt som klassifikatorer. For KNN er de 10 nærmeste naboene blitt tatt hensyn til, og det er blitt brukt vektingen foreslått i 3.5.2. KNN er blitt brukt sammen med  $d_m$ -metrikken. Andre metrikker kunne også ha vært brukt, og  $d_f$ -metrikken viste seg å gi veldig like resultater, mens  $d_g$ -metrikken gav vesentlig dårligere resultater i noen utvalgte tester. Det er også verdt å nevne at metoden `classify` i MATLAB også kan brukes til kvadratisk diskriminant analyse i tillegg til LDA. I de tilfeller det ble testet ut fungerte det dårligere, så dette kapitlet fokuserer på LDA.

For å måle separabilitet brukes de to kriteriene,  $d$  og  $J$ , fra seksjon 3.5.4. Separabilitetsanalysen er utført der det er brukt AR-modeller i systemidentifikasjonen, og er med på å verifisere resultatene fra klassifiseringen. Kriteriene  $J$  og  $d$ , sier noe om hvor godt 2 klasser kan skilles fra hverandre i rommet, og vi forventer et samsvar mellom verdiene på disse kriteriene og den lineære klassifikatoren.

Vi vil først se på SISO-modeller.

#### 4.5.1 SISO-modell

I den første analysen er modellene identifisert fra utgansvariable bestående av en akse ( $x, y$  eller  $z$ ) på en enkelt sensor.

(a) Sensor 1-6, akse $x$			(b) Sensor 1-6, akse $y$		
variabel	J	d	variabel	J	d
$y_{1_x}$	4.1331	0.7583	$y_{1_y}$	4.0768	0.4611
$y_{2_x}$	4.1709	0.9457	$y_{2_y}$	4.1353	0.7723
$y_{3_x}$	<b>4.2491</b>	<b>1.2892</b>	$y_{3_y}$	4.1865	1.0174
$y_{4_x}$	4.2067	1.1067	$y_{4_y}$	4.1528	0.8573
$y_{5_x}$	4.0965	0.5686	$y_{5_y}$	4.0124	0.0793
$y_{6_x}$	4.1673	0.9286	$y_{6_y}$	<b>4.3457</b>	<b>1.6610</b>

(c) Sensor 1-6, akse $z$			(d) Gjennomsnitt over $x, y$ og $z$		
variabel	J	d	variabel	J	d
$y_{1_z}$	4.0356	0.2227	$y_1$	4.0818	0.4807
$y_{2_z}$	4.0165	0.1048	$y_2$	4.1075	0.6076
$y_{3_z}$	4.1178	0.6824	$y_3$	4.1845	0.9963
$y_{4_z}$	<b>4.1536</b>	<b>0.8601</b>	$y_4$	4.1710	0.9414
$y_{5_z}$	4.0180	0.1143	$y_5$	4.0423	0.2541
$y_{6_z}$	4.1204	0.6940	$y_6$	<b>4.2111</b>	<b>1.0945</b>

Tabell 4.2: Separabilitet for sensor 1-6, akse  $x$ ,  $y$  og  $z$

I Tabell 4.2 vises separabilitet-verdier for AR-parameterne. De største verdiene for hver kolonne er uthevet i fet skrift. Den høyeste verdien for både  $J$ -kriteriet og  $d$ -kriteriet oppnås ved å betrakte sensor 6(panne) i  $y$ -retning. Hvis vi ser på gjennomsnittet av de to kriteriene over alle aksene, kommer også sensor 6 best ut. Vi ser også at sensor 3 og 4(armene) skiller bedre mellom de to klassene enn sensor 1(venstre ankel), 2(bryst) og 5(høyre ankel). Sensor 1 og 5 skiller jevnt over minst mellom de to gruppene. Dette må sies å være noe overraskende med tanke på mengden av bevegelse i forhold til hodet og bryst.

I Tabell 4.3 vises spesifisitet og sensitivitet ved bruk av LDA på AR-parameterne, der

(a) Sensor 1-6, akse  $x$ 

Variabel	Spesifisitet	Sensitivitet
$y_{1_x}$	0.6087	0.6699
$y_{2_x}$	0.5652	0.7282
$y_{3_x}$	0.5652	0.7767
$y_{4_x}$	0.4783	0.8350
$y_{5_x}$	0.2609	0.5922
$y_{6_x}$	0.6087	0.7476

(b) Sensor 1-6, akse  $y$ 

Variabel	Spesifisitet	Sensitivitet
$y_{1_y}$	0.6087	0.6117
$y_{2_y}$	0.4783	0.6505
$y_{3_y}$	0.5652	0.6214
$y_{4_y}$	0.5652	0.6990
$y_{5_y}$	0.4348	0.4078
$y_{6_y}$	<b>0.7826</b>	<b>0.7864</b>

(c) Sensor 1-6, akse  $z$ 

Variabel	Spesifisitet	Sensitivitet
$y_{1_z}$	0.5652	0.5243
$y_{2_z}$	0.4348	0.5437
$y_{3_z}$	0.5217	0.6019
$y_{4_z}$	0.7826	0.5437
$y_{5_z}$	0.3478	0.6117
$y_{6_z}$	0.4348	0.7476

Tabell 4.3: Spesifisitet og sensitivitet for sensor 1-6, akse 1-3, LDA

(a)  $x$ 

variabel	Spesifisitet	Sensitivitet
$y_{1_x}$	0.8447	0.4783
$y_{2_x}$	0.6796	0.5652
$y_{3_x}$	0.7670	0.5652
$y_{4_x}$	0.8932	0.4348
$y_{5_x}$	0.7087	0.5652
$y_{6_x}$	0.7767	0.4783

(b)  $y$ 

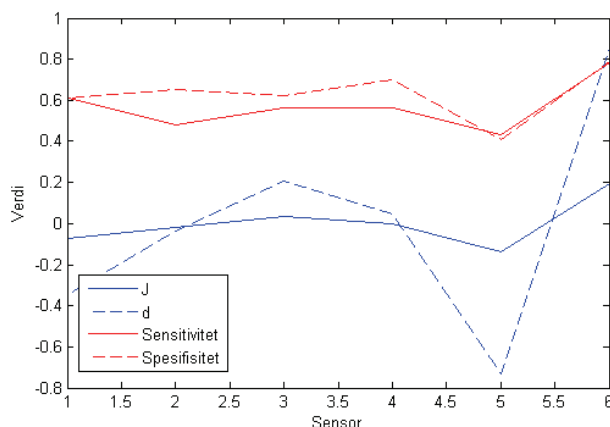
variabel	Spesifisitet	Sensitivitet
$y_{1_y}$	0.6505	0.5217
$y_{2_y}$	0.6893	0.3478
$y_{3_y}$	0.6505	0.5217
$y_{4_y}$	0.6796	0.6087
$y_{5_y}$	0.7184	0.4783
$y_{6_y}$	<b>0.9029</b>	<b>0.5652</b>

(c)  $z$ 

variabel	Spesifisitet	Sensitivitet
$y_{1_z}$	0.6699	0.2609
$y_{2_z}$	0.6602	0.2174
$y_{3_z}$	<b>0.6990</b>	<b>0.7391</b>
$y_{4_z}$	<b>0.6990</b>	<b>0.6957</b>
$y_{5_z}$	0.7476	0.3043
$y_{6_z}$	<b>0.8058</b>	<b>0.6087</b>

Tabell 4.4: Spesifisitet og sensitivitet for sensor 1-6, akse 1-3, KNN

gode resultater er uthevet i fet skrift. Det klart beste resultatet oppnås, som ved separabilitetsanalysen, for sensor 6 langs  $y$ -aksen. Dette resultatet må sies å være noe overraskende. Sensor 6 (panne) er den av de målte kroppsdelene som beveger seg minst, hvis vi måler bevegelse i form av varians. Til gjengjeld har sensor 6 mesteparten av variansen langs  $y$ -aksen. En nærmere analyse av dette resultatet gjøres i seksjon 4.6, der vi ser på om de identifiserte modellene fra de to gruppene kan forklare resultatet.



Figur 4.7: Sammenheng mellom separabilitet, sensitivitet og spesifisitet

Av Figur 4.7 ser vi at det er en sammenheng mellom separabilitetsanalysen og sensitivitet og spesifisitet, der kriteriene  $J$  og  $d$  er justert for gjennomsnittet. Vi ser tydelig at resultatene fra separabilitetsanalysen er korrelert med spesifisiteten og sensitiviteten.

For KNN fås best resultater for sensor 3, 4 og 6 ved å betrakte målinger langs  $z$ -aksen. At  $z$ -retning alene skiller såpass bra er noe overraskende. Akse  $y$  på sensor 6 viser nå kun høy sensitivitet, men lav spesifisitet.

Samme analyse foretas så på utangsvare beregnet fra avstander mellom sensorene. Av Tabell 4.5 ser vi en klar tendens: Alle separabilitetsverdier regnet ut for avstand mellom sensorer i  $x$ -retning, er større enn tilsvarende avstander i  $y$ -retning, som igjen er større enn avstandene i  $z$ -retning. Separabilitet beregnet ut i fra absoluttavstander viser marginalt bedre verdier enn  $x$ -retning alene, men ikke på det sensorparet som skiller mest (sensor 3 og 4). Det klart beste resultatet får vi ved å betrakte avstanden mellom de to armene langs  $x$ -aksen (langs spedbarnet).

I Tabell 4.6 presenteres resultater for spesifisitet og sensitivitet for LDA. Igjen er det samsvar med beste resultat fra separabilitetsanalysen. For KNN fås beste resultat noe overraskende ved å betrakte absolutt-avstanden mellom sensor 4 og sensor 5.

Regner vi med utgangsvare basert på avstand fra origo (bryst) til sensorene, får vi separabiliteten vist i Tabell 4.8. I Tabell 4.9 vises resultater for spesifisitet og sensitivitet

(a) $x$			(b) $y$		
variabel	J	d	variabel	J	d
$\bar{y}_{13_x}$	4.1838	1.0039	$\bar{y}_{13_y}$	4.1522	0.8546
$\bar{y}_{14_x}$	4.1734	0.9545	$\bar{y}_{14_y}$	4.1085	0.6337
$\bar{y}_{15_x}$	4.1074	0.6279	$\bar{y}_{15_y}$	4.0297	0.1867
$\bar{y}_{34_x}$	<b>4.3334</b>	<b>1.6146</b>	$\bar{y}_{34_y}$	<b>4.1900</b>	<b>1.0332</b>
$\bar{y}_{35_x}$	4.2439	1.2674	$\bar{y}_{35_y}$	4.0371	0.2320
$\bar{y}_{45_x}$	4.1934	1.0472	$\bar{y}_{45_y}$	4.1546	0.8665

(c) $z$			(d) absolutt-avstand		
variabel	J	d	variabel	J	d
$\bar{y}_{13_z}$	4.0276	0.1738	$\bar{y}_{13}$	4.1639	0.9104
$\bar{y}_{14_z}$	4.0606	0.3699	$\bar{y}_{14}$	4.1736	0.9558
$\bar{y}_{15_z}$	4.0224	0.1416	$\bar{y}_{15}$	4.0272	0.1712
$\bar{y}_{34_z}$	<b>4.1572</b>	<b>0.8776</b>	$\bar{y}_{34}$	4.2071	1.1103
$\bar{y}_{35_z}$	4.0369	0.2299	$\bar{y}_{35}$	<b>4.2460</b>	<b>1.2759</b>
$\bar{y}_{45_z}$	4.0560	0.3431	$\bar{y}_{45}$	4.1954	1.0566

Tabell 4.5: Separabilitet, avstand mellom sensorer

(a) $x$			(b) $y$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{13_x}$	0.6602	0.3913	$\bar{y}_{13_y}$	0.6408	0.4783
$\bar{y}_{14_x}$	0.7573	0.4348	$\bar{y}_{14_y}$	0.6699	0.5652
$\bar{y}_{15_x}$	0.5922	0.5652	$\bar{y}_{15_y}$	0.4272	0.6087
$\bar{y}_{34_x}$	<b>0.7961</b>	<b>0.8261</b>	$\bar{y}_{34_y}$	0.7379	0.5217
$\bar{y}_{35_x}$	0.7184	0.4783	$\bar{y}_{35_y}$	0.5146	0.4783
$\bar{y}_{45_x}$	0.7670	0.4348	$\bar{y}_{45_y}$	0.6796	0.5217

(c) $z$			(d) absolutt-avstand		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{13_z}$	0.5437	0.6087	$\bar{y}_{13}$	0.6602	0.3913
$\bar{y}_{14_z}$	0.6019	0.6522	$\bar{y}_{14}$	0.7282	0.5652
$\bar{y}_{15_z}$	0.6019	0.4348	$\bar{y}_{15}$	0.4757	0.6087
$\bar{y}_{34_z}$	0.6019	0.6522	$\bar{y}_{34}$	0.6990	0.4783
$\bar{y}_{35_z}$	0.6019	0.4348	$\bar{y}_{35}$	0.7282	0.5217
$\bar{y}_{45_z}$	0.5922	0.4783	$\bar{y}_{45}$	0.6796	0.5217

Tabell 4.6: Spesifisitet og sensitivitet, avstand mellom sensorer, LDA

(a) $x$			(b) $y$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{13_x}$	0.6990	0.4783	$\bar{y}_{13_y}$	0.7184	0.5217
$\bar{y}_{14_x}$	0.8641	0.5217	$\bar{y}_{14_y}$	0.7961	0.3913
$\bar{y}_{15_x}$	0.6408	0.3913	$\bar{y}_{15_y}$	0.6214	0.5217
$\bar{y}_{34_x}$	0.7767	0.6087	$\bar{y}_{34_y}$	0.7476	0.6522
$\bar{y}_{35_x}$	0.7476	0.4783	$\bar{y}_{35_y}$	0.7087	0.3478
$\bar{y}_{45_x}$	0.7767	0.5652	$\bar{y}_{45_y}$	0.7087	0.3913

(c) $z$			(d) absolutt		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{13_z}$	0.6893	0.4783	$\bar{y}_{13}$	0.7184	0.4348
$\bar{y}_{14_z}$	0.7282	0.3913	$\bar{y}_{14}$	0.8058	0.5652
$\bar{y}_{15_z}$	0.7573	0.2174	$\bar{y}_{15}$	0.5534	0.5217
$\bar{y}_{34_z}$	0.7087	0.6087	$\bar{y}_{34}$	0.6311	0.6957
$\bar{y}_{35_z}$	0.5631	0.4783	$\bar{y}_{35}$	0.7767	0.4348
$\bar{y}_{45_z}$	0.6505	0.4348	$\bar{y}_{45}$	<b>0.7961</b>	<b>0.6522</b>

Tabell 4.7: Spesifisitet og sensitivitet, avstand mellom sensorer, KNN

(a) Avstand fra origo i xy-planet			(b) Avstand fra origo i xz-planet		
variabel	J	d	variabel	J	d
$\bar{y}_{1_{xy}}$	4.0949	0.5599	$\bar{y}_{1_{xz}}$	4.1078	0.6287
$\bar{y}_{3_{xy}}$	4.1574	0.8805	$\bar{y}_{3_{xz}}$	<b>4.1939</b>	<b>1.0503</b>
$\bar{y}_{4_{xy}}$	<b>4.1924</b>	<b>1.0439</b>	$\bar{y}_{4_{xz}}$	4.1642	0.9111
$\bar{y}_{5_{xy}}$	4.1041	0.6091	$\bar{y}_{5_{xz}}$	4.0217	0.1374
$\bar{y}_{6_{xy}}$	4.1673	0.9285	$\bar{y}_{6_{xz}}$	4.1232	0.7102

(c) Avstand fra origo i yz-planet			(d) Avstand fra origo		
variabel	J	d	variabel	J	d
$\bar{y}_{1_{yz}}$	4.0562	0.3448	$\bar{y}_1$	4.0909	0.5385
$\bar{y}_{3_{yz}}$	4.0979	0.5773	$\bar{y}_3$	4.1542	0.8644
$\bar{y}_{4_{yz}}$	<b>4.1408</b>	<b>0.7979</b>	$\bar{y}_4$	<b>4.1563</b>	<b>0.8738</b>
$\bar{y}_{5_{yz}}$	4.0132	0.0840	$\bar{y}_5$	4.0069	0.0446
$\bar{y}_{6_{yz}}$	4.0991	0.5826	$\bar{y}_6$	4.1094	0.6383

Tabell 4.8: Separabilitet, Avstand fra origo



(a) Avstand fra origo i xy-planet			(b) Avstand fra origo i xz-planet		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{1_{xy}}$	0.6019	0.5217	$\bar{y}_{1_{xz}}$	0.6408	0.5217
$\bar{y}_{3_{xy}}$	0.7184	0.4783	$\bar{y}_{3_{xz}}$	0.6214	0.6214
$\bar{y}_{4_{xy}}$	0.7184	0.5217	$\bar{y}_{4_{xz}}$	<b>0.6699</b>	<b>0.8261</b>
$\bar{y}_{5_{xy}}$	0.5437	0.2609	$\bar{y}_{5_{xz}}$	0.5437	0.6957
$\bar{y}_{6_{xy}}$	0.7767	0.5652	$\bar{y}_{6_{xz}}$	0.6699	0.6522

(c) Avstand fra origo i yz-planet			(d) Avstand fra origo		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{1_{yz}}$	0.5825	0.5217	$\bar{y}_1$	0.6117	0.5217
$\bar{y}_{3_{yz}}$	0.6019	0.5217	$\bar{y}_3$	0.6408	0.6957
$\bar{y}_{4_{yz}}$	0.4951	0.9130	$\bar{y}_4$	<b>0.6408</b>	<b>0.8696</b>
$\bar{y}_{5_{yz}}$	0.5825	0.3478	$\bar{y}_5$	0.4951	0.5652
$\bar{y}_{6_{yz}}$	0.7184	0.5652	$\bar{y}_6$	0.7184	0.5652

Tabell 4.9: Spesifisitet og sensitivitet, avstand fra origo, LDA

(a) Avstand fra origo i xy-planet			(b) Avstand fra origo i xz-planet		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{1_{xy}}$	0.8058	0.4348	$\bar{y}_{1_{xz}}$	0.8252	0.4348
$\bar{y}_{3_{xy}}$	0.6699	0.4348	$\bar{y}_{3_{xz}}$	<b>0.7087</b>	<b>0.7391</b>
$\bar{y}_{4_{xy}}$	0.7087	0.5217	$\bar{y}_{4_{xz}}$	0.7476	0.6522
$\bar{y}_{5_{xy}}$	0.6796	0.5652	$\bar{y}_{5_{xz}}$	0.6990	0.3043
$\bar{y}_{6_{xy}}$	0.8641	0.5217	$\bar{y}_{6_{xz}}$	<b>0.7961</b>	<b>0.6522</b>

(c) Avstand fra origo i yz-planet			(d) Avstand fra origo		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\bar{y}_{1_{yz}}$	0.7476	0.4783	$\bar{y}_1$	0.7670	0.4348
$\bar{y}_{3_{yz}}$	0.6893	0.5217	$\bar{y}_3$	0.7476	0.6522
$\bar{y}_{4_{yz}}$	0.7379	0.5217	$\bar{y}_4$	0.6893	0.6522
$\bar{y}_{5_{yz}}$	0.7573	0.2609	$\bar{y}_5$	0.6796	0.3043
$\bar{y}_{6_{yz}}$	<b>0.8058</b>	<b>0.6522</b>	$\bar{y}_6$	0.8155	0.4783

Tabell 4.10: Spesifisitet og sensitivitet, avstand fra origo, KNN

for LDA. Ingen av resultatene i dette tilfellet vise både høy spesifisitet og sensitivitet. Det beste resultatet oppnås ved å betrakte avstanden til sensor 4, enten i  $xz$ -planet, eller absolutt-avstanden. For KNN, Tabell 4.10, er det sensor 3 ( $x,z$ ) og sensor 6 for ( $x,z$ ) og ( $y,z$ ) som gir best resultat.

variabel	J	d
$\bar{y}_{1PCA}$	4.1344	0.7652
$\bar{y}_{2PCA}$	4.1316	0.7542
$\bar{y}_{3PCA}$	4.3036	1.5051
$\bar{y}_{4PCA}$	4.2031	1.0907
$\bar{y}_{5PCA}$	4.1006	0.5906
$\bar{y}_{6PCA}$	<b>4.3360</b>	<b>1.6259</b>

Tabell 4.11: Spesifisitet og sensitivitet, sensor 1-6, akse 1-3 projisert på prinsipalkomponenten

variabel	Spesifisitet	Sensitivitet
$y_{1PCA}$	0.6214	0.5217
$y_{2PCA}$	0.6796	0.4783
$y_{3PCA}$	<b>0.7379</b>	<b>0.7391</b>
$y_{4PCA}$	0.8350	0.4783
$y_{5PCA}$	0.6311	0.3913
$y_{6PCA}$	<b>0.7864</b>	<b>0.7826</b>

Tabell 4.12: Spesifisitet og sensitivitet, sensor 1-6, akse 1-3 projisert på prinsipalkomponenten, LDA

variabel	Spesifisitet	Sensitivitet
$y_{1PCA}$	0.7087	0.4348
$y_{2PCA}$	0.6408	0.4783
$y_{3PCA}$	0.8058	0.5652
$y_{4PCA}$	0.8835	0.3478
$y_{5PCA}$	0.7184	0.1739
$y_{6PCA}$	0.9126	0.5652

Tabell 4.13: Spesifisitet og sensitivitet, sensor 1-6, akse 1-3 projisert på prinsipalkomponenten, KNN

Istedenfor å betrakte sensorverdier langs en gitt akse i rommet, ser vi nå på sensorverdier i rommet projisert ned på den akse som forklarer mest av variansen i dataene. I Tabell 4.11 vises resultater fra separabilitetsanalysen og i Tabell 4.12 vises oppnådd spesifisitet og sensitivitet for LDA-algoritmen. For sensor 6 er resultatene veldig like som da  $y$ -aksen ble studert for denne sensoren. Dette er ikke overraskende om vi ser på egenvektoren:  $v = [0.0200.9990.039]^T$ , altså nesten kun i  $y$ -retning. Denne egenvektoren forklarer alene 68 % av variansen i dataene. Vi ser også at sensor 3(høyre arm) gir forholdsvis gode

resultater mens sensor 4 (venstre arm) kun har høy spesifisitet, og ikke sensitivitet. I Tabell 4.13 vises det at KNN-algoritmen ikke gir gode resultater i dette tilfellet.

Resultatene for SISO-modeller har vist at sensor 3, 4 og 6 skiller best mellom spedbarn med normale og avvikende bevegelsesmønstre. De to klassifikatorene, LDA og KNN, har ofte sine beste resultater på forskjellige utgangsvARIABLE. Dette er naturlig, da LDA som tidligere nevnt er en lineær klassifikator, mens KNN er ulineær.

Vi vil nå gå over til å se på MIMO-modeller.

#### 4.5.2 MIMO-modell

I denne seksjonen vil det presenteres resultater fra klassifisering der det er brukt MIMO-modeller i form av både AR-modeller og tilstandsrommodeller.

Vi vil først se på tilfellet der vi har 2 og 3 utganger, der utgangene representerer forskjellige akser på en gitt sensor.

(a) Sensor 1-6 $x$ og $y$			(b) Sensor 1-6 $x$ og $z$		
variabel	J	d	variabel	J	d
$y_{1_{xy}}$	16.3816	1.7839	$y_{1_{xz}}$	<b>16.6554</b>	<b>2.5577</b>
$y_{2_{xy}}$	16.3024	1.5032	$y_{2_{xz}}$	16.3554	1.6968
$y_{3_{xy}}$	16.4246	1.9254	$y_{3_{xz}}$	16.3457	1.6594
$y_{4_{xy}}$	16.3804	1.7795	$y_{4_{xz}}$	16.4293	1.9389
$y_{5_{xy}}$	16.2137	1.1375	$y_{5_{xz}}$	16.2662	1.3585
$y_{6_{xy}}$	<b>16.4994</b>	<b>2.1540</b>	$y_{6_{xz}}$	16.3729	1.7589

(c) Sensor 1-6 $y$ og $z$			(d) Sensor 1-6, $x$ , $y$ og $z$		
variabel	J	d	variabel	J	d
$y_{1_{yz}}$	16.2507	1.2944	$y_{1_{xyz}}$	<b>37.1019</b>	<b>3.3783</b>
$y_{2_{yz}}$	16.2140	1.1415	$y_{2_{xyz}}$	36.5324	2.2459
$y_{3_{yz}}$	16.2837	1.4281	$y_{3_{xyz}}$	36.5262	2.2264
$y_{4_{yz}}$	16.3800	1.7789	$y_{4_{xyz}}$	36.6811	2.6138
$y_{5_{yz}}$	16.1682	0.9305	$y_{5_{xyz}}$	36.5791	2.3676
$y_{6_{yz}}$	<b>16.5284</b>	<b>2.2349</b>	$y_{6_{xyz}}$	36.7984	2.8687

Tabell 4.14: Separabilitet, 2 utganger: 2 og 3 akser på en sensor

Ved MIMO 4.ordens AR-modeller blir antall variable 16 med 2 utgangsvARIABLE og 36 med 3 utgangsvARIABLE. Generelt sett viser separabilitetsanalysen for MIMO-modeller mindre sammenheng med resultatene for spesifisitet og sensitivitet. Av Tabell 4.14 og 4.15 ser vi at høye verdier på kriteriene  $J$  og  $d$  ofte svarer til rimelig høy spesifisitet, men med lav sensitivitet. I det etterfølgende vil det derfor fokuseres på verdiene på spesifisitet

(a) Sensor 1-6, $x$ og $y$			(b) Sensor 1-6, $x$ og $z$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$y_{1_{xy}}$	0.7476	0.4783	$y_{1_{xz}}$	<b>0.7961</b>	<b>0.6957</b>
$y_{2_{xy}}$	0.6990	0.4783	$y_{2_{xz}}$	0.8155	0.4348
$y_{3_{xy}}$	0.7767	0.5652	$y_{3_{xz}}$	0.6117	0.6522
$y_{4_{xy}}$	<b>0.7961</b>	<b>0.6522</b>	$y_{4_{xz}}$	<b>0.7961</b>	<b>0.7826</b>
$y_{5_{xy}}$	0.5922	0.3478	$y_{5_{xz}}$	0.6505	0.4783
$y_{6_{xy}}$	0.8447	0.5652	$y_{6_{xz}}$	0.8738	0.5652

(c) Sensor 1-6, $y$ og $z$			(d) Sensor 1-6, $x, y$ og $z$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$y_{1_{yz}}$	<b>0.6893</b>	<b>0.6957</b>	$y_{1_{xyz}}$	0.8641	0.5217
$y_{2_{yz}}$	0.6019	0.4348	$y_{2_{xyz}}$	0.7379	0.4783
$y_{3_{yz}}$	0.6311	0.6957	$y_{3_{xyz}}$	0.6699	0.5652
$y_{4_{yz}}$	<b>0.6214</b>	<b>0.9130</b>	$y_{4_{xyz}}$	<b>0.7961</b>	<b>0.7391</b>
$y_{5_{yz}}$	0.5922	0.4348	$y_{5_{xyz}}$	0.7087	0.4783
$y_{6_{yz}}$	0.8835	0.4348	$y_{6_{xyz}}$	0.9126	0.4783

Tabell 4.15: Spesifisitet og sensitivitet, 2 utganger: 2 og 3 akser på en sensor, LDA

(a) Sensor 1-6, $x$ og $y$			(b) Sensor 1-6, $x$ og $z$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$y_{1_{xy}}$	0.8252	0.5217	$y_{1_{xz}}$	0.6990	0.6522
$y_{2_{xy}}$	0.6505	0.3478	$y_{2_{xz}}$	0.4466	0.7391
$y_{3_{xy}}$	0.7573	0.5652	$y_{3_{xz}}$	<b>0.6602</b>	<b>0.7826</b>
$y_{4_{xy}}$	<b>0.8350</b>	<b>0.7391</b>	$y_{4_{xz}}$	<b>0.8641</b>	<b>0.6522</b>
$y_{5_{xy}}$	0.6893	0.3913	$y_{5_{xz}}$	0.7184	0.3043
$y_{6_{xy}}$	0.7767	0.3478	$y_{6_{xz}}$	<b>0.7864</b>	<b>0.6957</b>

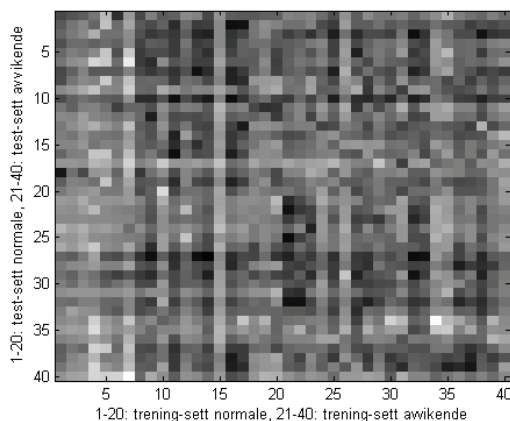
  

(c) Sensor 1-6, $y$ og $z$			(d) Sensor 1-6, $x, y$ og $z$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$y_{1_{yz}}$	0.6699	0.3913	$y_{1_{xyz}}$	<b>0.8155</b>	<b>0.6957</b>
$y_{1_{yz}}$	0.7379	0.4348	$y_{2_{xyz}}$	0.7670	0.3043
$y_{1_{yz}}$	0.5825	0.6522	$y_{3_{xyz}}$	0.7282	0.6522
$y_{1_{yz}}$	<b>0.7767</b>	<b>0.6957</b>	$y_{4_{xyz}}$	0.7961	0.5652
$y_{1_{yz}}$	0.6505	0.3913	$y_{5_{xyz}}$	0.6796	0.3913
$y_{1_{yz}}$	0.7767	0.3478	$y_{6_{xyz}}$	0.7282	0.3913

Tabell 4.16: Spesifisitet og sensitivitet, 2 og 3 akser på en sensor, KNN

og sensitivitet, mens separabilitetsresultatene finnes i tillegg A. Av Tabell 4.15 ser vi at de beste resultatene forekommer for sensor 4, men også sensor 1 har gode verdier når  $x$  og  $z$  er utgangsvARIABLE. Det er interessant å se at sensor 1 (venstre ankel) skiller konsekvent bedre enn sensor 5 (høyre ankel) når vi har 2 utgangsvARIABLE, både når det gjelder separabilitetsanalysen og spesifisiteten/sensitiviteten. Det samme gjelder til en viss grad også sensor 4(venstre håndledd) og sensor 3 (høyre håndledd).

I Tabell 4.16 vises samme analyse for KNN brukt på tilstandsrommodeller. Også her skiller sensor 1 bedre enn 5. De beste resultatene fås for sensor 4, men også sensor 1 og sensor 6 viser gode resultater når henholdsvis  $(x, y, z)$  og  $(x, z)$  betraktes. I Figur 4.8 illustreres avstanden mellom modeller fra treningssett og testsett ved bruk av metrikken  $d_m$ . Mørk piksel betyr liten avstand mellom 2 modeller. Ideelt sett burde øvre venstre kvadrat(0-20,0-20), og nedre høyre kvadrat(21-40,21-40) være mørke, da dette ville bety at de normale i trening-settet ligner på de normale i testsettet, og samme for de avvikende.



Figur 4.8: Avstand mellom modeller med metrikken  $d_m$

I stedet for å betrakte 2 eller 3 gitte akser i rommet, vil vi nå betrakte data projisert ned på 2 prinsipale komponenter.

variabel	Spesifisitet	Sensitivitet
$\mathbf{y}_{1PCA}$	0.7379	0.4348
$\mathbf{y}_{2PCA}$	0.6796	0.4783
$\mathbf{y}_{3PCA}$	0.7476	0.5217
$\mathbf{y}_{4PCA}$	<b>0.7961</b>	<b>0.6957</b>
$\mathbf{y}_{5PCA}$	0.6214	0.4783
$\mathbf{y}_{6PCA}$	0.8350	0.5652

Tabell 4.17: Sensor 1-6, projisert ned på 2 prinsipale komponenter, LDA

variabel	Spesifisitet	Sensitivitet
<b>YPCA</b>	0.7379	0.5217
<b>YPCA</b>	0.6311	0.3913
<b>YPCA</b>	0.8447	0.6087
<b>YPCA</b>	0.8155	0.5652
<b>YPCA</b>	0.6990	0.2609
<b>YPCA</b>	0.7767	0.3478

Tabell 4.18: Sensor 1-6, projisert ned på 2 prinsipale komponenter, KNN

I Tabell 4.17 vises det at best verdi på spesifisitet og sensitivitet for LDA fåes for sensor 4. I Tabell 4.18 vises resultater for k-nærmeste-nabo algoritmen, som ikke gir nevneverdig gode resultater i dette tilfellet.

Vi betrakter så utgangsvariabelen  $y$  bestående av 2 utganger, der utgangene representerer samme akse på forskjellige sensorer.

(a) $x$			(b) $y$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
<b>Y13<sub>x</sub></b>	<b>0.7379</b>	<b>0.7391</b>	Y13 <sub>y</sub>	0.7864	0.5652
Y14 <sub>x</sub>	0.7670	0.5652	Y14 <sub>y</sub>	0.7379	0.6087
Y15 <sub>x</sub>	0.7476	0.4783	Y15 <sub>y</sub>	0.6893	0.3913
Y34 <sub>x</sub>	0.4348	0.8155	Y34 <sub>y</sub>	0.5652	0.6602
Y35 <sub>x</sub>	0.7087	0.4783	Y35 <sub>y</sub>	0.7087	0.5652
Y45 <sub>x</sub>	0.7476	0.4348	Y45 <sub>y</sub>	0.5922	0.6087

(c) $z$		
variabel	Spesifisitet	Sensitivitet
Y13 <sub>z</sub>	0.5922	0.6087
Y14 <sub>z</sub>	0.5631	0.8261
Y15 <sub>z</sub>	0.6990	0.3478
<b>Y34<sub>z</sub></b>	<b>0.8696</b>	<b>0.6311</b>
Y35 <sub>z</sub>	0.7282	0.4783
Y45 <sub>z</sub>	0.5728	0.6087

Tabell 4.19: Spesifisitet og sensitivitet, 2 utganger: Samme akse på 2 forskjellige sensorer, LDA

Ut i fra Tabell 4.19 ser vi at beste kombinasjon på spesifisitet og sensitivitet for LDA forekommer når modellene identifiseres ut i fra  $x$ -retningen på sensor 1 og sensor 3. Sensor 3 og 4 i  $z$ -retning viser høy spesifisitet men noe lav sensitivitet.

I Tabell 4.20 vises resultater fra K-nærmeste-nabo algoritmen. Vi ser at de beste resultatene forekommer på andre utgangsvARIABLE enn hva som var tilfellet ved LDA.

(a) $x$			(b) $y$		
variabel	Spesifisitet	Sensitivitet	variabel	Spesifisitet	Sensitivitet
$\mathbf{y13_x}$	0.6117	0.6522	$\mathbf{y13_y}$	0.8058	0.3913
$\mathbf{y14_x}$	0.8155	0.4783	$\mathbf{y14_y}$	0.8447	0.6087
$\mathbf{y15_x}$	0.6796	0.6087	$\mathbf{y15_y}$	0.8058	0.4348
$\mathbf{y34_x}$	0.8252	0.6087	$\mathbf{y34_y}$	<b>0.7282</b>	<b>0.6957</b>
$\mathbf{y35_x}$	0.6214	0.6087	$\mathbf{y35_y}$	0.7087	0.4783
$\mathbf{y45_x}$	<b>0.7379</b>	<b>0.6957</b>	$\mathbf{y45_y}$	0.6990	0.6087

(c) $z$		
variabel	Spesifisitet	Sensitivitet
$\mathbf{y13_z}$	0.5243	0.5217
$\mathbf{y14_z}$	0.5922	0.7391
$\mathbf{y15_z}$	0.8544	0.3478
$\mathbf{y34_z}$	0.5728	0.7391
$\mathbf{y35_z}$	0.6505	0.6957
$\mathbf{y45_z}$	0.6796	0.5652

Tabell 4.20: Spesifisitet og Sensitivitet, 2 utganger: Samme akse på 2 forskjellige sensorer, KNN

Vi kan også la de to utgangene representere måledata projisert ned på den prinsipale komponenten for to forskjellige sensorer. Prinsipalkomponenten for en gitt sensor regnes ut i fra alle observasjoner fra treningsdata tilhørende de to gruppene. Prinsipalkomponentene for de to sensorene regnes altså ut hver for seg.

variabel	Spesifisitet	Sensitivitet
$\mathbf{y13_{PCA}}$	0.7767	0.6087
$\mathbf{y14_{PCA}}$	0.8058	0.5217
$\mathbf{y15_{PCA}}$	0.7670	0.4348
$\mathbf{y34_{PCA}}$	0.7573	0.6522
$\mathbf{y35_{PCA}}$	0.6990	0.5652
$\mathbf{y45_{PCA}}$	0.7670	0.5217

Tabell 4.21: Spesifisitet og sensitivitet, sensorpar, projisert ned på den prinsipale komponenten, LDA

Av Tabell 4.21 og 4.22 ser vi at ingen av sensor-parene viser spesielt gode verdier.

Resultatene i denne seksjonen viser at det også kan være nyttig å betrakte MIMO-modeller for å skille mellom de to gruppene. Sensorene som vi så var viktige for SISO-modeller, er fortsatt viktige for MIMO-modeller. I tillegg viser sensor 1 (venstre ankel) i noen tilfeller gode verdier på spesifisitet og sensitivitet, mens dette ikke gjelder for

variabel	Spesifisitet	Sensitivitet
$\mathbf{Y}_{13_{PCA}}$	0.5146	0.5217
$\mathbf{Y}_{14_{PCA}}$	0.5922	0.7391
$\mathbf{Y}_{15_{PCA}}$	0.8544	0.3478
$\mathbf{Y}_{34_{PCA}}$	0.5728	0.7391
$\mathbf{Y}_{35_{PCA}}$	0.6505	0.6957
$\mathbf{Y}_{45_{PCA}}$	0.6796	0.5652

Tabell 4.22: Spesifisitet og sensitivitet, sensorpar, projisert ned på den prinsipale komponenten, KNN

sensor 5(høyre ankel). Dette ble også observert for SISO-modeller, og det kan tyde på at venstre ben er viktigere enn høyre ben, når de skal betraktes hver for seg.

Resultatene for både SISO- og MIMO-modeller viser også at bevegelse i vertikalplanet ( $xz$ -planet) kan være viktig å betrakte i tillegg til bevegelse i horisontalplanet ( $xy$ -planet) når vi skal klassifisere bevegelser som normale eller avvikende.

Vi skal nå se på om det er mulig å forbedre noen av de beste resultatene ved å kombinere egenskapsvektorene fra forskjellige utgangsvARIABLE, som forklart i seksjon 3.5.5. For både KNN og LDA ble det testet ut 2 tilfeller basert på å kombinere egenskapsvektorene.

For KNN-algoritmen prøver vi først å kombinere de 3 beste resultatene oppnådd i Tabell 4.4(b) (sensor 3,4 og 6, akse  $z$ ). Dette førte til en spesifisitet på 0.5728%, og en sensitivitet på 0.8261%, altså et dårligere resultat.

Så testes det samme for MIMO-systemer, der de 3 beste resultatene fra Tabell 4.16 (b) kombineres(sensor 3,4, og 6, akser  $x$  og  $z$ ). Igjen ble resultatet dårligere, med en spesifisitet på 61% og en sensitivitet på 65%.

Vi prøver så å kombinere gode resultater funnet ved LDA, der vi først ser på SISO-modeller. De to beste resultatene i dette tilfellet ble funnet ved å se på hodet langs  $y$ -aksen, se Tabell 4.3 (b), og ved å betrakte avstanden mellom armene langs  $x$ -aksen, se Tabell 4.6(a). Systemparameterne funnet for hver enkelt ROI, fra modeller funnet fra de to utgangsvARIABLE, kombineres nå i en egenskapsvektor. Dette førte til en spesifisitet på 86.96% og en sensitivitet på 85.44%, noe som må sies å være et veldig godt resultat. For å validere dette resultatet ble treningssettet gjort om til testsett, og omvendt. Dette førte til en spesifisitet på 86.87% og en sensitivitet på 88.00%.

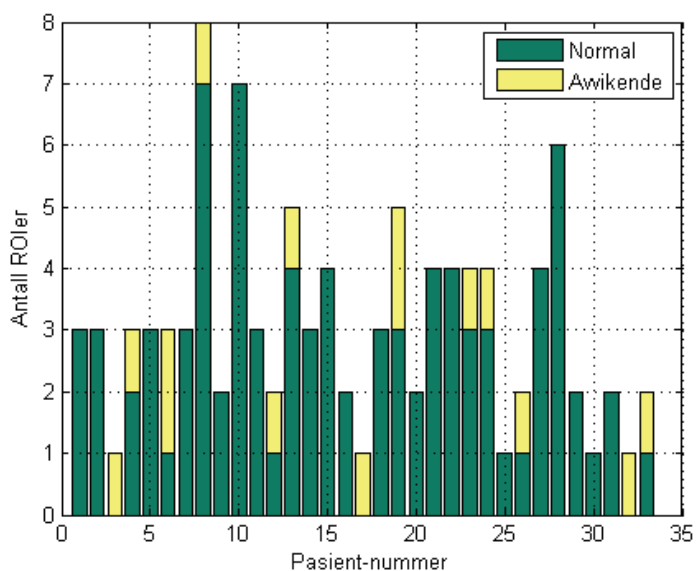
I tilfellet med MIMO AR-modeller kombineres sensor 1 og 4 langs  $x$  og  $z$ , se Tabell 4.15(b). Dette gir en spesifisitet på 0.8447% og en sensitivitet på 0.7826. Vi ser at vi ved å kombinere egenskapsvektorene klarte å beholde den beste verdien på sensitiviteten, og samtidig øke spesifisiteten.

Hittil har vi studert klassifisering på ROI-nivå, og vi vil nå gå over til å studere klas-



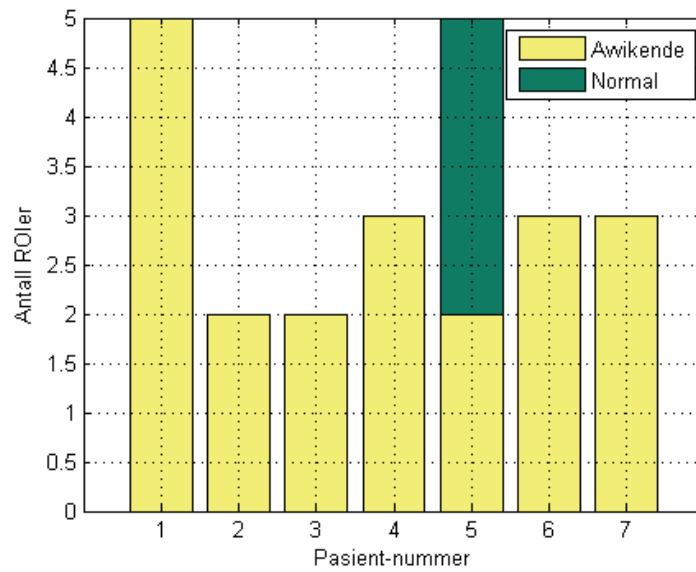
sifisering på pasient-nivå. For å studere klassifisering på pasient-nivå, brukes vektingen av ROI'er diskutert i seksjon 4.4. Vi vil nå studere de to tilfellene fra ROI-analysen som gav best resultat.

Det første tilfellet som studeres er kombinasjonen av SISO-modeller nevnt ovenfor (kombinasjon av hode og avstand mellom armer), der vi på ROI-nivå oppnådde en spesifisitet på 86.96% og en sensitivitet på 85.44%. Figur 4.9 og 4.10 viser klassifiseringsresultatene for ROI'er til hver pasient, for henholdsvis den normale og avvikende gruppen. Figur 4.11 viser et ROC-plott for klassifisering av pasienter der vektingen av ROI'er varieres. Hva som er det beste resultatet kommer an på hva man vil oppnå. 100% sensitivitet oppnåes ved å sette vektparameteren til  $w = 1.5849$ , men dette gir en spesifisitet på bare 75.76%. Det punktet som har den minste euklidske avstanden til den optimale verdien i ROC-plottet har en spesifisitet på 90.91%(30/33) og en sensitivitet på 85.71%(6/7). Tabell 4.23 viser de beste verdiene i ROC-plottet og tilhørende verdi på vektparameteren.

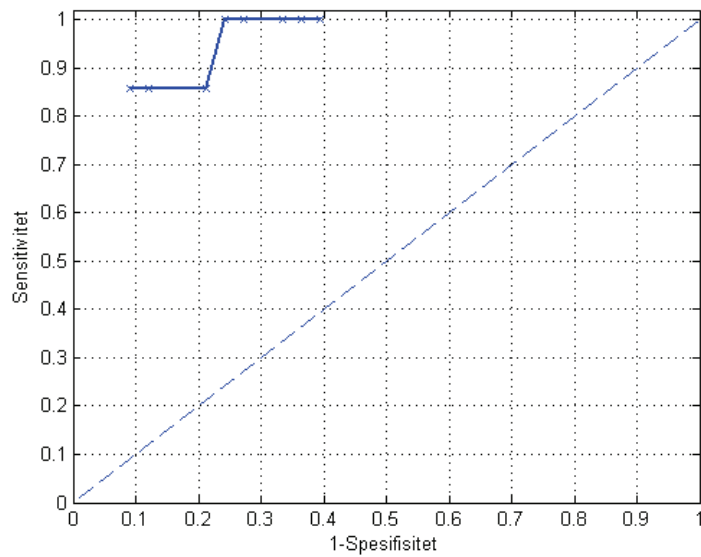


Figur 4.9: ROI'er klassifisert som normale eller avvikende for normal gruppe

Det neste tilfelle som analyseres er ved bruk av MIMO AR-modeller nevnt ovenfor (sensor 1 og 4 langs  $x$  og  $z$ ), der det ble oppnådd en spesifisitet på 0.8447% og en sensitivitet på 0.7826. Figur 4.12 og 4.13 viser klassifiseringsresultatene for ROI'er til hver pasient, for henholdsvis den normale og avvikende gruppen. Figur 4.14 viser et ROC-plott for klassifisering av pasienter der vektingen av ROI'er varieres. I Tabell 4.24 vises de beste verdiene i ROC-plottet og tilhørende verdi på vektparameteren. Vi ser at spesifisiteten nå må ned i 72.73% for å få 100% sensitivitet. Dette er også det punktet som ligger nærmest det optimale punktet i ROC-plottet.



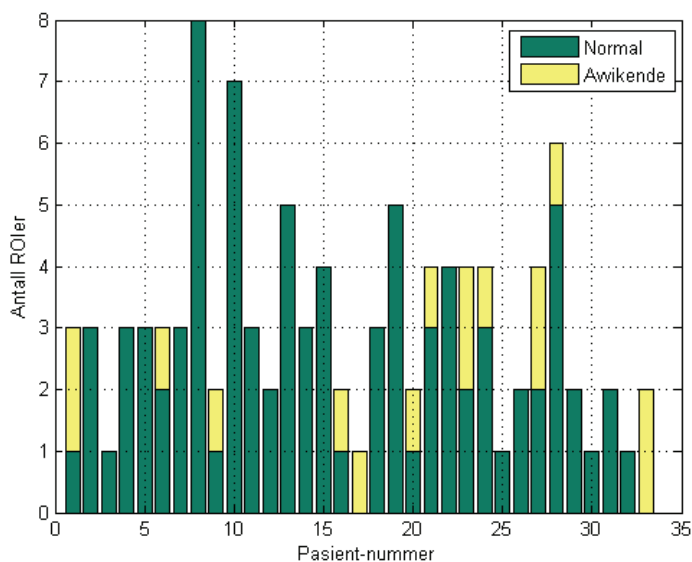
Figur 4.10: ROI'er klassifisert som normale eller avvikende for avvikende gruppe



Figur 4.11: ROC-plot for vekting av ROI'er, sensor 6, Akse2

Spesifisitet	Sensitivitet	Vektparameter
0.9091	0.8571	0.3981
0.8788	0.8571	0.5012
0.8788	0.8571	0.6310
0.8788	0.8571	0.7943
0.8788	0.8571	1.0000
0.7879	0.8571	1.2589
0.7576	1.0000	1.5849

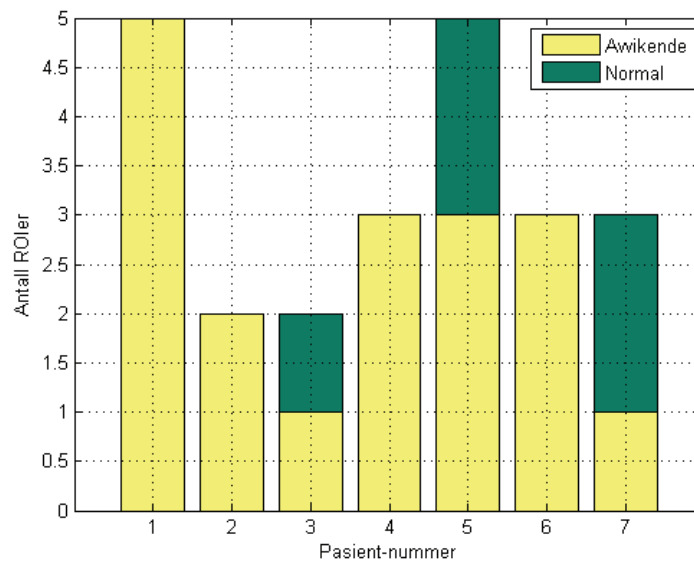
Tabell 4.23: Spesifisitet og sensitivitet på pasientnivå



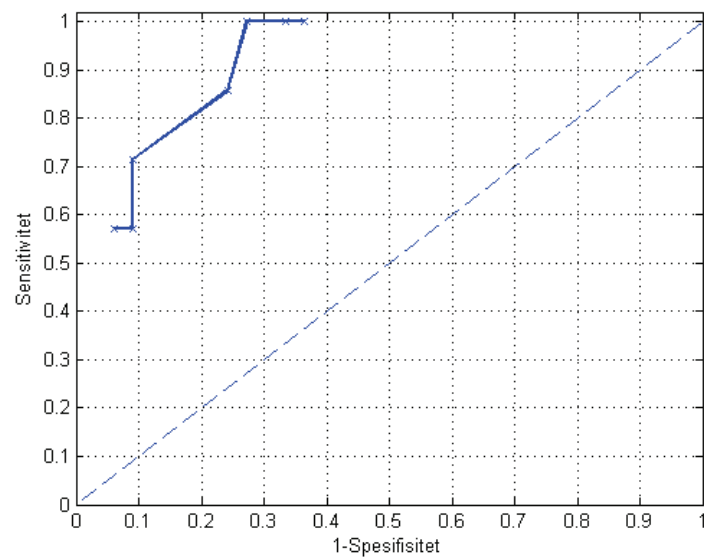
Figur 4.12: ROI'er klassifisert som normale eller avvikende for normal gruppe

Spesifisitet	Sensitivitet	Vektparameter
0.9091	0.7143	0.7943
0.9091	0.7143	1.0000
0.7576	0.8571	1.2589
0.7576	0.8571	1.5849
0.7576	0.8571	1.9953
0.7273	1.0000	2.5119

Tabell 4.24: Spesifisitet og sensitivitet på pasientnivå



Figur 4.13: ROI'er klassifisert som normale eller avvikende for avvikende gruppe



Figur 4.14: ROC-plot for vekting av ROI'er, sensor 6, akse  $y$

## 4.6 Endring av orden og metode

Vi så i seksjon 4.1 at det var fornuftig ut i fra et modelltilpasnings-perspektiv å bruke modeller av 4. orden. Det er dermed ikke sikkert at dette var optimalt med tanke på den endelige klassifiseringen. I denne seksjonen vil vi derfor studere de to beste resultatene fra SISO-tilfellet med AR-modeller, og studere hvordan spesifisitet og sensitivitet endrer seg når vi:

- Endrer ordenen på AR-modellen
- Endrer metode for systemidentifikasjon
- Innfører mini-ROI'er isteden for ROI'er

### Endring av orden

I Tabell 4.25 vises klassifiseringsresultater for modeller fra de to utgangsvariablene når vi varierer ordenen til modellene. Figur 4.15 viser ROC-plott for de samme tallene. Selv om man får enda litt bedre resultatet i tilfellet  $y_{6_y}$  ved å velge en litt høyere orden, ser det ut som en 4.ordens modell er et fornuftig valg.

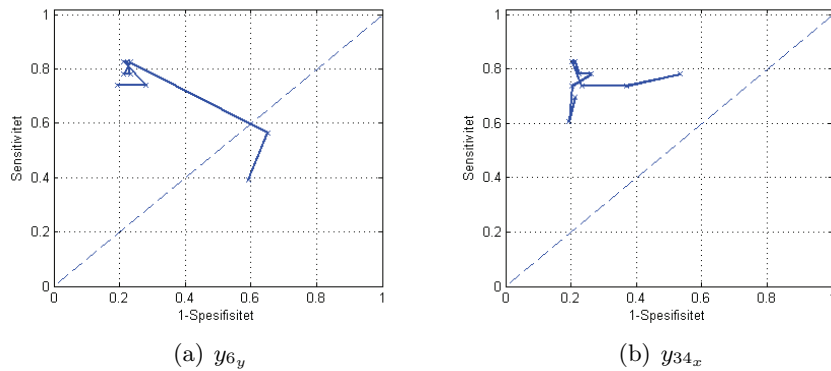
(a) $y_{6_y}$			(b) $y_{34_x}$		
Orden	Spesifisitet	Sensitivitet	Orden	Spesifisitet	Sensitivitet
1	0.4078	0.3913	1	0.4660	0.7826
2	0.3495	0.5652	2	0.6311	0.7391
3	0.7670	0.8261	3	0.7670	0.7391
4	0.7864	0.7826	4	0.7961	0.8261
5	0.7670	0.7826	5	0.7864	0.8261
6	0.7767	0.8261	6	0.7767	0.7826
8	0.7864	0.8261	8	0.7379	0.7826
10	0.7864	0.8261	10	0.7961	0.7391
15	0.7184	0.7391	15	0.8058	0.6087
20	0.8058	0.7391	20	0.7864	0.6957

Tabell 4.25: Spesifisitet og sensitivitet ved endring av orden på AR-modell

### Endring av systemidentifikasjonsmetode

I seksjon 4.5 ble minste kvadraters metode brukt for å identifisere AR-modeller. Vi skal nå se på effekten av å endre metoden til “yule walker”, “forward backward” og “burg” for 4.ordens AR-modeller. Av Tabell 4.26 ser vi at valg av metode ikke påvirker resultatene i stor grad, med unntaket av metoden Yule-Walker for utgangsvariabel  $y_{6_y}$ . En nærmere analyse av årsaken til dette ble det ikke funnet tid til i dette arbeidet.

I seksjon 4.5 ble alle SISO-modeller modelert som AR-modeller, der klassifikatorene KNN (basert på  $d_m$ -metrikken) og LDA ble brukt på systemparameterne. Vi kan også bruke



Figur 4.15: ROC-plott: Endring av orden

(a) $y_{6y}$			(b) $y_{34x}$		
Metode	Spesifisitet	Sensitivitet	Metode	Spesifisitet	Sensitivitet
MK	0.7864	0.7826	MK	0.7961	0.8261
YW	0.6602	0.6522	YW	0.7961	0.7826
FB	0.7864	0.7826	FB	0.7961	0.8261
Burg	0.7864	0.7826	Burg	0.7961	0.8261

Tabell 4.26: Spesifisitet og sensitivitet ved endring av systemidentifikasjonsmetoder

en tilstandsrommodell, sammen med KNN. I dette tilfellet blir resultatene som vist i Tabell 4.27.

Variabel	Spesifisitet	Sensitivitet
$y_{6_y}$	0.8544	0.6087
$y_{34_x}$	0.7670	0.6957

Tabell 4.27: Spesifisitet og sensitivitet for SISO tilstandsrommodeller, klassifisert med KNN

### Segmentering

Vi vil nå se på tilfellet der vi deler opp hver enkelt ROI i segmenter. Vi vil studere resultater med segmentlengder på 30, 60, 120 og 240 sekunder. Rester av ROI'er under segmentstørrelsen blir forkastet, såfremt ROI'en er lengre enn segmentlengden. I tilfellet der ROI-lengden er kortere enn segmentlengden, brukes ROI-en slik den er. Av Tabell 4.28 ser vi at resultatene blir bedre jo lengre segmenter vi betrakter. Dette kan ha sammenheng med klassifikatoren som er lineær(LDA). Ved bruk av korte segmenter, kan det hende at en enkelt klasse i større grad fordeler seg i forskjellige grupperinger, som ikke fanges opp av den lineære klassifikatoren. Det er derfor interessant å test ut den ulineære KNN-klassifikatoren for AR-modeller i dette tilfellet. Resultatene vises i 4.29, og vi ser segmenteringen har liten hensikt også i dette tilfellet.

(a)  $y_{6_y}$

Segmentlengde(s)	Spesifisitet	Sensitivitet
30	0.6071	0.7294
60	0.6458	0.7683
120	0.6779	0.7805
240	0.7664	0.8462

(b)  $y_{34_x}$

Segmentlengde(s)	Spesifisitet	Sensitivitet
30	0.6554	0.7176
60	0.6863	0.7683
120	0.7450	0.8780
240	0.7664	0.8846

Tabell 4.28: Spesifisitet og sensitivitet ved endring av segmentlengde, LDA

### ARMA-modell

Vi vil nå prøve å utvide AR(4)-modellen til en ARMA(4,2)-modell, for å se om dette påvirker resultatet. I dette tilfellet brukes en annen systemidentifikasjonsmetode, basert på prediksjonsfeil(`armax`). Av Tabell 4.30 ser vi at resultatene blir dårligere enn ved en ren AR-modell.

(a)  $y_{6_y}$

Segmentlengde(s)	Spesifisitet	Sensitivitet
30	0.6125	0.6412
60	0.6494	0.6220
120	0.6913	0.5854
240	0.8598	0.5385

(b)  $y_{34_x}$

Segmentlengde(s)	Spesifisitet	Sensitivitet
30	0.6732	0.6529
60	0.7122	0.5976
120	0.7450	0.6829
240	0.7664	0.6923

Tabell 4.29: Spesifisitet og sensitivitet ved endring av segmentlengde, KNN

Variabel	Spesifisitet	Sensitivitet
$y_{6_y}$	0.7767	0.3913
$y_{34_x}$	0.8252	0.5652

Tabell 4.30: Spesifisitet og sensitivitet for ARMA(4,2)-modell

## 4.7 Analyse av modell

I denne seksjonen vil vi se litt nærmere på noen av SISO-modellene som i seksjon 4.5 viste seg å skille mellom de to gruppene. Vi vil studere AR-modeller funnet ut i fra utgangsvariablene  $y_{6_y}$  og  $y_{34_x}$  med minste kvadraters metode. Modeller tilhørende den normale og avvikende gruppen vil analyseres etter 4 punkter:

1. Gjennomsnitt på AR-parameterne
2. Effekt-tetthetsspektre
3. Impulsrespons
4. Poler

Vi vil først betrakte  $y_{6_y}$ . I Tabell 4.31 vises gjennomsnitt av de 4 AR-parameterne beregnet over de to gruppene. At gjennomsnittet er forskjellig for de to gruppene er forventet basert på separabilitetsanalysen gjort tidligere i kapitlet.

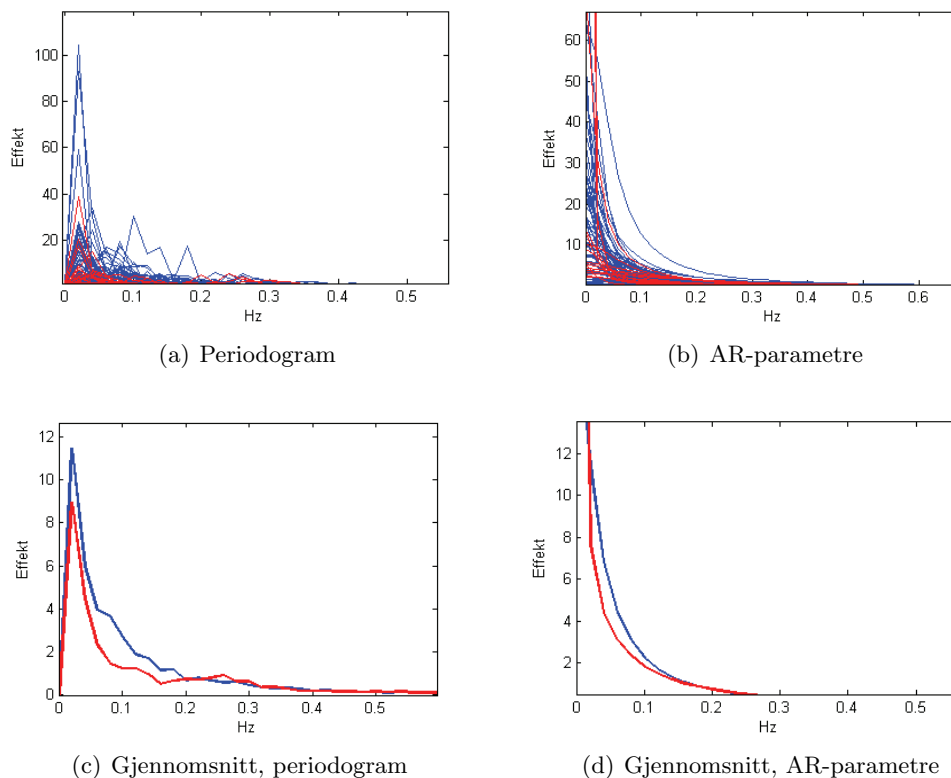
I Figur 4.16 vises effekt-tetthetsspektre for spedbarn med normale(blå) og avvikende (rød) bevegelser. De to øverste figurene viser henholdsvis spektre funnet ved periodogram(a)



Gruppe	AR(1)	AR(2)	AR(3)	AR(4)
Normal	-2.0571	1.2493	-0.1168	-0.0720
Avvikende	-2.0265	1.1445	-0.0449	0.0694

Tabell 4.31: Gjennomsnitt AR-parametre,  $y_{6,y}$ 

og AR-parametre(b). Hver graf representerer en pasient, og er et gjennomsnitt av effekt-tetthetsspektre til alle ROI'ene tilhørende pasienten. Figurene *b* og *c* viser gjennomsnittet over alle pasientene.

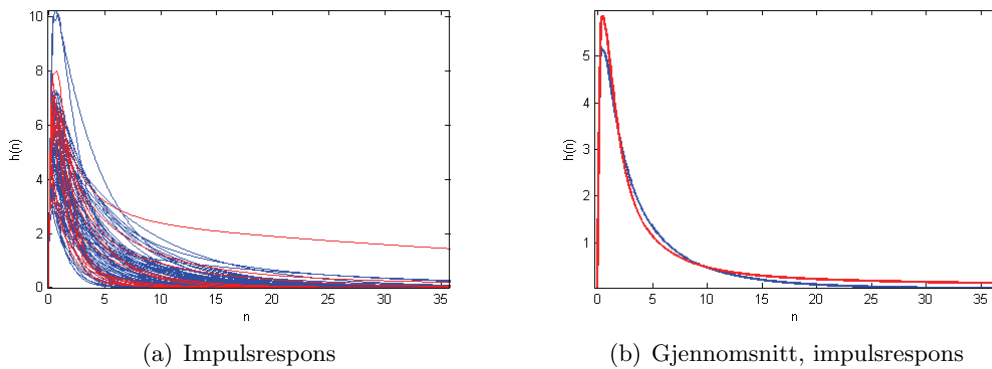


Figur 4.16: Effekt-tetthetsspektre for spedbarn med normale(blå) og avvikende(rød) bevegelser,  $y_{6,y}$

Det er et par ting vi kan merke oss med Figur 4.16. For det første ser vi at den avvikende gruppen har lavere effekt ved lave frekvenser. Dette vises både av spektrene basert på periodogram og de basert på AR-parametre. At de to metodene viser det samme er med på å verifisere at AR-modeleringen er fornuftig. Det virker også som den avvikende gruppen ligger relativt samlet. Det skal her nevnes at de røde grafene er plottet over de blå, slik at de to gruppene kan virke mer atskilte enn de egentlig er. En annen ting som bør bemerkes er at effekten er størst på veldig lave frekvenser for begge gruppene,  $0.1Hz$

tilsvarer en svingning per tiende sekund.

Det neste steget i analysen er å se på impulsresponsen til modellene. I utgangspunktet representerer AR-modellene brukt i denne oppgaven stokastiske prosesser, som ikke har noen definert impulsrespons, da vi ikke har noen deterministisk inngang på modellen. AR-modeller med en deterministisk inngang kalles ofte for ARX-modeller. Vi kan likevel finne impulsresponsen til en AR-modell ved å betrakte inngangen som deterministisk. Figur 4.17 viser impulsresponsen for de to gruppene. Det er ingen stor skilnad mellom de to gruppene, men hvis vi skal lete etter en forskjell kan det se ut som den avvikende gruppen oftere har en brattere impulsrespons.

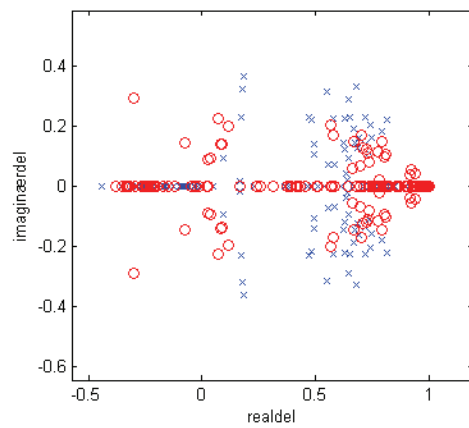


Figur 4.17: Impulsrespons for spedbarn med normale (blå) og avvikende (rød) bevegelser,  $y_{6y}$

I Figur 4.18 vises poler for modeller identifisert fra 40 ROI'er, for både normale og avvikende barn. De komplekse polene for friske og avvikende ligger noe atskilt, mens poler langs real-aksen ikke gjør det samme (igjen: rød er plottet til slutt). Merk at det er 4 poler (4.ordens modell) per modell. Det er mulighet for å brukes poler direkte i klassifisering, men siden de 4 polene ikke har noen bestemt rekkefølge, må de sorteres etter et gitt kriterium først hvis dette skal ha noen hensikt. En slik sortering kan by på problemer, da modellene ikke nødvendigvis har like mange komplekskonjugerte polpar. Siden det er under 40 komplekskonjugerte røde polpar i Figur 1, er det noen modeller som ikke har komplekskonjugerte poler. Skal poler brukes som egenskapsvektor holder det naturligvis å betrakte en av de komplekskonjugerte polene.

Det vil nå foretas en tilsvarende analyse for utgangsvariabelen  $y_{34x}$ . Av Tabell 4.32 ser vi at det som forventet er forskjell i gjennomsnitt på AR-parameterne.

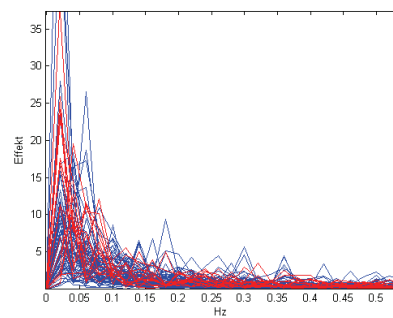
Figur 4.19 og Figur 4.20 viser henholdsvis effektetthetsspektre og impulsresponsen til modellene. Den avvikende gruppen har gjennomsnittlig mindre energi på lavere frekvenser enn den normale. Impulsresponsene til de to gruppene er ikke vesentlig ulike i dette tilfellet. Hvis vi derimot studerer modellenes poler i Figur 4.21, ser vi at det er en viss gruppering mellom den normale og avvikende gruppen.



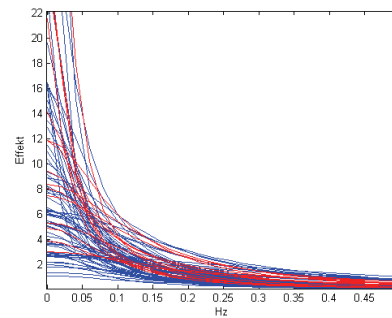
Figur 4.18: Polplott for 40 ROI'er tilhørende spedbarn med normale (blå) og avvikende (rød) bevegelser,  $y_{6_y}$

Gruppe	AR(1)	AR(2)	AR(3)	AR(4)
Normal	-1.9474	1.3029	-0.3185	-0.0225
Avvikende	-2.0093	1.3585	-0.3480	0.0104

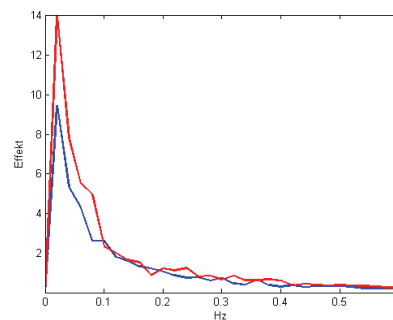
Tabell 4.32: Gjennomsnitt AR-parametere,  $y_{34_x}$



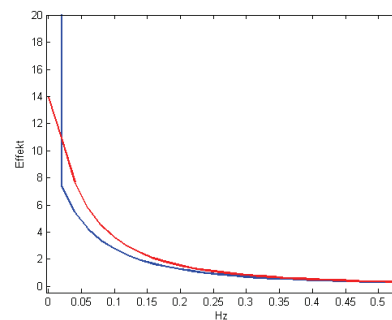
(a) Periodogram



(b) AR-parametere

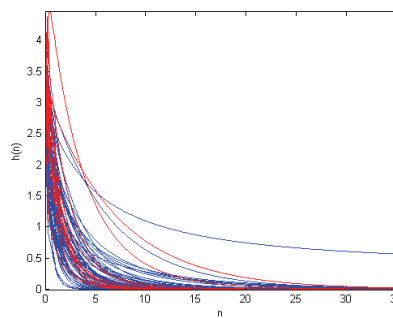


(c) Gjennomsnitt, periodogram

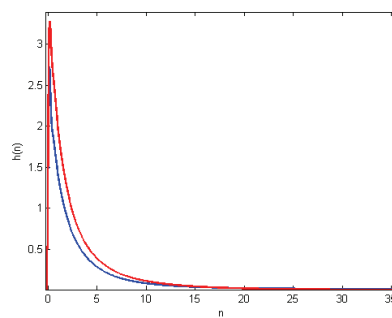


(d) Gjennomsnitt, AR-parametere

Figur 4.19: Effekt-tetthetsspektre for spedbarn med normale(blå) og avvikende(rød) bevegelser,  $y_{34_x}$

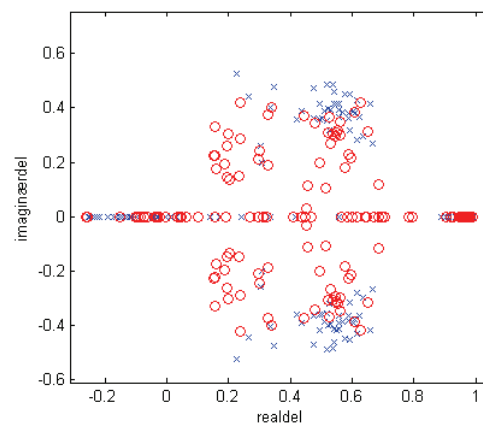


(a) Impulsrespons



(b) Gjennomsnitt, impulsrespons

Figur 4.20: Impulsrespons for spedbarn med normale(blå) og avvikende(rød) bevegelser,  $y_{34_x}$



Figur 4.21: Polplott for 40 ROI'er tilhørende spedbarn med normale(blå) og avvikende(rød) bevegelser,  $y_{34x}$



## Kapittel 5

# Diskusjon

Vi så i forrige kapittel at det for SISO modeller ble oppnådd en spesifisitet på 90.91% og en sensitivitet på 85.71% når det gjelder klassifisering på pasientnivå. Dette resultatet ble oppnådd ved å kombinere systemparametere fra modeller som ble indentifisert fra henholdsvis hodet i y-retning og avstand mellom armene i x-retning. Det er interessant at resultatet i dette tilfellet ble forbedret ved å kombinere egenskapsvektorene. Dette tyder på at det er viktig å se på flere kroppsdelers samtidig for å få et optimalt resultat. Dette er egentlig ikke så overaskende, da det er naturlig at karakteristiske bevegelser kan opptre på ulike kroppsdelers på forskjellige tidspunkt.

Det kanskje mest overaskende resultatet, er den oppnådde spesifisiteten og sensitiviteten kun ved å betrakte hodet, henholdsvis på 78.26% og 78.64%. Seksjon 4.7 var et forsøk på å finne en systemteoretisk begrunnelse for dette resultatet. For eksempel kan det hende at den avvikende gruppen har mesteparten av bevegelsene sine rundt andre frekvenser enn de friske. Vi så at selv om det var en forskjell i fordelingen av effekt på de ulike frekvensene, var ikke denne forskjellen veldig stor. Dessuten viser effekt-tetthetsspektrene at AR-modellene gir veldig glatte spektrere, med stor energi på veldig lave frekvenser. Det trengs derfor en nærmere systemteoretisk analyse til, for å forklare årsakene bak dette resultatet.

En annen ting som bør merkes er at det beste resultatet ble oppnådd ved å bruke AR-parametrene direkte som egenskapsvektor i en lineær klassifikator. Lineære klassifikatorer er robuste, men kan fungere dårlig hvis egenskapsvektorene fordeler seg i flere grupper per klasse. Ut i fra et systemteoretisk perspektiv, er det litt overaskende at et så godt resultat fåes ved å betrakte systemparameterne direkte. Vi så blant annet i kapittel 2 at en metrikk basert på systemparameterne ble kritisert på en rekke punkter. Likevel er det lite som tyder på at det er tilfeldigheter i modellidentifiseringen som gir de gode resultatene, da vi så at gode resultater også ble oppnådd ved å endre systemidentifikasjonsmetoder.

Metodene brukt i denne oppgaven er generelle, og de er ikke på noen som helst måte

tilpasset spesielt til spedbarn, annet et at datamaterialet er et resultat av spedbarns-bevegelser. Når vi bruker såpass generelle teknikker som tilpasning av AR-modeller, kan vi bruke teknikker fra en rekke fagområder der tilpasning av dynamiske modeller inngår.

## 5.1 Videre arbeid

I denne seksjonen presenteres det hvordan de allerede diskuterte metodene kan utvikles videre, samt metoder som hittil ikke har vært diskutert.

I denne oppgaven er det blitt brukt systemidentifikasjonsmetoder som betrakter en tidsserie med måledata, og som gir et sett med systemparametere for hver hele tidsserie, som er i form av en ROI eller mini-ROI. En annen innfallsvinkel er å bruke rekursive metoder, som metoden `rarx` i MATLAB. Dette er i prinsippet parameterestimering, der vi ønsker å estimere systemparametere fortløpende. Som diskutert tidligere er det ikke sikkert at spedbarnsbevegelsene kan modelleres godt ved å betrakte hele ROI'er om gangen. Segmentering av ROI'ene i mini-ROI'er var et forsøk på å ta hensyn til nettopp dette. Med en rekursive metode, kan vi betrakte den fortløpende estimeringen av systemparametere som en tidsserie. Ut i fra en slik tidsserie kan det kanskje være mulig å si noe om hvor mye dynamikken til modellen varierer. Siden en rekursiv metode gir et sett med systemparametere i et gitt tidspunkt, kan metrikkene innført i seksjon 3.5.3 brukes for å si hvor mye dynamikken har endret seg fra et tidspunkt til et annet.

Vi så i seksjon 4.6 at segmentering av tidsseriene for SISO AR-modeller førte til at de beste resultatene ble dårligere. En ting som kan prøves ut er å få til en mer intelligent segmentering. Vi så blant annet i kapittel 2 en segmenteringsalgoritme som baserte seg på variasjon i AR-parametrene. Et alternativ, som nevnt ovenfor, hadde vært å brukt metrikkene fra seksjon 3.5.3 til å beregne variasjonen.

Hittil i oppgaven har utgangsvARIABLE vært koordinater i rommet, det vil si tidsserier med  $x$ ,  $y$  og  $z$ . I tillegg til romkoordinatene kan også rotasjonsdata brukes. Det ble testet ut å gjøre om kvaternion-parametrene til Euler-vinkler, og så foreta en klassifisering for hver sensor med hensyn på Euler-vinklene, altså 18 forskjellige tilfeller. Dette førte ikke til noen nevneverdig gode resultater. En nærmere analyse av rotasjonsdataene er nødvendig for å si om de kan skille mellom de to gruppene. En mulighet er å bruke PCA-analyse sammen med romkoordinatene.

I kapittel 2 ble det nevnt at friske spedbarn har såkalte fidgety-bevegelser, bevegelser med stor grad av kompleksitet og variasjon. En måte å betrakte variasjon og kompleksitet på matematisk er å se på dimensjonene på dataene. Ved å bruke PCA er det en mulighet for at prinsipalkomponentene og egenverdiene brukes direkte som et mål på kompleksitet og variasjon i menneskelig bevegelse. Hvis variansen er jevnt fordelt over flere egenvektorer betyr dette at bevegelsen beskrives best av et tilsvarende antall dimensjoner. Stor variasjon i retning av bevegelse vil også kunne tilsvare stor variasjon



i de prinsipale komponentene, hvis de prinsipale komponentene regnes ut på nytt på forskjellige segmenter. Kanskje er det mulig å betrakte tidsserier med prinsipalkomponenter med tilhørende egenverdier. Vi så også i kapittel 2 en algoritme for 2-steps PCA. En slik algoritme bruker ikke dynamiske modeller, og har derfor ikke vært sentral i dette arbeidet. På grunn av metodens enkelhet ble det likevel prøvd å bruke tidsserier fra den normale og avvikende gruppen som observasjonsvektor i en PCA-analyse, og resultatet vises i tillegg B.

Andre metoder som er svært relevante fra bakgrunnskapittelet er bruk av HMM og SLDS. Sistnevnte er hakket mer avansert enn HMM, da man svitsjer mellom forskjellige lineære dynamiske system, og ikke bare diskrete tilstander. I lys av problemstillingen i denne oppgaven er det en veldig interessant algoritme, men den er relativt avansert og det ble ikke funnet tid til å studere den nærmere.

I seksjon 4.3 så vi at det var stor forskjell i variansen for sensorene 1 og 5 mellom den friske og den avvikende gruppen. Mens sysemparametere i dynamiske modeller sier noe om dynamikken til bevegelsene, gir variansen et mål på mengden av bevegelse. Egenskapsvektorene med systemparametere kan utvides med variansen til signalet, for å få med et til aspekt av bevegelsene til spedbarna.

Utgangsvariable for de dynamiske modellene i denne oppgaven, ble valgt etter kriterier listet opp i seksjon 3.2. Et annet valg av utgangsvariable, begrunnet ut i fra kjennskap om fidgety-bevegelser, kan muligens gi bedre resultater.

Romkoordinatene i denne oppgaven har vært betraktet i kartesiske koordinater. Polarkoordinater i planet utspent av 2 prinsipale komponenter fra PCA ble også testet ut, uten å gi bedre resultater enn de kartesiske. Kulekoordinater kunne også vært aktuelt å teste ut.

I denne oppgaven har det blitt brukt to klassifikatorer, LDA og KNN. Når det gjelder KNN ble  $k$ -parameteren, som sier hvor mange naboer man skal ta hensyn til, satt til 10. Dette var et valg som ble tatt ut i fra tester med forskjellige utgangsvariable. For et optimalt resultat i hvert enkel tilfelle, kunne  $k$  blitt optimalt tilpasset for hver enkelt klassifisering.

Et nevralt nettverk ble testet ut for klassifisering. Det viste seg å gi tilsvarende resultater som LDA ved bruk av lineære transferfunksjoner i nettverket. Dette er fordi nettverket reduseres til en lineær klassifikator ved bruk av slike transferfunksjoner. Med andre transferfunksjoner i nettverket ble det svært varierende resultater fra gang til gang. Dette ble ikke undersøkt nærmere, og det er godt mulig gode resultater med slike nettverk er mulig.

## 5.2 Evaluering opp mot GMA

Det beste resultatet i dette arbeidet har en spesifisitet på 90.91% og en sensitivitet på 85.71%. Som vi så i kapittel 2 er det ved bruk av GMA blitt oppnådd en spesifisitet på 96% og en sensitivitet på 95%. Det er ikke usannsynlig, at ved å blant annet studere andre valg av utgangsvariable og kombinasjon av egenskapsvektorer fra modeller identifisert fra disse variablene, at en metode basert på dynamisk systemteori kan gjøre det nesten eller like godt som GMA.

Selv om dette ikke skulle være tilfellet, kan dynamisk systemteori være en verdifull innfallsvinkel for å se på hvilke sensorer som er viktige, og for å studere dynamikken til bevegelser som skiller mellom de to gruppene. Vi så for eksempel at hodet(pannen) gav gode resultater alene. Ved en nærmere systemteoretisk analyse av modellene identifisert fra den friske og den avvikende gruppen, er det mulighet for å koble systemteoretiske egenskaper med det en kliniker observerer. Kanskje er det også slik at det kan oppdages nye egenskaper ved avvikende bevegelser ved bruk av dynamisk systemteori, som er for komplekse eller subtile at det ikke oppdages ved visuell observasjon.

## Kapittel 6

# Konklusjon

Tilpasning av dynamiske modeller er en nyttig metode for klassifisering av spedbarnsbevegelser. Modeller identifisert fra bevegelsesdata tilhørende hodet og armer skiller mer mellom de to gruppene enn benene og bryst. Ved å kombinere systemparameterne fra modeller basert på hodebevegelser og armbevegelser, oppnåes en spesifisitet og sensitivitet på henholdsvis 90.91% og 85.71%. For å oppnå dette resultatet ble det brukt SISO AR-modeller sammen med LDA som klassifikator. Dette er ikke like bra som den kliniske metoden GMA, men det er sannsynligvis mange muligheter for å forbedre resultatet.

Resultatene bekrefter at de dynamiske modellene til de to gruppene er forskjellige, og det er interessant å studere de fra et systemteoretisk perspektiv med blant annet frekvens-tetthetsspektre og poler. Ved å finne forklaringer på forskjeller i dynamikk ut i fra slike termer, kan de muligens relateres til GMA-metoden og det en kliniker observerer. Således er den en mulighet for at bruk av modelltilpasning for klassifisering av spedbarnsbevegelser, i tillegg til å være en selvstendig automatisk metode, kan brukes til å få større innsikt i skilnaden mellom normale og avvikende bevegelser.



# Bibliografi

- Barbic, J., Safonova, A., Pan, J., Faloutsos, C., Hodgins, J. & Pollard, N. (2004), 'Segmenting motion capture data into distinct behaviors'.
- Bissacco, A., Chiuso, A., Ma, Y. & Soatto, S. (2001), 'Recognition of human gaits'.
- Biswas, S. & Quwaider, M. (2008), 'Body posture identification using hidden markov model with wearable sensor networks'.
- Brockwell, P. J. & A.Davis, R. (2001), *Introduction to Time Series and Forecasting, second edition*.
- Chen, M., Huang, B. & Xu, Y. (2007), 'Human abnormal gait modeling via hidden markov model'.
- Cock, K. D. & de Moor, B. (2000), 'Subspace angles and distances between arma models'.
- de Oliveira Andrade, A. & Soares, A. B. (2007), 'Emg pattern recognition for prosthesis control'.
- F.Kelly, M., A.Parker, P. & N.Scott, R. (1990), 'The application of neural networks to myoelectric signal analysis: A preliminary study'.
- Fod, A., J.Mataric, M. & Jenkins, O. C. (2002), 'Automated derivation of primitives for movement classification'.
- Gibet, S., Kamp, J.-F. & Poirier, F. (2004), 'Gesture analysis: Invariant laws in movement'.
- Hadders-Algra, M. (2004), 'General movements: A window for early identification of children at high risk for developmental disorders'.
- Henriksen, R. (1998), *Stokastiske systemer, Analyse, estimering og regulering*.
- J.G.Barton & A.Lees (1995), 'An application of neural networks for distinguishing gait patterns on the basis of hip-knee joint angle diagrams'.
- Jin, N. & Mokhtarian, F. (2005), 'Human motion recognition based on statistical shape analysis'.

- L.Meinecke, Breitbach-Faller, N., C.Bartz, R., G.Rau & C.Disselhorst-Klug (2006), 'Movement analysis in the early detection of newborns at risk for developing spasticity due to infantile cerebral palsy'.
- Lu, C. & Ferrier, N. J. (2003), 'Automated analysis of repetitive joint motion'.
- Martin, R. (1997), 'Autoregression and cepstrum-domain filtering'.
- Martin, R. J. (2000), 'A metric for arma processes'.
- Meyer, D. (1997), 'Human gait classification based on hidden markov models'.
- Moeslund, T. B., Hilton, A. & Krüger, V. (2006), 'A survey of advances in vision-based human motion capture and analysis'.
- Nam, Y. & Wahn, K. (1997), 'Recognition of hand gestures with 3d, nonlinear arm movement'.
- Overschee, P. V. & Moor, B. D. (1994), 'N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems'.
- Overschee, P. V. & Moor, B. D. (1996), 'Subspace algorithms for the stochastic identification problem'.
- Pavlovic, V. & Rehg, J. M. (2000), 'Impact of dynamic model learning on classification of human motion'.
- Pfau, T., Ferrari, M., Parsons, K. & Wilson, A. (2008), 'A hidden markov model-based stride segmentation technique applied to equine inertial sensor trunk movement data'.
- Picci, G. (1996), 'Geometric methods in stochastic realization and system identification'.
- Prechtl, H., Einspieler, C., Cioni, G., Bos, A., Ferrari, F. & Sontheimer, D. (1997), 'An early marker for neurological deficits after perinatal brain lesions'.
- Proakis, J. G. & Manolakis, D. G. (1996), *Digital Signal Processing, Principles, algorithms, and applications, third edition.*
- P.Viviania & C.Terzuolo (1982), 'Trajectory determines movement dynamics'.
- R.Das, S., C.Wilson, R., T.Lazarewicz, M. & H.Finkel, L. (2006), 'Gait recognition by two-stage principal component analysis'.
- Robertson, S. S. (1982), 'Intrinsic temporal patterning in the spontaneous movement of awake neonates'.
- Sanger, T. D. (2000), 'Human arm movements described by a low-dimensional superposition of principal components'.
- Theodoridis, S. (2006), *Pattern Recognition, third edition.*

Veeraraghaven, A., Chowdhury, A. R. & Chellappa, R. (2004), 'Role of shape and kinematics in human movement analysis'.

Wu, J., Wang, J. & Liu, L. (2007), 'Feature extraction via kpca for classification of gait patterns'.





## Tillegg A

# Seperabilitet

(a) Sensorpar, akse  $x$

variabel	J	d
$y_{13_x}$	16.4246	1.9247
$y_{14_x}$	16.3429	1.6477
$y_{15_x}$	16.2936	1.4652
$y_{34_x}$	16.4082	1.8720
$y_{35_x}$	16.3892	1.8101
$y_{45_x}$	16.3813	1.7815

(b) Sensorpar, akse  $y$

variabel	J	d
$y_{13_y}$	16.3450	1.6573
$y_{14_y}$	16.3149	1.5482
$y_{15_y}$	16.1744	0.9592
$y_{34_y}$	16.3026	1.5010
$y_{35_y}$	16.3472	1.6652
$y_{45_y}$	16.2885	1.4475

(c) Sensorpar, akse  $z$

variabel	J	d
$y_{13_z}$	16.2230	1.1793
$y_{14_z}$	16.2741	1.3899
$y_{15_z}$	16.1530	0.8588
$y_{34_z}$	16.3374	1.6284
$y_{35_z}$	16.1828	0.9990
$y_{45_z}$	16.2854	1.4336

Tabell A.1: Separabilitet, 2 utganger: Samme akse på 2 forskjellige sensorer

variabel	J	d
$\mathbf{y}_{13_{PCA}}$	16.4813	2.0981
$\mathbf{y}_{14_{PCA}}$	16.3249	1.5836
$\mathbf{y}_{15_{PCA}}$	16.2896	1.4501
$\mathbf{y}_{34_{PCA}}$	16.4591	2.0317
$\mathbf{y}_{35_{PCA}}$	16.4140	1.8917
$\mathbf{y}_{45_{PCA}}$	16.3246	1.5820

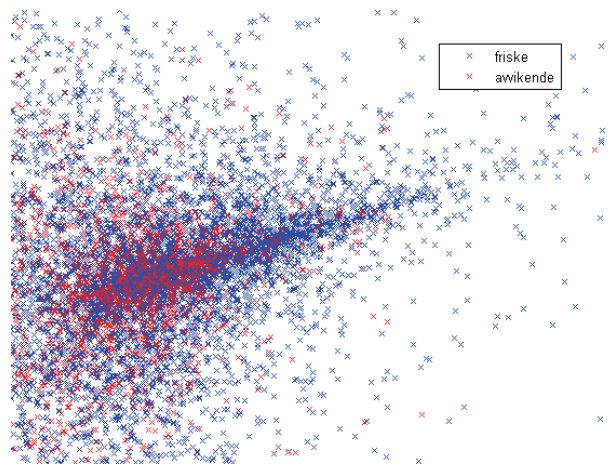
Tabell A.2: Sensorpar, projisert ned på den prinsipale komponenten

variabel	J	d
$\mathbf{y}_{1_{PCA}}$	16.4822	2.1018
$\mathbf{y}_{2_{PCA}}$	16.3093	1.5298
$\mathbf{y}_{3_{PCA}}$	16.4469	1.9957
$\mathbf{y}_{4_{PCA}}$	16.4285	1.9370
$\mathbf{y}_{5_{PCA}}$	16.2025	1.0886
$\mathbf{y}_{6_{PCA}}$	16.5000	2.1555

Tabell A.3: Sensor 1-6, projisert ned på 2 prinsipale komponenter

## Tillegg B

# PCA med tidsserie som observasjonsvektor



Figur B.1: PCA analyse på sensor 1, akse 3. Segmenter på 4 sekunder er blitt brukt som observasjonsvektor, og data er blitt projisert ned på 3 prinsipale komponenter. Figuren viser et utsnitt av de projiserte observasjonsvektorene



## Tillegg C

# Implementering i MATLAB

Denne seksjonen vil kort ta for seg hvordan bevegelsesdata er representert i MATLAB. For informasjon om implementering av metoder, henvises det til filen `kjormeg.m`.

ROI'er fra friske og syke spedbarn er samlet i filene `normal.mat` og `abnormal.mat`. Disse filene ble lagd ved å hente ut data fra et annet datasett, gitt av veileder, som ikke var inndelt i ROI'er.

`normal.mat` og `abnormal.mat` er strukter, og ved å skrive navnet på førstnevnte i MATLAB dukker følgende opp:

```
normal =  
1x202 struct array with fields:  
tab  
id  
cnt  
ROI_number
```

Vi ser strukten med bevegelsesdata fra normale barn inneholder 202 ROI'er. Vi får opp en tabell med 202 strukter, der hver strukt inneholder selve dataene (x,y,z, [kvaternion]), id'en på barnet som ROI'en kommer fra, hvilket opptak ROI'en er hentet fra, samt et ROI\_number som teller oppover for hver ROI innenfor en gitt id innenfor et gitt opptak. For å hente ut posisjonsdata til den første ROI'en for sensor 1 skrives:  
`normal(1).tab(:,1:3,1)`

Filten `lag_testsett.m` deler inn datasettet inn i 2 testsett (normal og avvikende) og 2 treningsett (normal og avvikende) der formatet på elementene er som ovenfor. Ved f. eks skrive `trening_sett_norm` i MATLAB fås:

```
trening_sett_norm =
```

```
1x99 struct array with fields:
```

```
tab
```

```
id
```

```
cnt
```

```
ROI_number
```