

BioPatRec som forskningsplattform med fokus på inngangssignaler

Kristian Berrum

Master i teknisk kybernetikk

Innlevert: juni 2013

Hovedveileder: Øyvind Stavdahl, ITK

Medveileder: Anders Fougner, ITK

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

Masteroppgave

Studentens navn: Kristian Berrum

Fag: Teknisk kybernetikk

Tittel (norsk): BioPatRec som forskningsplattform med fokus på inngangssignaler

Title (english): BioPatRec as a research platform focusing on input signals

Beskrivelse:

BioPatRec er et fremvoksende åpent kildekode-prosjekt, og tilbyr en forskningsplattform for testing og utvikling av algoritmer til protesestyring. Vi vil undersøke om denne plattformen er egnet for våre forskningsområder. Oppgaven omfatter følgende punkter:

1. Utvikle en funksjonsspesifikasjon for et program til en proteseforskningsplattform. Spesiell vekt skal legges på inngangssignaler.
2. Relater BioPatRec-systemet til de etablerte kravene, og gi en vurdering om hvorvidt det bør innføres som plattform i vår lab.
3. Basert på resultatene i punkt 2, adapter BioPatRec-systemet for bruk i vår lokale lab.

Faglærer: Øyvind Stavadahl, Institutt for teknisk kybernetikk

Veileder: Anders Fougner, Institutt for teknisk kybernetikk

Trondheim, 07.01.2013

Sammendrag

Denne masteroppgaven er gjort under instituttet for teknisk kybernetikk (ITK) ved Norges teknisk-naturvitenskapelige universitet (NTNU). Oppgaven vurderer BioPatRec som er et internasjonalt proteseprojekt med åpen kildekode. Det vil her bli lagt spesielt vekt på signalopptak aspektene ved programvaren.

Det må settes krav for å vurdere BioPatRec som et alternativ til ITK-programvaren som så langt har blitt brukt til protesestyring ved det nevrologiske laboratoriet hos NTNU. Et fullstendig sett med funksjonsspesifikasjoner ble satt opp og vurdert for å bedømme om BioPatRec tilfredsstiller kravene det nevrologiske laboratoriet har til protesestyring.

Med bakgrunn i disse spesifikasjonene ble BioPatRec vurdert. Det ble funnet noen mindre problematiske aspekter ved programvaren. BioPatRec kom gjennom som et godt alternativ etter programvarevurderingen og etter selve implementasjonen. De små problemene som ble funnet veier ikke opp for de store fordelene BioPatRec bringer som et alternativ til ITK-programvaren.

Det ble implementert løsninger til to av hovedmodulene i BioPatRec. Begge løsningene går i hovedsak ut på å implementere det tilgjengelige sensorsystemet på laboratoriet i BioPatRec. Det er derfor en løsning for hver av disse to hovedmodulene i BioPatRec. Dette resulterte i at BioPatRec nå kan ta fullstendige signalopptak med sensorutstyret brukt i det nevrologiske laboratoriet.

Det er i denne oppgaven også lagt arbeid i å utvikle en norsk terminologi. Det er mange tekniske begreper innen moderne proteseteknikk og det er derfor viktig at man har en terminologi for dem.

Abstract

This thesis is completed under the department of engineering cybernetics (ITK) at the Norwegian university of science and technology (NTNU). The thesis takes a closer look at BioPatRec which is an international open source project for prosthetic control and development. The emphasis in this thesis will be the signal output aspects of the software.

Some requirements is needed in order to consider BioPatRec as an alternative to the ITK-software which is currently being used for prosthetic control at the prosthetic laboratory at NTNU. A complete set of functional specifications was created and considered in order to determine if BioPatRec can satisfy the demands of the current prosthetic laboratory

BioPatRec was evaluated and turned out to be a good alternative to the current software. This was the conclusion after the first software consideration and after the software implementations. The benefits of using BioPatRec was more than enough to compensate for minor problems that was encountered during the implementations.

It was implemented solutions for two of the main moduls in BioPatRec. Both solutions is regarding implementations of the available sensor system at the laboratory in BioPatRec. This means that there is a solution for both these two main moduls. This resulted in functional signal recording in BioPatRec using the equipment at the prosthetic laboratory.

This thesis also includes a development of a Norwegian terminology. There are many technical terms in prosthetics and it is therefor important to have a proper terminology.

Forord

Denne oppgaven er et resultat av min mastergrad ved Teknisk Kybernetikk på NTNU. Masteroppgaven har vært givende og til tider meget krevende. Oppgaven gav gode resultater og jeg er meget fornøyd med utfallet.

Jeg vil derfor benytte anledningen til å takke mine veiledere Anders Fougner og Øyvind Stavadahl for å ha kommet med gode råd og forslag underveis. Jeg vil også takke min medstudent Alf Bertnum for samarbeidet vi har hatt gjennom våre oppgaver. Vi har hatt mange verdifulle diskusjoner og samtaler under denne oppgaven.

Jeg ønsker også å rette en takk til familie og venner som har støttet meg under denne oppgaven. Jeg vil spesielt takke Andreas Tronsen som har gitt meg god veiledning til den skriftlige rapporten.

Til slutt vil jeg takke Torleif Anstensrud for uerstattelig støtte gjennom de fem årene jeg har tilbrakt på NTNU.

Innhold

Forord	v
Figurer	x
Tabeller	xi
Nomenklatur	xii
I Introduksjon	1
1 Motivasjon	1
2 Disposisjon og bidrag	2
2.1 Disposisjon av oppgaven	2
2.2 Bidrag	3
3 Bakgrunn	4
3.1 Proteseteknologi	4
3.2 Signalopptak og signalegenskaper	4
3.2.1 Myoelektriskesignaler og elektromyografi	5
3.2.2 Akselerasjon og akselerometer	5
3.2.3 Kraft	6
3.2.4 Multimodal sensorenhet	7
3.3 Protesestyring	7
3.3.1 Mønsterkjennning	7
3.3.2 Proporsjonalstyring	8
3.4 Skjermguidet- og proteseguidet-trening	9
3.5 BioPatRec	9
3.5.1 Signal recordings	10
3.5.2 Signal treatment	10
3.5.3 Feature extraction	11
3.5.4 Pattern recognition	12
3.5.5 Control algorithms	12
4 Mål	14
II Metode	17
5 Utstyr og funksjonspesifikasjon	17
5.1 Protesen	17

5.2	Sensorene	19
5.3	Programvare	21
5.4	Funksjonsspesifikasjon	21
6	Programvarevurdering	26
6.1	Preprosessering	26
6.2	Intensjonsestimering	28
6.3	Output	29
6.4	System training	29
6.5	Konklusjon av programvarevurderingen	30
7	Implementasjon	31
7.1	Recordings	31
7.1.1	Første implementasjon: SEMG signaler	32
7.1.2	Andre implementasjon: Opprydning av initialisering	33
7.1.3	Tredje implementasjon: Frekvensspekteret	35
7.1.4	Fjerde implementasjon: Tidsstempler	35
7.2	Recording Session	37
7.2.1	Første implementasjon: Overfør funksjonaliteten	39
7.2.2	Andre Implementasjon: Inkludering av akselerometer	40
III	Resultater	43
8	Bidrag og utførelse	43
8.1	Norsk terminologi	43
8.2	Funksjonsspesifikasjonen	43
8.3	Bidrag i BioPatRec	43
8.4	Trigno i bruk	44
IV	Diskusjon	51
9	Diskusjon	51
9.1	Implementasjonshåndtering	51
9.1.1	Tidsstempler	52
9.1.2	Ressursbruk	52
9.1.3	Ikke kompatible datasett	53
9.2	BioPatRec	54
9.2.1	BioPatRec i forhold til ITK-programvaren	55
10	Konklusjon	56
11	Fremtidig Arbeid	57

11.1 Videre implementasjoner	57
11.1.1 Delsys Trigno Wireless System	57
11.1.2 Recording Session	58
11.1.3 Kraft sensorer	58
11.2 Norsk terminologi	58
11.3 Laboratoriet	58
Referanser	61
Appendiks A CD	63

Figurer

1	SEMG forsterker	6
2	Klassifisering av fisk	8
3	BioPatRec oversikt	10
4	Hovedmenyen i BioPatRec	11
5	Motion Control protesen	18
6	Motion Control protesen på laboratoriet	18
7	NI DAQ-enhet	19
8	Delsys Trigno Wireless System	20
9	Protesestyringsmodell	25
10	BioPatRec Grensesnitt for opptak	32
11	Delsys Trigno Control Utility	34
12	Hovedmeny for Recording Session	38
13	Meny for valg av sensor	39
14	Trigno Session: Bare SEMG	45
15	Trigno Session: All data inkludert	46
16	Recording Session under signalopptak	46
17	Recording Session etter signalopptak	47
18	Trigno Session med akselerometerdata	47

Tabeller

1	Tabell med signalegenskaper fra BioPatRec	13
2	Tabell med funksjonsspesifikasjon	23
3	Tabell med foreslått norsk terminologi	49
4	Bidrag i BioPatRec	50

Nomenklatur

DAQ	Data acquisition eller datainnsamling
DTWS	Delesys Trigno Wireless System, er sensorsystemet utviklet av Delsys som benyttes ved det nevrologiske laboratoriet ved NTNU.
EMG	Elektromyografi, opptak av de ekstracellulære feltpotensialet produsert av musklene.
ITK	Instituttet for Teknisk Kybernetikk
MC	Motion Control, protesen brukt ved det nevrologiske laboratoriet
MES	Myoelektriske signaler, det ekstracellulære feltpotensialet produsert av muskelen
NI	National Instruments
NTNU	Norges Tekniske Naturvitenskapelige Universitet
PGT	Proteseguidet-trening, brukeren følger øvelsen ved og observere en protese
SDK	Software development kit, er verktøy som gjør at brukere kan utvikle egen programvare til produktet.
SEMG	Overflate elektromyografi
SGT	Skjermguidet-trening, brukeren følger øvelsene på en skjerm
USB	Universal Serial Bus, dette er en seriell databuss-standard for å koble enheter til en datamaskin.
VRE	Virtual Reality Environment er navnet BioPatRec har gitt en simulert protesehånd som kan brukes i programvaren for å simulere protesen

Del I

Introduksjon

1 Motivasjon

I flere år har Instituttet for teknisk kybernetikk (ITK) utviklet en egen programvare for bruk ved det nevromotoriske laboratoriet. Det er skrevet i Labview og mange studenter har deltatt i utviklingen av dette programmet. Studenter har skrevet prosjektoppgaver og masteroppgaver i dette programmet. Dette medfører at få har holdt på samtidig, og en studentens arbeid blir en påbygning på en annen students oppgave. Dette har i lengden gjort at kodens oppbygning har blitt litt rotete og det blir vanskeligere for nye studenter å sette seg inn i programvaren å fortsette utviklingen. Dette er blant annet påpekt i Berrum (2012). Det er derfor ønskelig og finne et alternativ til ITK-programvaren.

BioPatRec er et internasjonalt prosjekt for protesestyring. I tillegg har dette prosjektet åpen kildekode. Dette betyr at hvem som helst kan bidra til prosjektet og utvikle programvaren videre for eget bruk. BioPatRec er beskrevet i Ortiz-Catalan et al. (2013). Et viktig fokus i BioPatRec er modulariteten, siden BioPatRec skal være en forskningsplattform som så mange som mulig innen dette feltet kan benytte. Derfor ser programvaren ut som et svært lukrativt alternativ til ITK-programvaren.

Får å kunne benytte BioPatRec er det viktig at det er mulig å bruke sensor-systemet som er tilgjengelig. Derfor vil denne oppgave omfatte signalopptak og ha et spesielt fokus på dette under utførelsen.

2 Disposisjon og bidrag

Denne oppgaven er delt opp i fire hoveddeler. Under er en kort gjennomgang av hva hver av disse delene inneholder.

2.1 Disposisjon av oppgaven

Introduksjon er delen som trengs for å sette seg inn i oppgaven. Her finner man motivasjonen bak oppgaven, bakgrunnsstoffet som er nødvendig for å forstå og utføre oppgaven, samt utredelse av målet til oppgaven.

Metode inneholder oppgavens utførelse. Dette er gjort slik at hvert kapittel her representerer et av punktene i oppgaveteksten. Slik blir det ryddig å finne frem til en spesifikk del av oppgaveutførelsen. Det er da altså et kapittel for funksjonsspesifikasjon, et kapittel som vurderer programvaren og et kapittel for implementasjonen.

Resultater her blir det gjennomgått hva som faktisk ble gjort under oppgaven. Resultatene i denne oppgaven vil på en måte være bidragene som oppgaven har resultert i. Dette inkluderer norsk terminologi, funksjonsspesifikasjoner og de forskjellige implementasjonene gjort underveis.

Diskusjon og konklusjon er delen som drøfter arbeidet gjort i oppgaven. Her vil det bli lagt vekt på problematikken som oppstod underveis, samt BioPatRec sine fordeler og ulemper. Det vil så tas en konklusjon om BioPatRec passer seg for videre bruk basert på resultater fra denne oppgaven.

2.2 Bidrag

Norsk Terminologi På bakgrunn av at denne oppgaven er skrevet på norsk er det blitt dannet en norsk terminologi i tabell 3 for ord og uttrykk som er bruk i forbindelse med denne oppgaven.

Funksjonsspesifikasjon I tabell 2 er funksjonsspesifikasjonene listet. Dette er krav som er satt opp for programvare som skal brukes til protesestyring ved det veromotoriske laboratoriet ved NTNU.

Enkeltopptak Enkeltopptak i BioPatRec er en hovedmodul av programvaren som lar deg enkelt gjøre signalopptak for å sjekke at kvaliteten på signalene er bra og at sensorene dine fungerer. Denne oppgaven har implementert en enkeltopptaks løsning med sensorene benyttet av ITK.

Fullstendig opptak BioPatRec har en hovedmodul som lar brukeren gjøre signalopptak av forskjellige øvelser som vil brukes av programvaren videre. Her er det implementert en løsning som benytter sensorene tilgjengelige hos ITK.

Forslag til akselerometer standard Ettersom BioPatRec bare støtter elektromyografi må det bli foreslått en ny standard for hvordan akselerometerdata lagres. Dette er nødvendig ettersom et akselerometer kan samle fra tre akser samtidig.

Fremtidig arbeid Denne oppgaven har også lagt frem hva som må gjøres videre for en fullstendig implementasjon av BioPatRec ved det nevmotoriske laboratoriet til NTNU.

Dokumentasjon På BioPatRec sin egen hjemmeside er det lastet opp en dokumentasjon av implementasjonene gjort her. Denne kan oppsøkes i Berrum and Bertnum (n.d.)

3 Bakgrunn

I dette prosjektet skal proteseprogramvaren BioPatRec studeres, å senere modifiseres for bruk på det nevromotoriske laboratoriet ved NTNU. For at dette skal være mulig er det nødvendig å ha en grunnleggende forståelse av hva BioPatRec gjør og hvordan den gjør det. Det er også nødvendig å forstå fagord og metoder som brukes innen proteseforskning. Dette kapitlet vil derfor fokusere på å gi den informasjonen som er nødvendig for å sette seg inn i prosjektet.

3.1 Proteseteknologi

En protese er en innretning som har som mål å erstatte en tapt lemsdel. En protese kan benyttes av rent kosmetiske grunner, eller den kan benyttes som et rent funksjonelt organ som er konstruert for å håndtere visse oppgaver. En kombinasjon er selvfølgelig også mulig der brukeren ønsker å få tilbake noe av funksjonaliteten til det tapte organet i tillegg til at protesen også likner det tapte organet. En kosmetisk protese kan være et kunstig øye, der en mer funksjonell protese kan være en armprotese som kan utføre visse oppgaver, som å gripe objekter. Det bør nevnes at armproteser kan ha kosmetiske egenskaper ettersom man kan trekke forskjellige materialer over protesen som ligner hud.

Det kan skilles mellom mekaniske proteser og elektriske proteser. Hvis man tar armproteser som eksempel kan forskjellen illustreres ganske enkelt. En slik protese kan styres av en kabel som går fra hånden til skulderen og derfor lar brukeren styre grepet i hånden via skulderen. Proteser som dette har eksistert lenge og fordeler med dem er at de er raske og meget nøyaktige. Dette er blant annet et resultat av simplisiteten til selve produktet.

Elektriske proteser er derimot mer avanserte. En armprotese kan ta opp signaler generert av muskelen under en kontraksjon i den gjenværende armstumpen. Disse signalene kan behandles og benyttes til å styre protesen. Dette gir potensialet til å styre flere funksjoner, som grepet i hånden og rotasjon rundt håndledet. Det trengs derfor flere aktuatorer for en slik protese. Slike proteser har ikke vært på markedet lenge og er i sterk utvikling i tråd med datakraft- og sensorutvikling.

3.2 Signalopptak og signalegenskaper

For videre utvikling innen protesestyring er det veldig viktig med høyt fokus på signalopptak og signalbehandling. Dagens proteseteknologi har mange utfordringer foran seg, og mange av disse utfordringene kan løses med bedre

sensorutstyr og bedre håndtering av signalopptakene. Det er derfor viktig at en grunnleggende gjennomgang av signalopptak og signalegenskaper blir presenter. En mer omfattende innføring av forskjellige sensortyper er beskrevet i Berrum (2012), der en del av oppgaven var og legge ned et omfattende studie i hvilke typer sensorer som kan brukes til avansert protesestyring. En gjennomgang av de mest aktuelle sensorene for denne oppgaven er beskrevet her.

3.2.1 Myoelektriskesignaler og elektromyografi

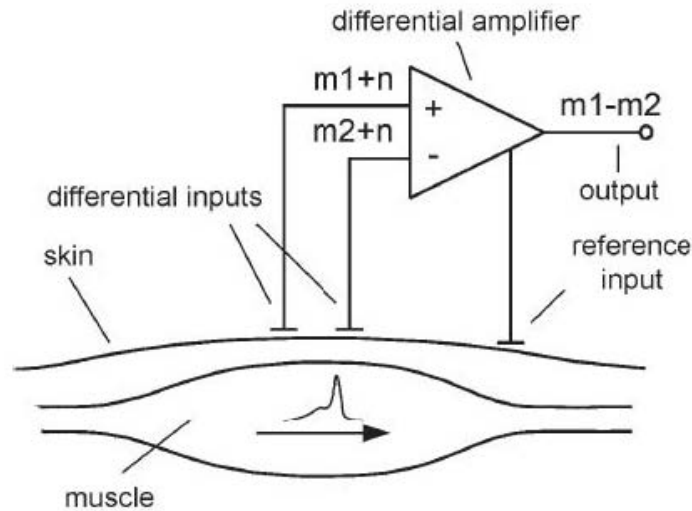
Myoelektriske signaler (MES) er et elektrisk signal som vandrer langs muskelen under en kontraksjon. Dette gjør at muskelens aktivitet er målbar og kan monitoreres relativt enkelt ved hjelp av forskjellig type sensorer. Dette fører til at det i dag er vanskelig å drive med proteseutvikling uten og benytte dette signalet.

Elektromyografi (EMG) er opptaket av MES. Denne målingen kan gjøres med flere typer sensorer men det vanligste er overflate (surface) EMG (SEMG). Dette betyr at elektrodene er på overflaten av huden og trenger ingen direkte kontakt med muskelen for å fungere. Elektrodene kan festes direkte på huden eller man kan benytte en ledende gel som forbedrer kontakten til sensoren. Oppsettet på laboratoriet bruker SEMG sensorer som ikke trenger gel. Det er mange utfordringer med SEMG. De kan utsettes for store forstyrrelser som kan være et resultat av at sensoren flytter seg langs huden eller at brukeren svetter. Slike forstyrrelser er et stort problem siden dette direkte kan påvirke oppførselen til selve protesen. Derfor er det viktig og finne metoder og løsninger som retter opp slike feil hos SEMG.

SEMG sensorer har ofte tre elektroder. To av elektrodene kan måle MES på forskjellige punkter av en muskel der den tredje elektroden er referansen. Strukturen av en slik sensor kan observeres i figur 1. Det er nødvendig og forsterke signalet ettersom MES i utgangspunktet er et svakt signal. forsterkeren er jordnet til referanse for å unngå store forstyrrelse fra omgivelsene.

3.2.2 Akselerasjon og akselerometer

Akselerasjons målinger kan gi mye informasjon til et protesesystem. Det er viktig å skjønne at muskelgrupper oppfører seg forskjellig med hensyn på hvilke posisjon kroppen er i og hvor raskt posisjoner endres. Et eksempel vil være å holde en arm rett opp. Du kan rolig senke armen ned i normal posisjon eller du kan dra den ned raskt. Bevegelsen er den samme men forskjellige muskler blir benyttet for øvelsene. Om armen holdes i to forskjellige posisjoner når man åpner hånden kan dette ha forskjellig påvirkning på EMG opptaket.



Figur 1: SEMG forsterker fra Sæther (2008)

Derfor passer det bra å inkludere et akselerometer. Et akselerometer med tre akser kan beskrive orienteringer i et tre-akset koordinatsystem og den kan gi beskrivelser av bevegelser. Denne dataen kan derfor utnyttes til å supplere EMG-dataen. Slik kan en riktig trent protese ta hensyn til bevegelser og posisjonering. Dette kan gjøres på mange måter. Det kan settes opp spesialtilfeller der et spesifikk mønster er knyttet til spesifikke responser hos protesen, akselerometerdataen kan også enkelt legges til EMG-dataen når signal egenskaper blir valgt. Dette i seg selv vil hjelpe ettersom dataen fra forskjellige posisjoneringer er bakt inn i signal egenskapene.

3.2.3 Kraft

På det nevromotoriske laboratoriet til NTNU er det eksperimenter med kraftmålinger, og hvilke konsekvenser dette kan ha i et protesesystem. Denne oppgaven kommer til å fokusere på MES og akselerasjon, men siden dataen den returnerer kan ha signifikans for resultatet er de absolutt verdt å ta opp. Kraftmålingene i denne sammenhengen er den målte kraften som virker på sensoren fra overflaten på huden. Det vil si at om en SEMG sensor skulle oppleve *lift off* som er at sensoren mister kontakt med huden vil kraft sensoren kunne måle dette og det bør være mulig og fjerne SEMG målingene siden de ikke lenger har muligheten til å ta opp MES. Kraft signaler vil også generelt kunne inkludere i resten av måledataen, slik at systemet også bedømmer kraft på et generelt plan når den detekterer mønster.

3.2.4 Multimodal sensorenhet

I avanserte protesesystemer brukes ofte mer enn bare en type sensor. Det er derfor greit å ha et par definisjoner.

*Et **multimodalt system** er et system som består av to eller flere forskjellige typer sensorer.*

*En **multimodal sensorenhet(MMU)** er en enkelt enhet som i seg selv er et multimodalt system.*

Definisjonene over er hentet fra Suleng (2012) der fokuset er utvikling av en MMU, som skulle fungere med ITK-programvaren satt opp på dette proteselaboratoriet. Sensorer som inkluderer flere type måleinstrumenter er altså en MMU. Denne oppgaven benytter en slik MMU senere ettersom dette kan gi mange fordeler. Som nevnt tidligere er SEMG en viktig måling og den har hovedansvaret for opptaket av muskelaktiviteten. SEMG kvaliteten kan derimot forfalle på grunn av ytre påvirkninger og andre forstyrrelser. Derfor er det lurt å inkludere flere typer målinger i samme sensorer, på denne måten kan målinger som gir muligheten for å detektere og fjerne feil i SEMG målingene. Disse målingene kan selvfølgelig også benyttes selvstendig som enkeltsignaler som i seg selv kan gi nyttig informasjon til protesestyringen.

En MMU konstruert for proteselaboratoriet ved NTNU kan studeres i detalj i Suleng (2012). Denne sensoren ble videre implementert i Berrum (2012), der fordelene ved å inkludere en slik sensor er diskutert i oppgaven.

3.3 Protesestyring

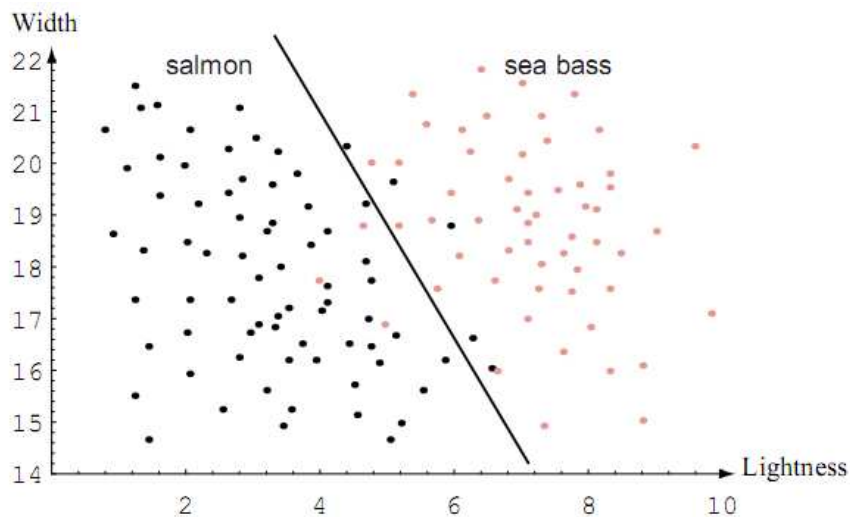
Selve protesestyringen er ikke den mest relevante delen av denne oppgaven. Men siden flere emner under protesestyring vil bli nevnt i denne oppgave er det om ikke annet nødvendig med et raskt overblikk med hensyn til noen av de viktigste begrepene.

3.3.1 Mønstergjenkjenning

Mønstergjenkjenning handler om å klassifisere data som et mønster. Når det er snakk om proteser vil dette si å klassifisere data fra for eksempel en SEMG sensor, ved å se etter spesielle mønster som her kan bety en ønsket bevegelse. Det vil være mulig å skille en rotasjon av hånden fra en knytting av neven ved å klassifisere SEMG-dataen som er et resultat av disse bevegelsene.

Mønstergjenkjenning kan brukes i mange flere sammenhenger enn prote-sestyring. Man kan bruke mønstergjenkjenning til å skille fiske arter fra hver-andre. Ved å la et kamera observere fisk på et samleband kan den skille fisk fra hverandre ved og se på spesifikke egenskaper i dataen. Det er mulig å set-te opp egenskapsvektorer som holder verdier for de forskjellige egenskapene. Derfor øker vektoren i dimensjoner med antall egenskaper. Ved bruk av man-ge egenskaper ender man opp med mange dimensjoner og det er derfor lurt å kun inkludere egenskaper som best klassifiserer objektene. Et to-dimensjonalt eksempel for klassifisering av fisk er vist i figur 2. En full innføring av dette eksempelet finnes i Duda et al. (2001).

Den mest brukte mønstergjenkjenningen i en protese sammenheng er av/på-klassifisering, der en klasse enten er av eller på. Av/på-klassifisering er til når det som er brukt i BioPatRec.



Figur 2: Klassifisering av havabbor mot laks fra Duda et al. (2001)

3.3.2 Proporsjonalstyring

Proporsjonalstyring skiller seg fra av/på-klassifisering ved forskjellen i aktive-ringsprofilen. Aktiveringsprofilen til av/på-klassifisering dikterer alt eller ingen ting. I en protese sammenheng betyr dette at en aktuator enten er på eller av. Dette betyr at den alltid vil få samme input og derfor alltid yte det samme. Med proporsjonalstyring kan en aktuator få et varierende pådrag. Dette fører til at brukeren kan bruke sine egne muskelkontraksjoner til å styre pådraget.

På det nevro-motoriske laboratoriet har det blitt utviklet programvare for simultan proporsjonalstyring. Dette betyr at i tillegg til å styre med varierende

pådrag er det også mulig å styre flere funksjoner samtidig. Et eksempel på dette kan være åpning og rotasjon av hånden.

3.4 Skjermguidet- og proteseguidet-trening

Skjermguidet-trening (SGT) er implementert i BioPatRec. Dette betyr at brukeren utfører bevegelser under et signalopptak. Disse signalene brukes til å hente ut egenskaper slik at de kan brukes i protesestyring. Brukeren blir guidet i bevegelsene ved å følge instruksene gitt på en skjerm, ofte via bilder og tekst. Slik kan programmet og brukeren være synkroniserte under opptaket.

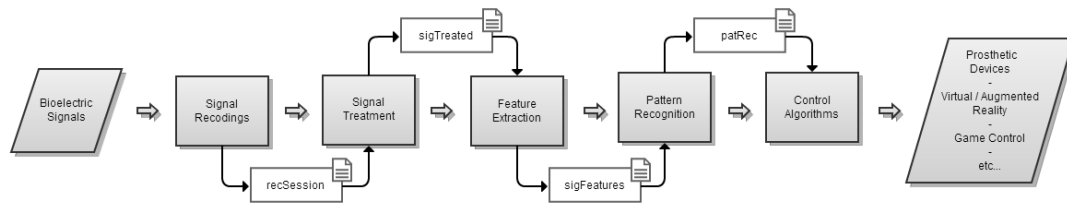
Proteseguidet-trening (PGT) bruker derimot en protese. Dette betyr at en protese gjør bevegelsene sammen med brukeren. På denne måten kan brukeren ikke bare følge med på bevegelsen men kjenne styrken protesen legger i bevegelsen. Dette gjør at brukeren lettere kan følge protesen. Derfor er PGT godt egnet for proporsjonalstyring, ettersom brukeren kan kjenne styrken bak bevegelsen og derfor selv etterlikne styrken og bevegelsen til protesen.

Det bør nevnes at innen forskning er SGT mye mer utbredt enn PGT. Dersom PGT legges til i BioPatRec vil dette muliggjøre at mange som tidligere ikke har hatt muligheten vil få den gjennom BioPatRec. I den kommersielle proteseverdenen brukes hverken SGT eller PGT i stor grad. SGT og PGT er fortsatt bare på forskningsstadiet.

3.5 BioPatRec

BioPatRec er en forskningsplattform utviklet for protesestyring, det vil si et verktøy for testing og utvikling av algoritmer dette feltet. BioPatRec er et åpent kildekodeprosjekt som vil si at alle som ønsker kan benytte seg av programmet, og kildekoden er tilgjengelig for alle slik at de som skulle ønske det kan kopiere og modifisere det ettersom man skulle ha behov for det. Målet med dette er å lage en plattform som alle som arbeider med utvikling av proteser kan benytte seg av, slik at denne programvaren kan bli et felles verktøy for videre protese forskning. Dette kan bidra til å gjøre deling av arbeid mye enklere i fremtiden.

BioPatRec kan studeres videre i (Ortiz-Catalan, n.d.) og i Ortiz-Catalan et al. (2013). Figur 3 gir en oversikt over hvordan BioPatRec er satt opp. Videre i dette kapitlet vil hovedboksene fra figur 3 bli introdusert.



Figur 3: BioPatRec oversikt

3.5.1 Signal recordings

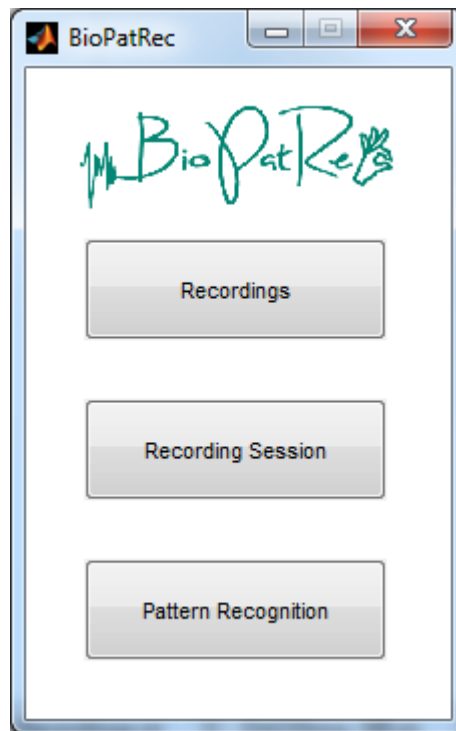
BioPatRec har to muligheter for signalopptak. Figur 4 er et bilde av oppstarts menyen i BioPatRec. De to øverste knappene gir tilgang til de to opptaksmetodene av signaler i BioPatRec. *Recordings* er signalopptaks implementasjonen som gir rask opptak av signaler, så brukeren kan sjekke signalet kvaliteten i sanntid og at alle ønskede sensorer fungerer som de skal. *Recording Session* brukes når brukeren vil ta opp signaler som skal prosesseres videre og benyttes til styring i sanntid senere.

Dataen hentet fra *Recording Session* lagres i strukturen *recSession*. Denne strukturen kan benyttes av alle BioPatRec brukere, og har som fordel at all data lagret i denne strukturen, også kan benyttes av alle andre i BioPatRec uavhengig av hvilke sensorer de benytter seg av så lenge *recSession* blir brukt som datastruktur. Foreløpig er det bare støtte i *recSession* for EMG signaler. Skal annen sensordata lagres for videre bruk, må *recSession* strukturen tillegges nye datasett. Det skal sies at å legge til ekstra variabler i *recSession* ikke vil ødelegge for BioPatRec senere ettersom den kun bruker variabler den selv spør om.

3.5.2 Signal treatment

Signal treatment behandler data fra en *recSession*. Det er her flere mulige rutiner for behandlinger av rådata. Det er her støtte for å fjerne kanaler og bevegelser fra dataen som ikke er signifikante for videre bruk av signalene. Det er støtte for å fjerne deler av muskelsammentrekningen, hviletilstanden i signalene kan automatisk settes som en bevegelse og dataen kan filtreres med et frekvens filter og et romfilter.

Behandlingen av signaler har en egen GUI som automatisk åpnes når brukeren velger en *recSession* å gå videre med. Hvilke signaler som skal benyttes videre og hvordan de skal behandles må derfor bli gjort før brukeren kan gå til



Figur 4: Hovedmenyen i BioPatRec

neste steg i BioPatRec. Den ferdig behandlede dataen fra dette steget blir lagret i strukturen *sigTreated*.

3.5.3 Feature extraction

Denne seksjonen behandler data som har strukturen *sigTreat*. Etter at en *recSession* er valgt og signalene er behandlet, kan brukeren velge hvilke egenskaper (*features*) som skal være med i systemtreningen. Dataen fra *Feature extraction* lagres i *sigFeatures* strukturen. Det er denne strukturen som benyttes videre under mønstergjenkjenningen.

Egenskaper som blir benyttet er for eksempel *zero-crossing*, *median* og *average*. En liste over egenskapene som finnes i BioPatRec er funnet i tabell 1. BioPatRec er bygd slik at implementasjon av egenskaper som ikke allerede er støttet er ganske enkelt. Dette er gjort slik at så mange som mulig skal kunne dra nytte av BioPatRec.

3.5.4 Pattern recognition

I BioPatRec benyttes mønstergjenkjenning i to moduler, offline og online. Offline vil si at trening, testing og validering er mulig ved bruk av signalopptak som tidligere er tatt opp i BioPatRec. Det benyttes altså både offline trening og online trening. Offline treningen er de behandlede signalene som ble tatt opp under *Recording Session*. Egenskapene funnet under denne treningen kan brukes til å trene videre signalopptak i sanntid, dette kalles online trening. Online trening brukes i BioPatRec når en protese styres i sanntid.

Offline trening krever data med strukturen *sigFeatures*. Brukeren får velge hvilke algoritme som skal benyttes til mønstergjenkjenningen. BioPatRec er først og fremst en forskningsplattform, og det er derfor lagt stort fokus på å holde mønstergjenkjenningen modulær. Dette betyr at brukere kan benytte seg av BioPatRec som et verktøy til å prøve ut sine egenutviklede algoritmer. BioPatRec er bygd slik at det er enkelt og lite tidkrevende å implementere sin egen algoritme inn i systemet.

3.5.5 Control algorithms

Denne delen av BioPatRec inneholder protokoller for å benytte mønstergjenkjenningen til å styre en protese. Foreløpig har BioPatRec tre mulige utganger. BioPatRec har muligheten til å fungere med DC-motorer, servomotorer og et virtuelt system designet i BioPatRec kalt *Virtual Reality Environment (VRE)*.

Som nevnt under mønstergjenkjenningen, er det muligheter for å implementere egne algoritmer også her. Det skal være relativt enkelt å benytte BioPatRec for å teste egne algoritmer innenfor styring av proteser, slik at BioPatRec blir et bra verktøy for testing av styringsalgoritmer. Da kan brukeren holde på med sin egen algoritme uten og at for mye tid brukes til å skrive kode som tester den.

ID	Feature	Domain
tmn	Mean	Time
tmabs	Mean Absolute value	Time
tmd	Median	Time
tstd	Standard deviation	Time
tvar	Variance	Time
twl	Waveform length	Time
trms	RMS	Time
tzc	Zero-crossing	Time
tpks	Peaks (over RMS)	Time
tmpks	Peaks mean	Time
tmvel	Mean velocity	Time
tslpch1	Slope changes (peaks)	Time
tslpch2	Slope changes (diff)	Time
tpwr	Power	Time
tdam	Difference abs. mean	Time
tmfl	Max fractal length	Time
tfd	Fractal dimension	Time
tfdh	Fractal dim. Higuchi	Time
tren	Rough entropy	Time
tcr	Correlation	Time
tcv	Co-variance	Time
fwl	Waveform length	Frequency
fmn	Mean	Frequency
fmd	Median	Frequency
fpmn	Peaks mean (Top 5)	Frequency
fpmd	Peaks median (Top 5)	Frequency
fpstd	Peaks std (Top 5)	Frequency

Tabell 1: Tabell med noen av signalegenskapene tilgjengelig i BioPatRec-systemet. Tabellen er hentet fra Ortiz-Catalan (n.d.).

4 Mål

Målet med denne oppgaven er å forbedre det nåværende oppsettet som blir brukt på det nevromotoriske laboratoriet til NTNU. Det er her et stort fokus på proporsjonal simultanstyring og det er ønskelig å forbedre og videreutvikle systemet som til nå blir brukt. Det nevromotoriske laboratoriet bruker i dag en protese designet av Motion Control Inc og sensorsystemet benyttet er laget av Delsys. For en videre innføring disse produsentene er dette mulig i MC (n.d.) og Delsys (n.d.). Programvaren er utviklet i National Instruments (NI) Labview med hensyn til utstyret som er tilgjengelig på laboratoriet.

Programmet har vist seg og fungerer bra på mange aspekter, spesielt med hensyn til realiseringen av proporsjonal simultanstyring. Systemet i Labview er designet til å styre to bevegelser både simultant og proporsjonalt. Motion Control (MC) protesen har som sagt muligheten til å styre to bevegelser, rotasjon av håndleddet og åpning/lukking av håndleddet. Men det er visse aspekter av programmet som trenger forbedringer. Ettersom programmet øker i både størrelse og kompleksitet har det blitt lite oversiktlig. Grafisk programmering kan fort miste lesbarheten når programmet blir stort og dokumentasjonen ikke er god nok. Selv om det har blitt gjort tiltak for å holde koden modulær er det fortsatt store deler av programmet som ikke er så enkle å modifisere eller bytte ut, uten at dette krever store endringer i resten av systemet. Dette kan blant annet observeres i Berrum (2012). Her bør det også nevnes at programvaren som brukes på proteselaboratoriet er skreddersydd til å fungere med utstyret på laboratoriet, og det skal derfor mye til for at programvaren kan benyttes av andre.

BioPatRec er programvaren som skal tillate brukeren å fokusere på sitt felt og fortsatt kunne gjøre de nødvendige testene og øvelsene. Dette bør medføre at brukeren ikke bruker for mye tid på andre implementasjoner. Dette betyr at brukeren har muligheten for å implementere sin egne algoritmer for mønstergjenkjenning og protesekontroll i et fungerende system uten å tenke på for mye annet. Dette betyr at BioPatRec er bygd modulært, slik at koden kan tillegges nye algoritmer uten at noe annet trenger endring. Ettersom BioPatRec også har implementert signalopptak via en *data acquisition* (DAQ) enhet eller en datainnsamlings-anordning betyr dette at veldig mange enkelt kan gjøre seg kompatible med BioPatRec systemet. BioPatRec kan også være mer oversiktlig siden programmet er skrevet i *Matlab*.

Disse egenskapene til BioPatRec er ønskelige her på det nevromotoriske la-

laboratoriet ved NTNU. Studentene som arbeider her kommer og går og det er derfor meget ønskelig med et mer håndterlig program. Det vil også komme fordeler av at koden kan benyttes av andre enn bare oss, på den måten kan vi dele arbeidet gjort her og adoptere andres enklere. For å kunne vite om BioPatRec er kompatibelt med laboratoriet oppsettet må programvaren studeres nærmere. Derfor er oppgaven beskrevet i tre steg.

Et sett med spesifikasjoner må settes som oppfyller alle krav som er nødvendige for at tilfredsstillende resultater kan oppnås på laboratoriet. Det må derfor føres opp krav til utstyret som benyttes. Det må også etableres krav til selve programvaren.

Neste skritt blir så å danne kjennskap med BioPatRec for å avgjøre om programvaren er kompatibel med de kravene som tidligere er bestemt i spesifikasjonene. Dette skrittet er nødvendig for å sikre at ønsket bruk av BioPatRec beskrevet i spesifikasjonen er mulig før programvaren blir tatt i bruk ved proteselaboratoriet. Det er ingen grunn til og benytte et nytt system hvis det ikke oppfyller kravene som er nødvendig for den ønskede funksjonaliteten.

Siste skritt vil bli å modifisere BioPatRec til bruk ved NTNU, slik at BioPatRec kan benyttes aktivt i videre proteseforskningen. Denne oppgaven har fokuset på signalopptak, og det vil derfor bli lagt mer vekt på kravene til signalopptak og implementasjonen BioPatRec relatert til signalopptak.

Del II

Metode

Del II fremlegger gjennomføringen av selve oppgaven. Den er derfor strukturert etter enkeltpunktene i oppgavebeskrivelsen. Videre under selve implementasjonen er modifikasjonene som er utført i denne oppgaven lagt frem. Her er kapitlet strukturert etter fremgangsmetoden som ble valgt. For å holde oversiktene over programvaren som ble modifisert ble det vektlagt å lage implementasjoner som fokuserer på enkelte mål av den fullstendige implementasjonen. På den måten begrenses feilene man kan få av en enkel implementasjon og det er muligheten for å stabilisere en funksjonalitet før nye implementeres. Dette fører til en stegvis prosess hvert steg er beskrevet i underkapitler.

5 Utstyr og funksjonspesifikasjon

Arbeid på det nevromotoriske laboratoriet krever en del utstyr. For å drive proteserelatert utvikling er det mange komponenter som er nødvendig.

Det er nødvendig å ha en metode for signalopptak. Dette krever sensorer. Avhengig av hvilke signaler og signalkvalitet man ønsker er det mye sensorutstyr som kan benyttes. I Berrum (2012) er implementasjonen av ny kraft sensor beskrevet for å se om en slik sensor kan bidra i et protesesystem.

Det er nødvendig å ha programvare som kjører protesesystemet som blir brukt. Og siden det ikke handler om daglig bruk av proteser, men heller om forskning og utvikling av en proteseplattform er det akseptabelt at programvaren kjøres fra en standard datamaskin som har kontakt med alle sensorene og selve protesen.

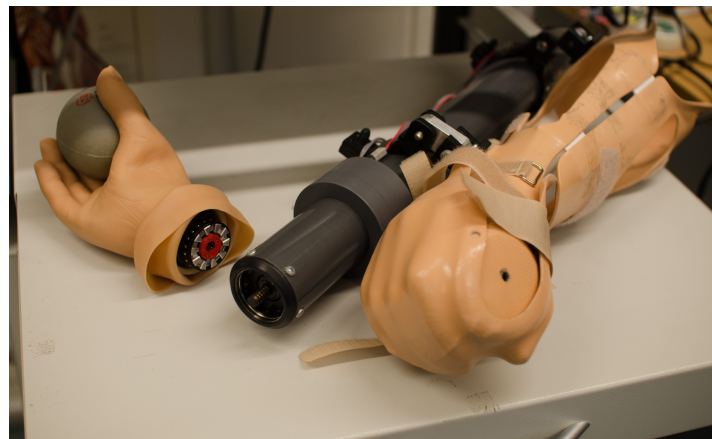
Til slutt er det behov for en protese, så programvaren kan testes og utvikles videre. Det bør her nevnes at i BioPatRec er det designet en virtuell protese slik at det er mulig å teste dine algoritmer uten å faktisk ha en fysisk protese. Dette er en stor fordel med programvaren.

5.1 Protesen

Armprotesen som benyttes på laboratoriet ved NTNU er laget av Motion Control Inc. Det spesielle med protesen er at den er bygget for å benyttes av brukere som ikke har en amputert arm. Figure 5 viser hvordan en slik protese ser ut. Den har



Figur 5: Protesen brukes av proteselaboratoriet til NTNU. Figuren fra er Fougner et al. (utgis senere) og er lisensiert under Creative Commons BY-NC-SA license (heretter forkortet til opphavsrett: CC BY-NC-SA)



Figur 6: Protesen brukt ved det nevromotoriske laboratoriet. Modellene som er brukt er MOtion Control ProHand og Motion Control ProWrist

et hylster for brukerens arm montert parallelt med protesen. Her bør det også nevnes at protesehånden er montert lengre fremme enn brukerens hånd. Dette er gjort slik at brukeren alltid kan ha visuell kontakt med med protesehånden.

Protesen har to frihetsgrader, den har en aktuator for rotasjon av hånden og en aktuator for åpning og lukking av hånden. Protesen tar inngangs signalene sine fra programvaren som kontrollerer den via en DAQ-enhet av National Instrument. Mer informasjon om protesen og produsenten studeres nærmere i MC (n.d.). Det er også et bilde av protesen i figur 6 og DAQ-enheten benyttet er vist i figur 7.



Figur 7: Her er DAQ-enheten brukt ved det nevrologiske laboratoriet. Modell nummer: cDAQ-9191

5.2 Sensorene

Sensorene tilgjengelige på laboratoriet er designet av Delsys. Delsys Trigno Wireless System (DTWS) er sensorer designet spesielt for forskning. Sensor systemet inneholder totalt 16 sensorer i en dokkingstasjon. Den kan kobles analogt via en DAQ eller den kan kobles direkte til en datamaskin ved en Universal Serial Bus USB kabel. Tidligere har USB blitt benyttet og den vil også bli implementert i denne oppgaven. Hver sensor er trådløs og har trådløs kontakt med dokkingstasjonen.

En av Trigno sensorene kan måle MES og akselerasjon i tre retninger. De myoelektriske signalene blir tatt opp som SEMG. DTWS operer generelt med to frekvenser den samler data på, 2000 og 4000 Hz. Dette er samplehastigheten for EMG-dataen. Akselerometerdataen kommer ikke inn med samme frekvens. Forholdet mellom akselerometerdata EMG-data er 13.5:1. Dette betyr at det er ca. 148 nye sampler med akselerometer data når det er 2000 EMG sampler. Akselerometeret måler akselerasjonen i tre akser og vil derfor returnere tre verdier for hvert sample som blir tatt.

Tidligere har NTNU eksperimentert med andre typer sensorer. Et eksempel er kraftsensor som ble utviklet til og fungerer med DTWS. Denne kan studeres nærmere i Suleng (2012) og den ble testet og implementert i ITK-programvaren i Berrum (2012). Denne oppgaven vil fokusere mest på DTWS siden disse sensorene måler den mest essensielle dataen. Figur 8 viser et bilde av DTWS



Figur 8: Bildet over viser Delsys Trigno Wireless System. Model Informasjon: FCC ID:W4P-SP-W01 (Sensorer), FCC ID: W4P-Sp-W02(Base Station) og IC: 818A-DTST01 (System).

5.3 Programvare

Tidligere på laboratoriet har Labview av NI vært utviklingsverktøyet, men som nevnt i kapittel 4 er det ønsket å bytte ut denne. BioPatRec er skrevet i Matlab derfor blir Matlab den nye plattformen for dette prosjektet. Arbeidsstasjonene på proteselaboratoriet er satt opp med 2012 versjonen av Matlab.

Opptakene sendes fra DTWS til dokkingstasjonen. Her blir de sendt til datamaskinen og kan derfor benyttes av programvare som Matlab og Labview så lenge datamaskinen kjører en Delsys Trigno Control Utility (DTCU) programvare i bakgrunnen. I Matlab hentes dataen fra en tabell som har lengden til antall samples som ikke er hentet ganger antall sensorer totalt. Dette vil si sensordataen kommer i en rekke og at alle sensorer alltid returnerer verdier. Skal man ha ut samplene fra en sensor kan man bare inkludere hvert 16. sample for SEMG data og tre samples vært 48 sample for akselerometerdata. Dette betyr også at selv om du er interessert i dataen fra et fåtall av sensorene vil du alltid få data fra alle, sensorene som ikke er aktive vil bare returnere null.

5.4 Funksjonspesifikasjon

Å sette opp funksjonspesifikasjon er en fornuftig metode for å bedømme om BioPatRec er programvare som kan benyttes på proteselaboratoriet med utstyret som er tilgjengelig hos NTNU. Funksjonspesifikasjonen ble satt opp i samarbeid med Alf Bertnum, som også jobber med BioPatRec.

Bertnum fokuserer derimot på intensjonsestimering og systemtreningen i BioPatRec, der arbeidet i denne oppgaven fokuserer på signalopptaket. Derfor ble den fullstendig funksjonspesifikasjonen satt sammen av kravene begge hadde satt. Dette betyr funksjonspesifikasjon blir mer detaljert siden begge setter kravene for sitt fokus i BioPatRec. Et godt utgangspunkt er funksjonspesifikasjonen fra Linnerud (2012) , funksjonspesifikasjonene der var satt opp med hensyn til et laboratoriet som er satt opp for og støtte proporsjonalstyring av proteser. Spesifikasjonene som ble satt for BioPatRec er beskrevet i tabell 2.

Funksjonspesifikasjonene er basert på Losiers modellen som har blitt tatt videre i Fougner et al. (2012). Figur 9 viser denne modellen.

	Requirements	Description
Preprocessing	R1	The sensor input of the system MUST be modular and with the possibility to combine sensors.
	R1.1	Data from the Trigno base MUST be transferred through USB.
	R1.1.1	MUST function in real-time with at least 5xEMG and 5x3xACC. (5 Trigno sensors)
	R1.1.2	SHOULD be real-time connection for more then 5 sensors.
	R1.1.3	SHOULD be independent of third-party-software for reading data from USB.
	R1.2	Additional sensors SHOULD be easily implemented.
Intent interpretation	R2	The possibility of using different intent interpretations MUST be available.
	R2.1	All control systems MUST be modular to more easily implement new ones and to edit existing ones.
	R2.2	Pattern recognition MUST be a possible option as a control system. (Should probably be crisp classification since pattern recognition is quite a wide term.)
	R2.3	Proportional control MUST be a possible option as a control system.
	R2.3.1	MUST be possible to manipulate the prosthesis input to control the force, velocity and/or position.
	R3	The new prosthesis platform MUST be sufficiently compatible to function properly with the prosthesis set-up that is currently at place in the prosthesis lab at NTNU.
	R3.1	The system MUST be able to use a communication protocol.

	Requirements	Description
Output	R3.1.1	It SHOULD be compatible with the NI DAQ devices.
	R3.1.2	It SHOULD be able to communicate through Bluetooth protocol.
	R3.1.3	It MUST work in real-time.
	R3.2	Continued use of the departments robot hand SHOULD be possible.
	R3.3	MUST work with the Motion Control prosthesis.
	R3.3.1	MUST work in real-time.
	R3.3.2	SHOULD be able to take advantage of all functions within the prosthesis.
System training	RS	MUST support alternative types of system training.
	RS.1	MUST support prosthesis guided training (PGT).
	RS.1.1	MUST have the possibility for exercise sets.
	RS.1.1.1	MUST support training through force-, velocity- or position-control.
	RS.1.1.2	MUST be optional to choose between the sets.
	RS.1.1.3	MUST be intuitive motions to follow.
	RS.2	MUST support screen guided training (SGT).

Tabell 2: Tabell med funksjonsspesifikasjon for et potensielt protesesystem. Funksjonsspesifikasjonen er adaptert fra Linnerud (2012).

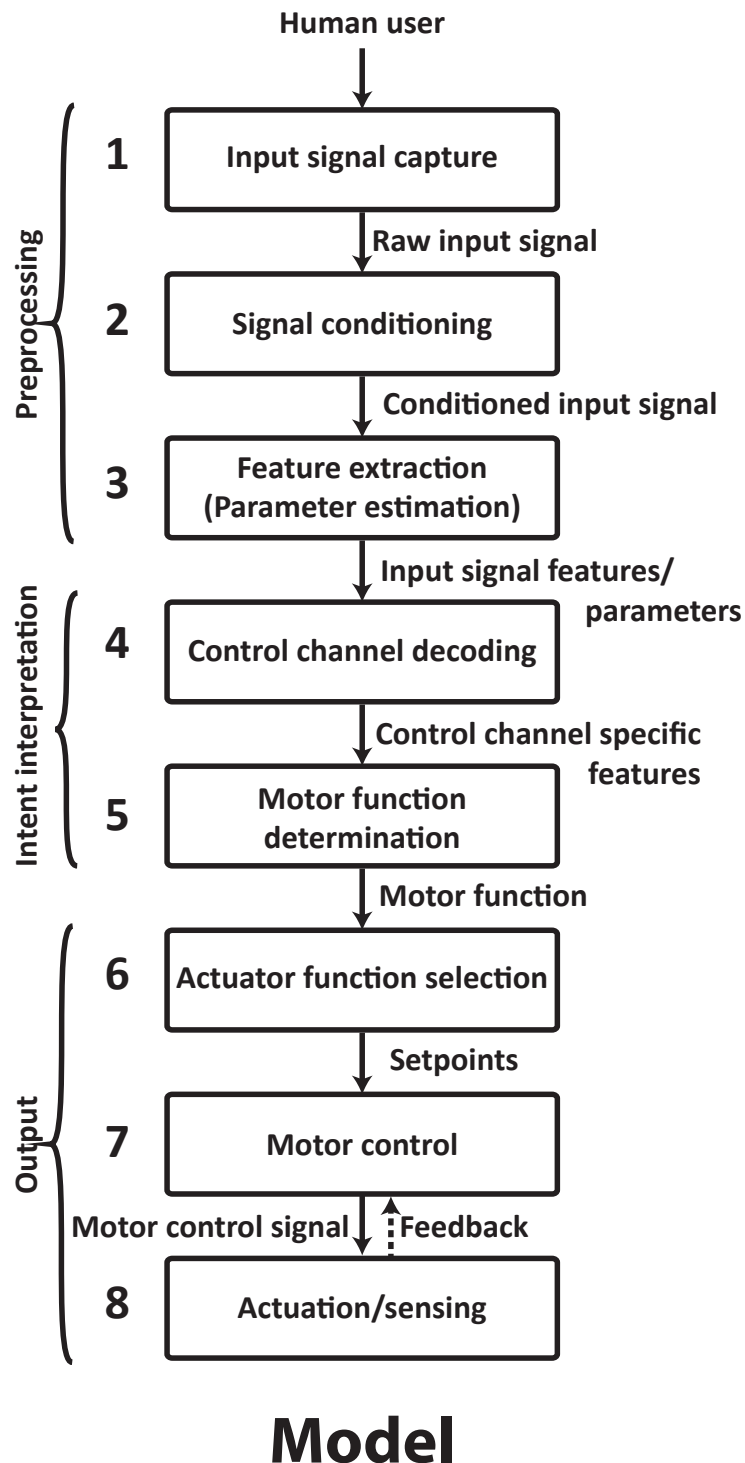
Funksjonspesifikasjonene ble delt inn i 4 grupper der de tre første gruppen følger Losiers modell. R1 er *Preprocessing*, R2 er *Intent interpretation*, R3 er *OutPut* og R4 er *System training*.

Preprocessing - R1 eller preprosessering omfatter krav til signalopptak og bruk av sensorer. Her er fokuset at flere typer signalopptak må kunne brukes og dette må skje i sanntid, og sensor implementasjonen i BioPatRec skal være modulær.

Intent interpretation - R2 eller intensjonsestimering setter krav til implementasjonene av kontrollsystemer i BioPatRec. Her er det viktig at kontrollsystem delen av BioPatRec er modulær nok til at nye kontrollsystemer kan innføres enkelt. Det er også viktig med mønstergjenkjenning og Proporsjonal styring må være mulig og kunne utnytte gjennom BioPatRec.

OutPut - R3 setter krav til at BioPatRec må være kompatibelt med oppsettet på proteselaboratoriet ved NTNU. Den må kunne fungere med DAQ-enhetene tilgjengelige og med protesen vi bruker, og at den videre støtter full funksjonalitet i utstyret.

System training - RS eller systemtrening setter krav til at BioPatRec må kunne støtte system trening. Det må være mulig å bruke BioPatrec til både PGT og SGT. Tidligere har det blitt lagt større fokus på utvikling av PGT på det nevro-motoriske laboratoriet ved NTNU.



Figur 9: Dette er en videreutvikling av Losiers modell. Den består av de tre hovedkategoriene preprosessering, intensjonsestimering og output. Denne modellen illustrere en klassevisning av et protesestyringssystem. Modellen over er tatt fra Fougner et al. (2012).

6 Programvarevurdering

Nå som funksjonspesifikasjonene er satt er det viktig at BioPatRec blir vurdert med hensyn på disse kriteriene. Funksjonspesifikasjonen vil gi et klart bilde av hvor godt egnet BioPatRec er for oppsettet som på NTNU sitt proteselaboratoriet. Siden oppgavens hovedfokus er signalopptak kommer preprosesseringen til å vektlegges høyest. Det vil derfor bli en nøye gjennomgang av punktene som er relatert til signalopptak og en mer generell gjennomgang av resten av systemet.

6.1 Preprosessering

For å få et bilde av hvor godt egnet BioPatrec er for DTWS er det viktig å se på hvordan sensor oppsettet i BioPatRec er sammenliknet med hvordan Matlab implementasjonen av Trigno sensorene er. Med DTWS følger *software development kit* (SDK) tester til forskjellige programmeringsspråk. Disse SDK testene er ganske enkle eksempler på hvordan å ta i bruk Trigno sensorene til å hente data og plote den for brukeren. SDK testene inkluderer blant annet C, Labview og Matlab. SDK testen i Labview ble brukt under utviklingen av Labview programvaren siden den er en guide til hvordan kommunikasjon mellom Labview og DTWS er satt opp. Det samme gjelder for Matlab. SDK testen kan gi en god indikasjon på hvor godt sensorene vil fungere i BioPatRec. SDK testen i Matlab kan finnes i Appendiks A. BioPatRec sin implementasjon er å finne i Appendiks A. Dette er koden bak *Recordings* fra figur 4. Denne delen av programmet blir mest brukt til testing av sensorene, og dataen som tas opp her kan ikke senere bli brukt i BioPatRec ettersom den ikke lagres i en *recSession* struktur. Vi kan observere fra koden til BioPatRec at opp til 8 grafer blir initialisert, avhengig av hvor mange sensorer du har valgt og kjøre. Systemet håndterer altså maks åtte aktive sensorer.

Et stort fokus i BioPatRec er modularitet, dette er i henhold til punkt R1. Dette med spesielt hensyn til styringsalgoritmer. Implementasjonen av nye sensorer er ikke så rett frem som implementasjon av for eksempel en mønstergjenkjenning algoritme, men det er helt tydelig at BioPatRec er utviklet med tanke på modularitet også når det kommer til sensorer. *Recording Session* er bygd modulært, ved at flere *if* sjekker blir benyttet til å skille mellom sensorer. Det kan derfor legges inn nye sensorer ved å utvide *if* sjekken. Slik at når du velger et sensorsystem i starten vil kun hendelsene for dette systemet bli aksessert i koden. Derimot er det store forskjeller i implementasjonen av sensorer som benyttes av BioPatRec i Matlab og Delsys sin egen implementasjon av Trigno sensorene. Det viser seg at BioPatRec bruker egne tidsstempler. Tidsstempler

kan for eksempel sette verdiene langs x-aksen i plottet. Om det blir hentet 1000 verdier fra en sensor vil x-aksen gå fra 0 til 1000. Hvis det derimot er ønskelig og beskrive tiden passert langs x-aksen kan egne tidsstempler settets. BioPatRec venter på at nok data er samlet inn fra sensorene, da blir en funksjon utført som henter og lagrer sensordataen samtidig som den oppdaterer listen av tidsstempler. Dette skjer altså i samme tråd.

Delsys sin SDK test starter to tråder, en tråd som venter på nok data for så å åpne en funksjon som tar dataen og lagrer den. Den andre tråden er en *timer function (timerFcn)*. *timerFcn* lar deg spesifisere en funksjon som skal utføres per tidsenhet. Delsys bruker *timerFcn* til å plote data samlet opp 10 ganger i sekundet. Problemet blir da at siden funksjonen som henter data og funksjonen som plotter er i to forskjellige tråder, noe som gjør det vanskelig å lage tidsstempler. Ikke bare fordi det er snakk om forskjellige tråder men også siden funksjonen som plotter er tidsbasert, og det er derfor vanskelig å predikere hvor mange tidsstempler som er nødvendig. BioPatRec bruker først og fremst tidsstemplene til å oppdatere en tidslinje slik at brukeren vet hvor lenge det er igjen av en øvelse.

Punkt R1 nevner også at flere sensorer må kunne kombineres. BioPatRec har bare muligheten for å ta opp EMG-data. Men *recSession* er bygd slik at du kan legge til egen data uten å ødelegge strukturen. Dette vil si at det er mulig å legge inn nye data typer uten at dette har konsekvenser for BioPatRec sin oppførsel.

Punkt R1.1 er tilfredsstilt ettersom BioPatRec ikke vil ha noe problem ved og bruke USB for og opprettholde kontakten med DTWS. Fra *Recordings* koden kommer det frem at den BioPatRec bruker en DAQ-enhet for og kommunisere med sensorer, men det er ingen ting i koden som tilsier at dette er et krav. Det vil si at BioPatRec i seg selv ikke er bundet til en DAQ-enhet løsning, å få endringer er nødvendig for å implementere sin egen sensor driver.

Fra *Recordings* og *Recordings Session* kommer det frem at det er mulig med åtte kanaler. Det vil si at BioPatRec kan gjøre opptak fra åtte sensorer. Dette er mest sannsynlig begrenset til ytelsen i BioPatRec. R1.1.1 sanntids håndtering av fem SEMG målinger og fem akselerometer målinger. Dette betyr at BioPatRec tilsammen må håndtere 20 forskjellige signaler. Det burde også vektlegges at DTWS henter data fra alle 16 sensorene uansett hvor mange som er koblet til. Dette kan derimot motvirkes til en grad med og raskt sortere ut den dataen som er fra de aktive sensorene. Slik at den overflødige dataen ikke blir med videre i signal håndteringen. Dette gjør at vi må ta en dypere sjekk av BioPatRec. Sensorene som er implementert i BioPatRec kan klokkes fra 2000Hz til 8000Hz. Trigno sensorene er klokke til 2000Hz for SEMG-data og akselerometeret er klokke til 148.15Hz. Selv om akselerometeret returnerer tre verdier per sample er dette fortsatt ganske lite data, og langt innenfor evnen til hva BioPatRec

håndterer.

Begrensningen på åtte kanaler er mest en grensesnitt begrensning. Siden programvaren bare er satt opp for og vise brukeren åtte signaler grafisk. Ytel- ses problemet vårt kan derfor være løselig ved at under signalopptak blir bare SEMG dataen vist grafisk. Akselerometerdataen og andre typer data i fremti- den, trenger ikke vises. Dette gir bedre ytelse og har ingen konsekvenser for selve opptaket. R1.1.1 virker derfor gjennomførbart.

Krav R1.1.2 blir en utvidelse av R1.1.1. Trolig vil BioPatRec klare mer enn fem Trigno sensorer, men ytelse kan bli et problem uansett og ettersom sensorer blir lagt til kommer nok mye av den tilgjengelige datakraften til å bli brukt. Men det er ikke sikkert det blir de største ytelses endring ved å legge til flere Trigno sensorer ettersom bare spesifikke deler av sensordataen plottes og data fra alle sensorer blir hentet. SDK testen som Delsys skrev i Matlab plotter all data fra alle sensorene. Den bruker da en del ressurser men gjør det fortsatt i sanntid. I BioPatRec vil det aldri bli nødvendig og plote data fra 16 sensorer.

For og holde kontakten med DTWS kreves som tidligere nevnt et tredjeparts programmet DTCU som er laget av Delsys, Dette gjør at R1.1.3 ikke kan oppfyl- les, men dette kravet er ikke en absolutt nødvendighet. Programvaren i Labview brukte også dette tilleggsprogrammet for og fungere.

Krav R1.2 er heller ikke en nødvendighet. Det å kunne legge til nye typer sensorer etter Trigno bygger på at BioPatRec må være modulært og fortsatt være det etter at DTWS er implementert. På grunn av *if* sjekkene som nevnt tidligere i *Recording Session* bør ikke dette være et problem. Den neste sensoren som implementeres i BioPatRec er kraftsensoren som er eksperimentert med i Labview tidligere. Denne kjøres nemlig gjennom en DAQ-enhet og er derfor meget kompatibel med BioPatRec.

6.2 Intensjonsestimering

Som sagt er BioPatRec bygd rundt modularitet til styringsalgoritmer. Dette betyr at kravene her blir enkelt tilfredsstilt. Dette er en kategori som gjorde BioPatRec til en attraktiv programvare i første omgang. Mønstergjenkjenning er allerede implementert i BioPatRec og mønstergjenkjenningsalgoritmer bør derfor være relativt enkelt å implementere. På Ortiz-Catalan (n.d.) kan du enkelt finne en kort guide til hvordan disse bør endres. Proporsjonal styring bør være imple- menterbart. Det er ikke med i BioPatRec fra før av og i Labview programvaren har den en helt egen implementasjon. Dette betyr at for å inkludere proporsjo- nalstyring slik den var tidligere kreves en mer omfattende implementasjon en det som er beskrevet på BioPatRec sin egen hjemmeside. Alf Bertnum sin opp-

gave fokuserer på *intent interpretation* og en mer detaljert innblikk i BioPatRec med hensyn på disse kravene da spesielt med tanke på proporsjonalstyring er tilgjengelig i Bertnum (2013).

6.3 Output

Det er viktig at mesteparten av utstyret som står på proteselaboratoriet kan bli benyttet videre i BioPatRec. Det er med tanke på dette at kravene i *Output* delen av funksjonspesifikasjonene er satt. Det er nødvendig at MC protesen fungerer med full funksjonalitet. Etersom både BioPatRec og ITK-programvaren benytter NI DAQ-enheter vil det være få problemer med å bruke protesen. Det kan være at bruk av forskjellige typer DAQ-enheter gjør at en endring i kommunikasjonsprotokollen er nødvendig. Dette er bare programvareendringer som ikke vil være omfattende eller endre modulariteten til resten av programmet.

Det har vært et ønske å legge til støtte for blåtann, slik at kommunikasjonen mellom programvaren og protesen kan gjøres trådløs. Driveren til MC protesen har kun støtte for blåtann i oppdatering av programvare. Det betyr at en driveroppdatering er nødvendig for å gå videre med dette. Den nye driveren må enten skrives av Motion Control eller så må oppgaven utføres på NTNU. Det største problemet ligger nok med driveren til protesen og ikke i BioPatRec. Det er her først og fremst et spørsmål om kompatibiliteten i protesedriveren.

For de samme grunnene til at MC protesen vil være kompatibel med BioPatRec vil også robothånden være det. Det må derfor utvikles en styringsalgoritme for bruk av den i BioPatRec.

Det å kunne kjøre protesen i sanntid vil nok ikke by på store problemer. Dette fordi BioPatrec sine mer omfattende operasjoner ikke inkluderer output. Det er viktig å merke seg at de samlede operasjonene kan bli omfattende. Ved styring av protesen i sanntid skjer det mye samtidig, og mer omfattende funksjonalitet som signalopptak kan føre til problemer. Sanntids protesestyring er ikke flaskehalsen i systemet.

6.4 System training

Til nå i BioPatrec er det god støtte for SGT. SGT er brukt under *Signal Recordings* i figur 3. Når en *Recording Session* starter vil brukeren bli fortalt på skjermen med tekst hvilke oppgave han skal utføre, dette vil også bli illustrert med bilder.

Det er ingen fullstendig PGT løsning implementert i BioPatRec. Dette er forventet ettersom PGT er et meget nytt konsept, og det er til nå ikke så mange

som anvender dette. Det BioPatRec har innført er muligheten til å velge *dummy repetition* under *Recording Session*. Dette gir muligheten for å få bevegelsen illustrert av selve protesen er tilgjengelig. I en PGT løsning vil brukeren være guidet av protesen. Dette betyr at protesen og brukeren vil utføre bevegelsen samtidig. Selv om dette ikke er situasjonen for øyeblikket i BioPatRec er det helt tydelig muligheter for en PGT implementasjon i *Recording Session*.

Det bør også nevnes at PGT er spesielt egnet for treningssett som skal brukes til proporsjonalstyring. Altså styring av protese som ikke bare definerer bevegelsen som må utføres, men også kraften. Det er enklere for brukeren å føle styrken på bevegelsen når det ikke bare er mulig å observere bevegelsen. Et godt eksempel på dette er åpning og lukking av hånden. Siden ikke proporsjonalstyring er implementert krever dette at datasettene i BioPatRec modifiseres for støtte av proporsjonaldata.

6.5 Konklusjon av programvarevurderingen

Det er tydelig at en implementasjon av BioPatRec på det nevrologiske laboratoriet vil være tidskrevende. Det kommer klart frem at det må forekomme implementasjoner i flere seksjoner av BioPatRec for at programvaren skal kunne brukes som et godt alternativ på laboratoriet. Det er derimot ingen av oppgaven som har vist seg umulig, så selv om BioPatRec vil kreve mye arbeid for å bli kompatibel med resten av utstyret er det ingen ting som hindrer en slik implementasjon.

Det er derfor konkludert med at BioPatRec egner seg for bruk ved NTNU. Denne oppgaven vil derfor gå videre med implementasjonen av DTWS. Alf Bertnum sin oppgave hadde fokus på intensjonsestimering, og har vurdert BioPatRec med spesielt hensyn til dette, der denne oppgaven har fokusert spesielt på inngangssignaler. Med forskjellig fokus har konklusjonen blitt den samme, og BioPatRec blir derfor tatt videre i begge oppgavene.

Om en fullstendig implementasjon av BioPatRec blir fullført kan man begynne å ta opp arbeid fra andre uten å gjøre noe ekstra. Etersom det er mange der ute som implementerer for BioPatRec kan det nå enkelt dras nytte av andres arbeid. Tidligere var laboratoriet begrenset til å måtte utvikle en implementasjon av slikt arbeid selv.

7 Implementasjon

Under programvarevurderingen i kapittel 6 ble BioPatRec godkjent for bruk. BioPatrec krever en rekke modifikasjoner for å fungere i henholdt til hva som er ønsket. Derfor vil dette kapitlet ta for seg implementasjonen som er gjort i denne oppgaven i detalj. Underveis vil noen egenskaper og funksjonaliteter ved BioPatRec bli forklart ettersom det er nødvendig for å sette seg inn i implementasjonen.

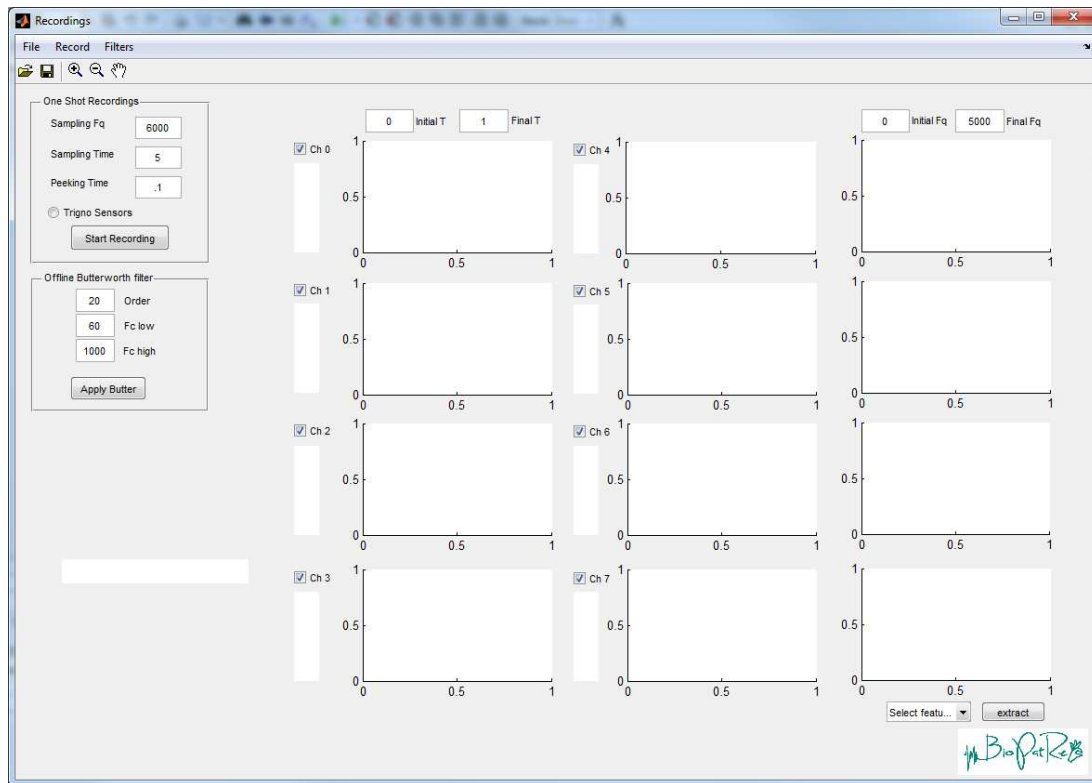
Opgaven har hatt stort fokus på signalopptak som kommer under preprocessingen. Det faller derfor naturlig at denne oppgaven vil ta for seg implementasjon av sensorer og riktig konfigurering av signalopptak. *Recording Session* hovedmodulen BioPatRec bruker til signalopptak for videre bruk, denne må derfor modifiseres for å fungere med DTWS. Ettersom BioPatRec er et meget komplekst verktøy kan derfor *Recordings* delen av BioPatRec bli brukt for å komme frem til en løsning som kan adapteres inn i *Recording Session*. *Recordings* og *Recording Session* er hovedmoduler av BioPatRec som er tilgjengelig fra hovedmenyen vist i figur 4, der *Recordings* bare er en ren fremvisning av signalene fra ønskede sensorer.

7.1 Recordings

En viktig egenskap med BioPatRec er at både *Recordings* og *Recording Session* benytter seg av det samme grensesnittet som er vist i figur 10. Dette betyr at modifikasjoner som implementeres under *Recordings* er enklere å ta videre til *Recording Session*. Derfor blir alle sensormodifikasjoner først implementert i *Recordings*. Dersom den fullstendige implementerte løsningen skulle være stabil nok, kan den derfra overføres til *Recording Session*.

Koden brukt av BioPatRec under et opptak tatt fra *Recordings* modulen finnes i Appendix A. Ettersom denne koden ikke inneholder *if* sjekkene som er inkludert i *Recording Session* vil koden som håndterer DTWS være en separat fil som baserer seg på koden fra *Recordings*. Her vil implementasjonen av DAQ-enheten byttes ut med implementasjonen av DTWS. Denne implementasjonen har fått navnet *Trigno Session*.

Figur 10 viser at det er et alternativ for Trigno sensorene før brukeren velger *Start Recording*. Dette alternativet setter hvilken Matlab funksjon som blir utført. Om den blir forlatt blank vil koden BioPatrec åpne *DAQShow-SBI* som er den vanlige funksjonen BioPatRec kjører under *Recordings*, men dersom den blir aktivert vil BioPatRec åpne en Matlab funksjon, modifisert for bruk med Delsys. Man må passe på at riktig sampling er valgt om man velger Trigno sen-



Figur 10: Dette er grensesnittet for opptak av signaler. Dette grensesnittet er tilgjengelig via *Recordings* og *Recording Session*

sorene, ettersom de bare kan benyttes ved 2000Hz i denne programvaren. Figur 10 viser også alle vinduene for signalvisning. De åtte første vinduene er til nå brukt av BioPatRec for å vise EMG signaler. Disse vinduene kommer med en bar som illustrerer signalstyrken, en funksjonalitet som bare blir tatt i bruk i *Recording Session*. Vinduet kommer også med en *checkbox* slik at brukeren kan velge å bruke denne sensoren som hadde blitt representert i dette vinduet, dette er en funksjonalitet som bare blir benyttet i *Recordings*. De fire siste vinduene gir signalet i frekvensspekteret.

7.1.1 Første implementasjon: SEMG signaler

I første implementasjon var hovedfokuset å komme med en løsning som først og fremst klarer å vise data fra DTWS i BioPatRec. Det ble nevnt under programvarevurderingen i kapittel 6.1 at det ville oppstå problemer med tidsstempel funksjonaliteten i BioPatRec. Dette ble derfor ikke tatt med i den første versjonen. Dette gjorde at det var mulig og få en fungerende BioPatrec løsning. Så kan heller koden tilpasses for tidsstempelen siden. Delsys sin SDK test er funnet

i Appendiks A ble benyttet som et utgangspunkt for denne implementasjonen. Det finnes ingen manualer eller liknende forklaringer til hvordan å bruke disse sensorene i Matlab. Delsys sin SDK test blir derfor den eneste metoden for å sette seg inn i bruken av DTWS i Matlab.

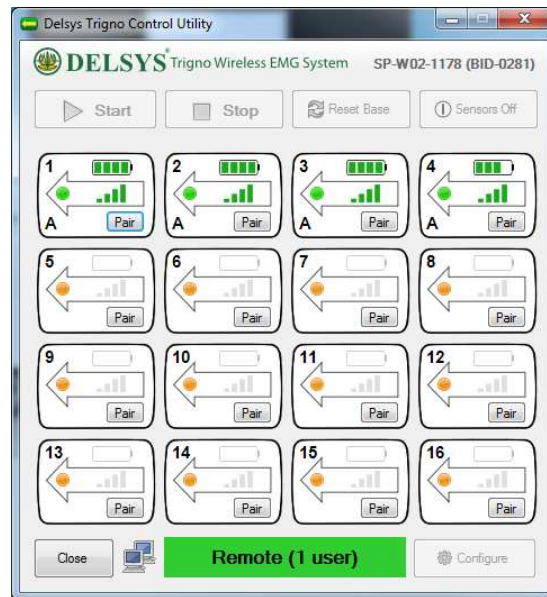
DAQShow-SBI benytter seg av en funksjon som henter og plotter data. Denne funksjonaliteten ble byttet ut med to funksjoner. *ReadEMG* som har ansvaret for å hente inn ny data fra sensorene når denne dataen er tilgjengelig. *updatePlots* er funksjonen som har ansvaret for å plote dataen. Her var det nødvendig å lage en funksjonalitet som både klarer å ta inn data fra Trigno sensorene og som også klarer vise frem dataen i sanntid. Dette ble gjort enkelt ettersom det ikke ble benyttet tidsstempler i denne versjonen.

Initialiseringen av DTWS skjer der DAQ-enheten tidligere ble initialisert i *DAQShow-SBI* funksjonen. Her blir det introdusert tre *tcpip* objekter som i Matlab er objekter som assosierer en ekstern vert, med en ekstern port. I dette tilfelle er den eksterne verten datamaskinen som kjører BioPatRec. EMG-data, akselerometerdata er tilgjengelig gjennom to porter i de to første *tcpip* objektene. Den tredje porten gjør kommunikasjon med Delsys sin SDK mulig. Dette objektet brukes for å gi kommandoer slik at BioPatRec kan starte og slutte å ta imot data fra sensorene. Delsys sin SDK inkluderer DTCU, eller Delsys Trigno Control Utility som er nødvendig programvare som vil kjøre i bakgrunnen. Denne programvaren er en servere som styrer sensorene. Programvaren viser også en oversikt over alle sensorene. Et bilde av DTCU er gitt i figur 11. *tcpip* objektene setter opp kommunikasjon med portene med *fopen*. Etter dette kan datahenting startes ved å sende kommandoer til DTCU. Det ble også satt opp krav til hvor mye data som skal være tilgjengelig før *ReadEMG* henter dataen. Det blir også satt opp en *timerfcn*. Dette objektet ble nevnt i kapittel 6.1, og blir satt opp med *updatePlots* slik at ny data blir plottet jevnlig. I den versjonen ble *updatePlots* kalt 10 ganger i sekundet.

Den fullstendige koden til den første implementasjonen finnes i Appendiks A med navnet *TrignoSession1*. I *TrignoSession1* ble det bare *tcpip* objektet for EMG-data inkludert. Dette betyr at i første omgang blir akselerometerdataen ignorert.

7.1.2 Andre implementasjon: Opprydning av initialisering

Den andre implementasjonen som ble gjennomført hadde ingen forandringer som endrer noen resultater. Nye implementasjoner vil være basert på den forrige implementasjonen, der målet med den nye implementasjonen er å forbedre den eller legge til noe som mangler. I denne implementasjonen var fokuset å rette opp i noe av kodenstrukturen. *DAQShow-SBI* hadde et veldig enkelt oppsett



Figur 11: Delsys Trigno Control Utility (DTCU). Programvaren gir en del informasjon til brukeren. Det er mulig å se batterinivået til enkelt sensorene samt hvor godt det trådløse signalet er til hver sensor. Det er også mulig og se om noen benytter seg av sensorene. Dette bildet ble tatt når BioPatRec benyttet fire av sensorene

av DAQ-enheten. Dette fordi mye av initialiseringen av DAQ-enheten var flyttet til en egen funksjon. Dette ser veldig ryddig ut og det er enklere og håndtere resten av koden med en slik strukturell forandring. Derfor ble oppgaven bak denne implementasjonen å skrive en egen funksjon som initialiserer DTWS. Det vil si en funksjon som setter opp de tre *tcpip* objektene og returnerer de slik at resten av koden kan benyttet dem til å hente signaldata. Oppsett av *updatePlots* og *ReadEMG* må fortsatt gjøres i hoved funksjonen *Trigno Session*, ettersom disse funksjonene er lokalisert i samme fil som denne. Den nye versjonen av *Trigno Session* finnes i Appendiks A som *TrignoSession2*.

Initialiseringsfunksjonen kalt *TrignoInit* inneholder nå oppsettet til de tre *tcpip* objektene. Den inkluderer også en kode som finner ip-adressen til data-maskinen som kjører BioPatRec ettersom du trenger denne ip-adressen for å sette opp alle *tcpip* objektene. Den returnerer så alle objektene, slik at *Trigno Session* kan bruke objektene som normalt. Appendiks A inneholder *TrignoInit* som beskriver initialiseringen.

7.1.3 Tredje implementasjon: Frekvensspekteret

Den tredje versjonen av *Trigno Session* ble laget for å dekke mer av funksjonaliteten i BioPatRec. Som nevnt tidligere er de fire vinduene lengst til høyre i figur 10 frekvensspekteret til signalet. Siden det bare er fire av disse vinduene, men åtte vinduer for de faktiske signalene er det kun de fire vinduene lengst til venstre som har muligheten for å få inkludert frekvensspekteret. Frekvensspekteret bærer ingen videre funksjon i BioPatRec. Dette fordi dataen her ikke blir lagret og tatt med videre i systemet. Det å plotte signalet under et signalopptak kan være nyttig ettersom det er lettere å oppdage støy i frekvensspekteret.

For å inkludere frekvensspekteret må initialiseringen av plottene inkludere dette. Når dette er gjort må frekvensspekteret inkluderes i *updatePlots*. Plotting av data er til nå gjort ved at dataen fra alle sensorene kommer i en lang rekke. Dette betyr at *updatePlots* bare plotter være 16. verdi i et vindu. Dette fordi det bare er ønskelig å ha data fra en sensor i et vindu. Dette ble endret slik at dataen først blir formatert til en tabell som har likt antall kolonner som aktive sensorer. Slik blir dataen til sensorer som ikke er aktive filtrert ut og en kolonne har bare data fra en sensor. Dette simplifiserer koden der sensordataen plottes.

For å generere frekvensspekteret til signalopptakene benyttes en *fast fourier transformer* (FFT).

Det bør også nevnes at i tidligere implementasjonen var den eneste dataen som ble tatt vare på den dataen som ble plottet av *updatePlots*. Den samlede dataen fra hele opptaket har ikke tidligere blitt lagret, kun dataen som blir plottet. Denne mengden er satt til 32832 sampler for alle sensorene. Dette betyr at det er 2052 sampler i et enkelt vindu. Dette betyr at et enkelt vindu plotter over en tidsperiode på tilnærmet et sekund ettersom sensoren har en samplingsfrekvens på 2000Hz. For en lengre økt går derfor mye data tapt. Dette er rettet opp her ved og introdusere *allData* som inneholder all data tatt opp av alle sensorer for en økt. *allData* blir da introdusert i tillegg til *data-arrayEMG* som bare inneholder data som fortsatt er inkludert i plottet. Den tredje versjonen av *Trigno session* er vedlagt i Appendiks A som *TrignoSession3*.

7.1.4 Fjerde implementasjon: Tidsstempler

Denne implementasjonen fokuserer på å generere egne tidsstempler. Til nå har det ikke blitt gitt, og x-aksen i signalopptaket vil derfor bare være sample nummeret i stedet for tiden, som ofte er den ønskede verdien langs x-aksen. Dette vil si at etter et sekund med signalopptak vil det stå 2000 som da blir antall sampler i enden av x-aksen i stedet for 1 som er tiden i sekunder.

Dette kan endres ved å lage en fast mengde tidsstempler slik at x-aksen blir tiden som har passert over vinduet. Dette betyr at tiden vinduet inkluderer vil være i x-aksen. Tidsaksen blir altså satt over 2052 sampler. Det medfølger da at x-aksen alltid viser at vinduet inneholder data tatt over en tidsperiode på ett sekund. Ettersom alle vinduene blir initialisert med disse tidsstemplene vil det derfor gå utover modulariteten til BioPatRec om egne tidsstempler ikke blir vedlagt. Dette fordi det da må spesifiseres helt egne håndtering av plottene. Dette er det som er gjort i de tidligere versjonene av *Trigno Session*. Det er foretrukket at alle sensor implementasjoner brukte samme håndtering av plottene. Derfor ble denne implementasjonen gjennomført.

I *Recording Session* benyttes tidsstemplene videre til å forteller hvor lang tid som er igjen av en bevegelse. På denne måten kan brukeren bli informert om hvor mye tid som er igjen av en spesifikk bevegelse. På grunn av dette er det viktig å få inkludert tidsstempler, slik at DTWS brukere også får fullstendig utnyttet programvare.

Dette blir en stor endring. *updatePlots* utføres med mellomrom gitt av tiden der *ReadEMG* utføres når data er tilgjengelig. Dette fører til at det blir vanskelig å beregne tidsstempler ettersom dataen som er tilgjengelig kan variere.

Løsningen ble her å slå sammen de to funksjonene til *ReadAndPlotEMG*. Dette gir en løsning som er nærmere BioPatRec sin egen håndtering av signalopptak. Dette betyr at dataen nå vil bli plottet rett etter den er hentet. Dette gir en forutsigbar datamengde til hvert eneste plot, som gjør oppgaven å lage tidsstempler mye enklere. En bekymring her er at når *updatePlots* blir fullstendig bundet til datahentingen kan dette bety at plottene mye oftere vil oppdateres, noe som kan føre til en økning i ressursbruk. For å spare ressurser ble kravet for tilgjengelig data økt for at *ReadAndPlotEMG* skal bli kalt.

Som nevnt i kapittel 6.1 er det forskjellig frekvens på samplingen til akselerometer og SEMG data. Forholdet mellom SEMG og akselerasjon er 13.5:1. Siden data fra alle sensorene kommer i en rekke er det viktig at man passer på at dataen kommer i fullstendige sett. Det betyr at det siste settet med data må inneholde alle sensorene. Derfor må datamengden som kan hentes være en størrelse gitt av funksjonen under.

$$\text{Bytes} = \text{Sampler} * \text{Kanaler} * (\text{Bytes}/\text{Sample})$$

Hvert sample tar fire bytes, det er 16 kanaler ettersom det 16 sensorer og det må minst være 27 sampler. Dette er fordi 13.5 er et vanskelig antall sampler å hente. 27 sampler vil være synkronisert med to sampler fra akselerometeret. Fra tidligere *Trigno Sessions* har det vært satt 1728 bytes som må være tilgjengelig før dataen blir hentet. Dette er den minste størrelsen du kan ha

ettersom dette er akkurat 27 samples per sensor. For og forbedre ytelse ble denne størrelsen doblet til 3456, altså 54 sampler.

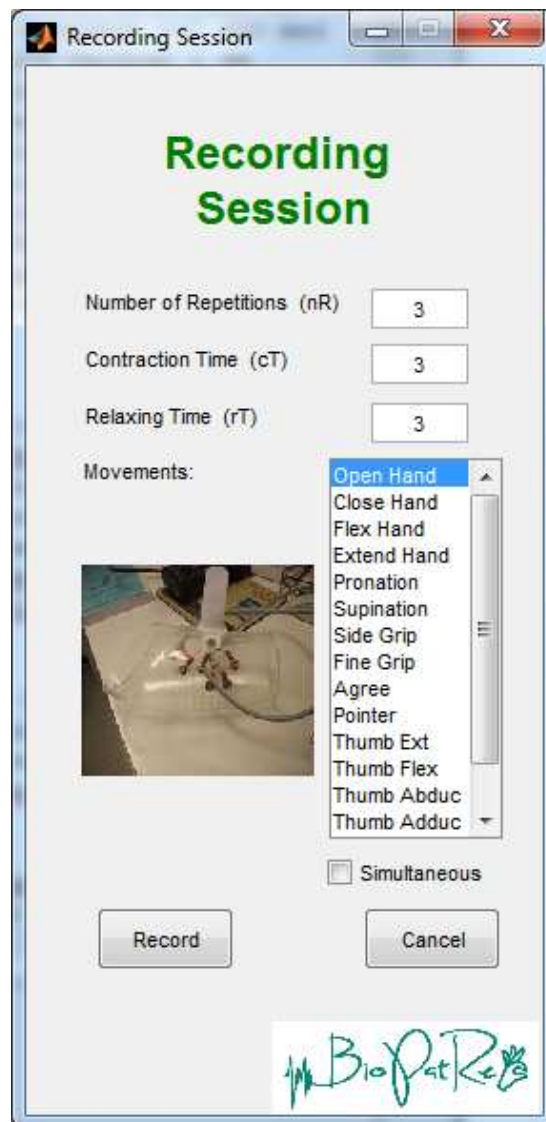
Nå som størrelsen på data som skal hentes er bestemt gjelder det og konstruere tidsstempler. Disse kan genereres ved å bruke størrelsen til den innkommende dataen og så generere en rekke som har samme lengde som den hentede dataen. Her er det altså snakk om lengden på dataen tilhørende en enkelt sensor. Det er derfor verdifullt å formatere hentet data slik at dataen fra de respektive sensorene ligger kolonnevis i en tabell. Verdien i rekken vil øke med en over frekvensen til sensorene, og den første verdien er satt av den siste verdien i de allerede genererte tidsstemplene. Når dette er gjort vil de nye tidsstemplene legges til de eksisterende. Tidsstemplene er lagret som *timeStamps* og ettersom lengden nå matcher antall sampler vil det nå være mulig å bruke disse tidsstemplene i *Recording Session*. Når signalopptaket er fullført i *Recordings* kan *timeStamps* brukes til og plote det fullstendige opptaket. Dette gjøres ved og plote *allData* mot tidsstemplene i *timeStamps*. Dette gjør at x-aksen vil gi den fullstendige tiden passert under opptaket. For å gjøre dette er det viktig at *timeStamps* er generert uten feil og at den har samme lengde som *allData*. Implementasjonen som tar med egne tidsstempler er vedlagt i Appendiks A som *TrignoSession4*

7.2 Recording Session

Når tidsstempler er implementert er det mulig å oppnå gode resultater i *Recording Session*. Det er derfor naturlig å ta koden videre til *Recording Session*. Det er nemlig denne modulen i BioPatRec som generer en *recSession*, det vil si strukturen som BioPatRec benytter for signalopptak. Hvis signalopptaket fra Trigno sensorene kan bli lagt inn i denne strukturen, vil det ikke være nødvendig med flere modifikasjoner senere i programmet. Dette er fordi *recSession* er en av standard strukturene i BioPatRec, og det er laget slik at dersom dataen er tilpasset denne, skal det ikke være kompatibilitetsproblemer med innholdet i resten av koden.

Trigno Session var bygget som en egen funksjon, som var separert fra koden til DAQ-enheten. I *Recording Session* er ikke dette nødvendig. Her kan brukeren selv spesifisere sensorer og andre alternativer. Deretter vil de bli sendt som variabler inne i funksjonen som utfører selve opptaket. Med disse variablene kan *Recording Session* utføre signalopptaket etter brukerens ønske.

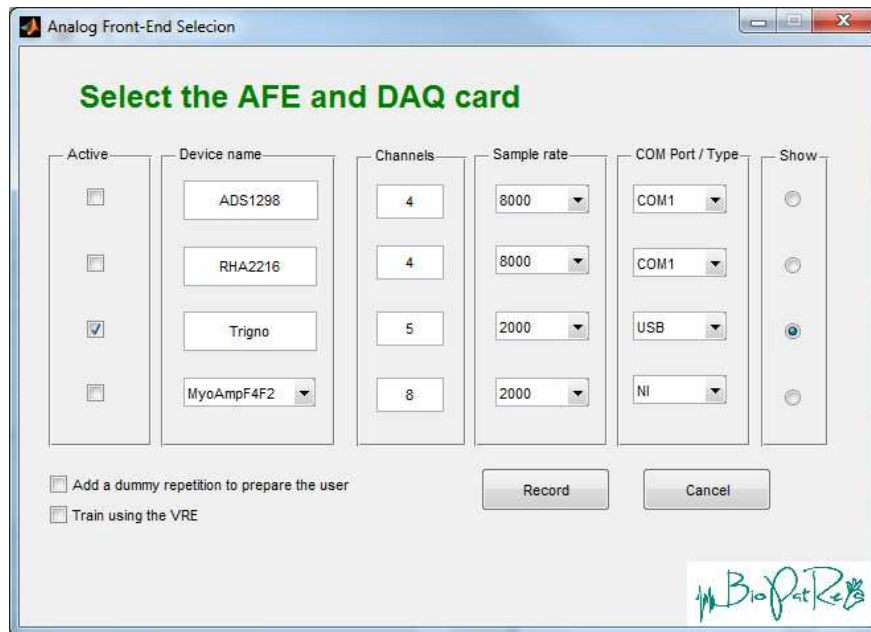
Recording Session kan velges fra hovedmenyen i figur 4. Deretter får brukeren velge hvilke type signalopptak som skal utføres. Brukeren får muligheten til å velge hvilke type bevegelser som skal utføres, lengden på muskelkontraksjonen, lengden på hviletilstanden og antall repetisjoner. Det er også en mulighet



Figur 12: Dette er hovedmenyen i Recording Session. Her velger brukeren hvilke

for å velge simultan bevegelser. Grensesnittet er vist i figur 12. Det er ikke gjort noen implementasjoner bak denne delen av koden, men den ble nevnte for å gi et bedre innblikk i hvordan *Recording Session* er strukturert.

Etter valget av signalopptak er det på tide å velge hvilke type sensorsystem som skal velges. I BioPatRec går da til *AFE Selection*. Der AFE står for *Analog front-end* eller en analog front. En analog front er det den analoge delen av en krets som kommer før analog til digital omformerer. I dette grensesnittet ble det laget et nytt alternativ for Trigno sensorer. Figuren 13 viser dette grensesnittet. For det kosmetiske er alle alternativene tilgjengelig som kan velges for de andre sensorene, men noen av de er ikke lenger valgbare ettersom det kun er et



Figur 13: Grensesnittet gir brukeren muligheten til å velge sensortype eller analog front.

par av alternativene som trenger å spesifiseres. Når brukeren går videre fra denne menyen starter opptaket automatisk, og alle valgene gjort i denne menyen og den forrige blir videresendt til funksjonen som styrer opptaket. Det er også verdt å nevne at det er mulig fra denne menyen å inkludere en repetisjonsvisning fra en protese, den kan da være koblet opp til programmet eller en virtuell hånd kan benyttes. Dette skal ikke forveksles med proteseguidet trening, siden brukeren ikke utfører bevegelsene sammen med protesen.

7.2.1 Første implementasjon: Overfør funksjonaliteten

Fokuset med denne implementasjonen er å få med funksjonalitet nok til at en gjennomføring av *Recording Session* er mulig. Det er derfor viktig at all funksjonalitet fra *Trigno Session* blir implementert her. Gjennom hele koden blir *if* sjekker brukt til og finne hvilken del av koden som skal kjøres. Det blir da sjekket hvilke sensor som er valgt av brukeren, og så blir koden for den riktige sensorer kjørt. Det er her altså mulig å kjøre flere sensorer samtidig hvis BioPatRec har mer enn et implementert sensorsystem tilgjengelig.

Ved første *if* sjekk blir samplingsfrekvensen og antall kanaler hentet ut. Dette må gjøres først slik at riktig håndtering av plottene kan legges til der de initialiseres. Etter dette kommer *if* sjekken som initialiserer sensorene. Her ble derfor Trigno sin initialiseringsfunksjonen *TrignoInit* lagt inn sammen med fo-

pen funksjonalitet som åpner koblingene til *tcpip* objektene. Etter denne kjøres øvelsene. Her blir det lagt inn pauser for å forberede brukeren på hva som er neste bevegelse. Etter denne pausen blir sensorene satt til å ta opp data med en ny *if* sjekk. Dette skjer slik det gjorde i *Trigno Session* med en ekstra pause som er lagt inn. Denne er lagt til for å unngå at opptaket starter før sensorene er klare. Det er en tidsforsinkelse fra serveren til DTWS får melding om å starte til den faktisk begynner å hente data. Mens sensorene henter data blir en øvelse kjørt. Først kjøres en gjennomgang av øvelsen og så en hvilepause. Dette repeteres like mange ganger som brukeren har valgt. Etter dette stoppes datahenting gjennom en ny *if* sjekk. Før neste øvelse startes lagres dataen som er hentet. Så blir neste øvelse kjørt. Dette fortsetter til alle øvelser er gjennomført. Når øvelsene er gjennomført og all data er lagret blir en *recSession* satt opp og lagret til fil, igjen skjer dette med en *if* sjekk siden det er ønskelig og lagre filene i forskjellige mapper etter hvilke sensorsystem som er benyttet.

Nå mangler en funksjon som henter data og plotter den. Denne blir mer kompleks en den som ble brukt i *Trigno Session*. Funksjonen *ReadAndPlotAdvancedEMG* er derfor delt inn i seksjoner, dette for at det skal være oversiktlig hva enkeltdelene av funksjonen gjør. Første del av koden henter ut variabler som trengs i resten av funksjonen. Neste del henter dataen som nå er tilgjengelig. Deretter blir funksjonaliteten for tidsstempler implementert. Etter dette skjer en strukturering av den samlede dataen. Den hentede dataen blir formatert før den legges inn i *allData*, og dataen som skal plottes blir formatert. Neste seksjon bruker tidsstemplene til å oppdatere hvor mye tid som er igjen av en øvelse slik at brukeren kan følge en statuslinje for når neste skritt i signalopptaket inntreffer. Etter dette kommer en implementasjon av frekvensspekteret til SEMG signalene, og så kommer plottingen av SEMG signalene, frekvensspekteret og oppdatering av signalstyrken.

7.2.2 Andre Implementasjon: Inkludering av akselerometer

Nå skal akselerometeropptaket inkluderes. Dette betyr at mange av sensor implementasjonene under *Recording Session* må modifiseres. Det er viktig å gå tilbake til *if* sjekkene og legge ved kode for akselerometerdataen. Dette blir ganske enkelt ettersom mye av implementasjonen er like SEMG implementasjonen. Et *tcpip* objekt må legges til for akselerometeret samt håndtering av dette. Tidligere har det akselerometer objektet ikke vært inkludert.

En funksjon for henting av data må også settes opp. Funksjonen som henter akselerometerdata vil kun bestå av en seksjon som henter data og en seksjon som strukturer dataen. Globale variabler utnyttes under datahenting for EMG og nye globale variabler settes derfor opp for henting av akselerometerdata. Denne funksjonen er altså mye simplere enn SEMG funksjonen, siden denne

dataen ikke blir implementert inn til grensesnittet. Grensesnittet er satt opp for EMG-data, og ved bruk av mange Trigno sensorer blir det vanskelig å finne en måte der all dataen kan bli vist. Det er i hovedsak viktigst at SEMG-dataen er synlig. Derfor blir fokuset å få lagt inn akselerometerdataen i *recSession*

En ny variable må introduseres til *recSession*. Dette er mulig å gjennomføre uten konsekvenser dersom ingen endringer blir gjort til den resterende strukturen. Sensordataen som foreløpig er inkludert i *recSession* er strukturert i en tredimensjonal matrise kalt *tdata*. Dimensjonene er data, kanaler og bevegelser. Akselerometeret i Trigno sensoren har akselerometerdata fra tre dimensjoner, og det må derfor legges til en ny datastruktur ettersom *tdata* ikke kan inkludere flere verdier for hvert sample. Derfor ble *aaData* konstruert. Den har en fire dimensjonal struktur der dimensjonene er data, akser, kanaler og bevegelser. Dette er et forslag til hvordan akselerometerdata generelt bør lagres i fremtidige versjoner av BioPatRec.

Andre små justeringer ble også utført i denne implementasjonen. Mengden data som må være tilgjengelig for å utføre datahentingen ble satt opp, dette ble gjort på grunn av høy ressursforbruk under kjøringen av *Recording Session*. Det ble også lagt til null utfylling av de fullstendige datasettene. Siden BioPatRec krever datasett av en spesifikk mengde som er basert på samplefrekvensen og tiden på øvelsen. Derfor måtte de få samplene som manglet i datasettene til Trigno sensorene null utfylles. Det er da snakk om små mengder data ettersom sensorene holder frekvensen de er satt til. De vil derimot aldri komme nøyaktig til den påkrevde summen, ettersom de må holdes synkront med akselerometerdataen.

TrignoSessionACC som er vedlagt i Appendiks A ble laget for å raskt teste om metoden for akselerometerdata fungerer. Dette gjøres ved at akselerometerdataen har en enkel plotting under *Recordings*.

Når koden beskrevet over fullføres er dette en fullstendig implementasjon av DTWS i *Recording Session*, og den modifiserte versjonen er å finne i Appendiks A.

Del III

Resultater

8 Bidrag og utførelse

Denne oppgaven har handlet om bruk av BioPatRec på NTNU sitt nevro-motoriske laboratoriet med spesielt fokus på signalopptak. De samlede bidragene som fremgår i oppgaven samt bruken av implementasjonen utgjør resultatene.

8.1 Norsk terminologi

Elektrisk protesestyring er et relativt nytt forsknings tema, og det er derfor en mangel på felles terminologi. I Fougner et al. (2012) er det foreslått en terminologi, men denne artikkelen beskriver engelsk terminologi. Under denne oppgaven var derfor norsk terminologi et tema. Det ble altså lagt et fokus på å oversette noen av den engelske terminologien til norsk. Derfor er det laget en tabell med norsk terminologi som er benyttet under denne oppgaven og under Alf Bertnum sin oppgave. Tabell 3 viser de engelske og de norske begrepene sammen med en rask beskrivelse og eksempler.

8.2 Funksjonsspesifikasjonen

Det har vært ønskelig å bytte ut programvaren som til nå er brukt på NTNU sitt nevro-motoriske laboratoriet. Programvaren var ganske omfattende å bruke og det manglet på mange måter modularitet. BioPatRec er en programvare som har fokus på disse manglene. Ettersom det er et prosjekt med åpen kildekode er modularitet veldig viktig. Det BioPatRec mangler er kompatibilitet for NTNU sitt utstyr. Derfor ble en vurdering gjort på bakgrunn av et sett med spesifikasjoner listet i tabell 2. Etter denne vurdering ble BioPatRec akseptert og modifisert for å ta hensyn til behovene NTNU har.

8.3 Bidrag i BioPatRec

I denne oppgaven ble det jobbet for en sensor implementasjon. Resultatet av dette bidraget er at BioPatRec ble introdusert til det nevro-motoriske laboratoriet. Implementasjonene som er gjort har medført at BioPatRec nå er anvend-

bart på laboratoriet til flere formål. Implementasjon er derimot ikke fullstendig og krever videre arbeid for at den skal fungere med alt utstyret. Men mye av grunnarbeidet er gjort. Denne oppgaven har fått *Recording Session* modulen og *Recordigs* modulen til og fungere med DTWS. Dette betyr at sensorene på det nevro-motoriske laboratoriet ved NTNU nå kan brukes til signalopptak i BioPatRec. Derfor kan signalopptak gjort ved NTNU nå for første gang benyttes av andre.

Under implementasjonen ble flere funksjoner laget og andre funksjoner ble modifisert. I tabell 4 er funksjonen som er benyttet i denne oppgaven listet. Alle funksjoner modifisert i denne oppgaven har blitt gjort på en slik måte at det ikke har noe konsekvens for opplevelsen av BioPatRec eller oppførselen til BioPatRec. Dette betyr at brukere som benyttet programvaren før implementasjonene gjort i denne oppgaven vil fortsatt ha muligheten til å bruke BioPatRec som vanlig, uten at de vil oppleve noen forskjeller eller problemer.

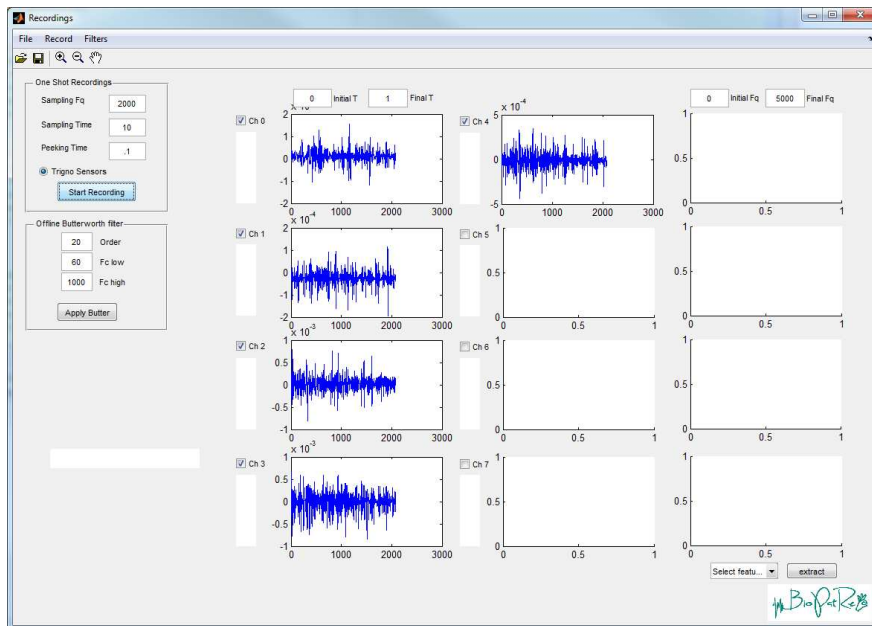
Det ble også foreslått en struktur for akselerometerdata i denne oppgaven. Det er kun støtte for EMG data i BioPatRec og en ny datastruktur for akselerometerdata måtte benyttes for å få lagret denne dataen til en *recSession*.

Implementasjonene gjort i denne oppgaven er også dokumentert i Berrum and Bertnum (n.d.). Denne dokumentasjonen er vedlagt i BioPatRec sin *wiki*. Her ligger en kortfattet forklaring på hva som er gjort. Slik at andre utviklere kan se på mer en bare koden for å få en innføring i dette arbeidet.

8.4 Trigno i bruk

Etter implementasjonen av DTWS har BioPatRec blitt testet for å se etter problemer og feil. Det er derfor lagt ved flere figurer som viser BioPatRec i bruk i både *Recordings* og *Recording Session*.

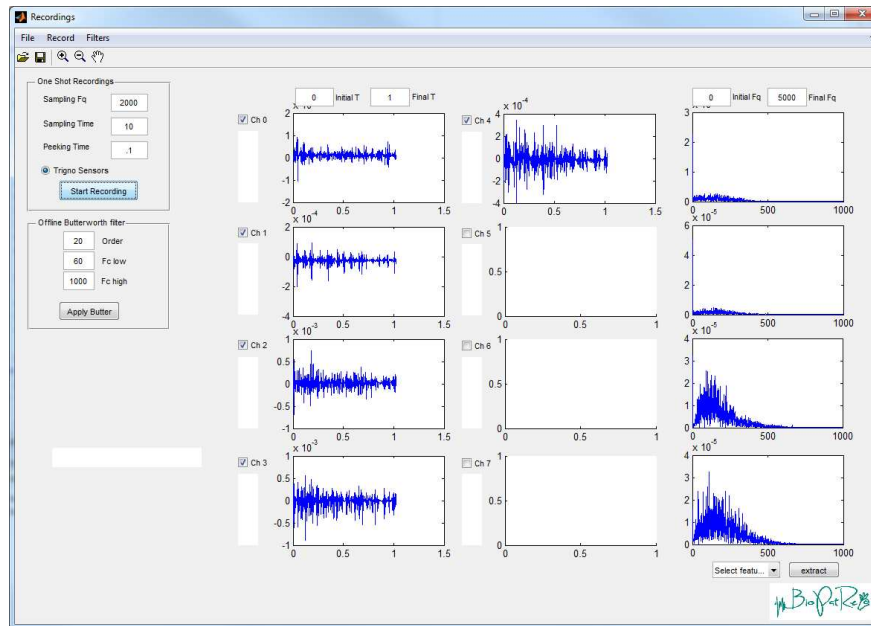
Figur 14 viser den aller første implementasjonen av *Trigno Session*. Her kan man legge merke til at kun det rene SEMG signalet er inkludert. Det er ikke tatt hensyn til noe annet. Denne implementasjonen var for å få en kode som lot seg kjøre i BioPatRec. Dette gav en bedre indikasjon på at systemene er kompatible. Man kan blant annet legge merke til at x-aksen ikke viser tidsrommet til vinduet men heller antall sampler. Figur 15 viser den siste implementasjon av *Trigno Session*. Her kan man finne frekvensspekteret til fire av inngangssignalene samt x-akse som nå illustrer tiden. Bruken av tidsstempler vil komme klarere frem i figur 16. Her kan man observere alle funksjonalitetene bli utnyttet. Alt fra tidslinjen og øvelsevisning til signalstyrken til plottet er inkludert. Etter at et signalopptak er gjennomført kan det siste øvelsen observeres i helhet, ettersom BioPatRec setter opp et plot for hele tidslinjen til den siste øvelsen. Figur 17



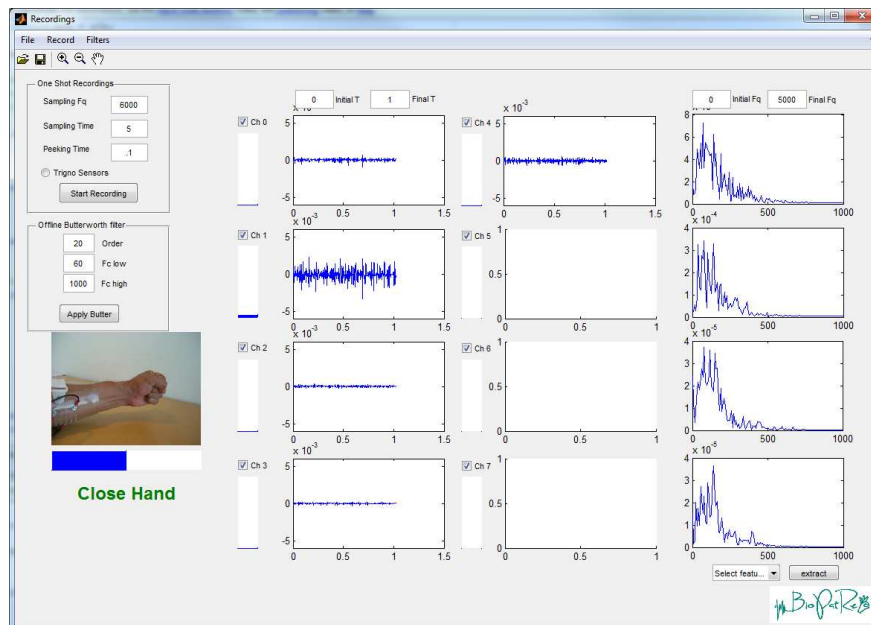
Figur 14: Fra *Recordings* i BioPatRec med den første implementasjonen av *Trigno Session*. Bare SEMG data er inkludert. x-aksen gir antall sampler. Vinduene har ikke en fast y-akse og y-aksen varierer derfor med signalstyrken som gis i volt.

viser dette.

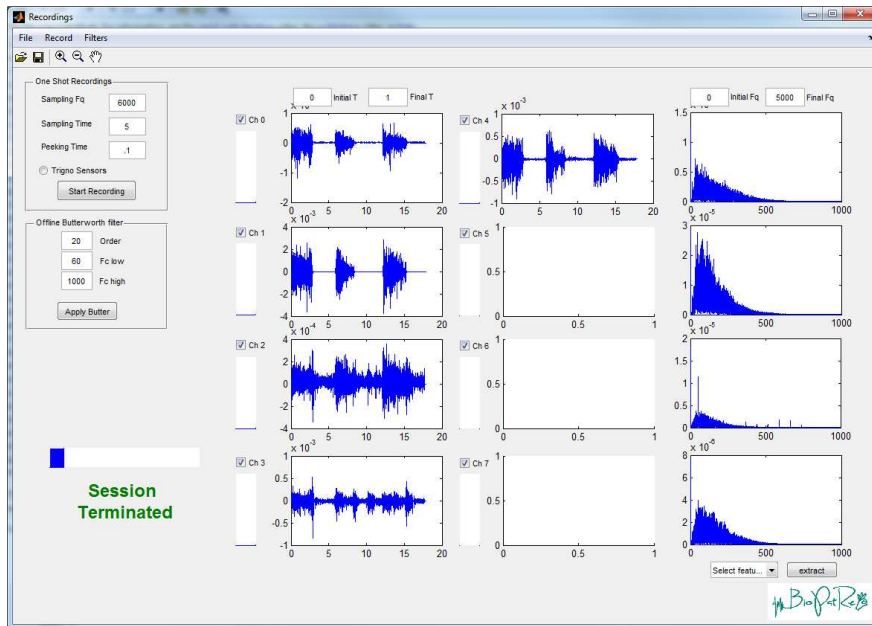
Til slutt er det tatt med et test plot av akselerometerdataen. En test versjon kalt *TrignoSessionACC* under *Recordings* ble satt opp for å ha en forsikring på at akselerometerdataen kom riktig inn. Dette blir da i tillegg til det faktumet av at dataen kom i riktig mengder og struktur i *Recording Session*. Dataen i denne testen ble nøye observert, og en bevegelse i y-retning under testen kan observeres i 18.



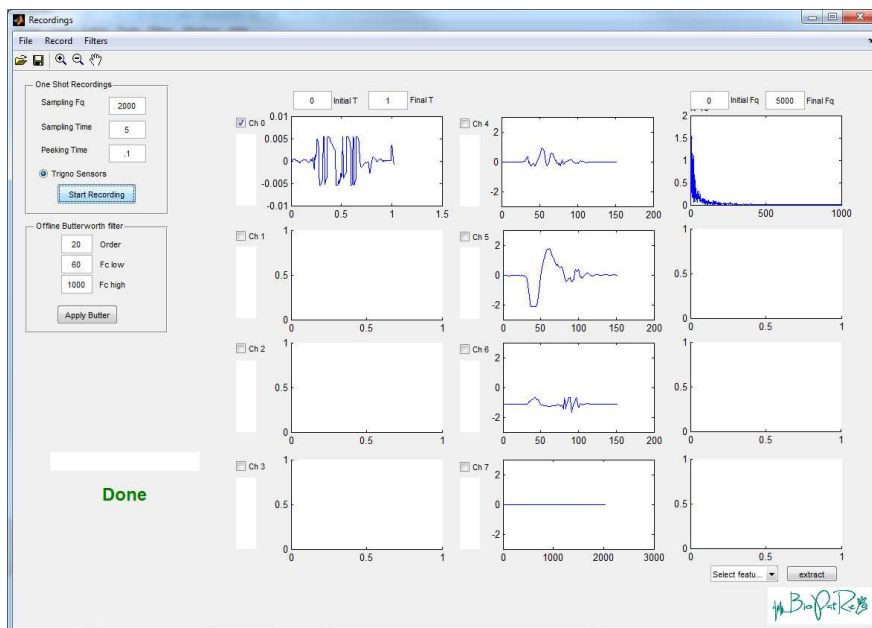
Figur 15: Fra *Recordings* i BioPatRec med den siste implementasjonen av *Trigno Session*. Frekvensspekteret er inkludert og x-aksen viser vinduets tidslengde i sekunder. Vinduene har ikke en fast y-akse og y-aksen varierer derfor med signalstyrken som gis i volt.



Figur 16: Fra *Recording Session* i BioPatRec. Signalopptaket er kommet til øvelsen for lukking av hånd. Her kan det observeres hvilke muskler som blir brukt i bevegelsen. y-aksen er nå fast i volt. Og x-aksen er tidslengden til vinduet i sekunder.



Figur 17: Når signalopptaket er over blir den siste øvelsen plottet. Her kan man tydelig se forskjell på kontraksjon og hvile. x-aksen er nå det samlede tiden til øvelsen i sekunder og y-aksen er gitt i volt.



Figur 18: Her kan er sensor observeres under en *Recordings* i BioPatRec. Denne testen av *Trigno Session* ble utviklet for å studere en sensor. Akselerometerdataen plottes i tre vinduer i midten. Her kan en tydelig akselerasjon i y-retning observeres. Dette viser at metoden for akselerometerdata fungerer etter ønsket oppførelse

Begrep = <i>original ord</i>	Beskrivelse	Eksempler
Intensjonsestimering = <i>Intent interpretation</i>	Tolkning av målesignaler fra en bruker for å estimere brukerens intensjon. Denne typen estimering kan gjerne deles opp i to deler, se figur 9.	PatRec-algoritmer
Ulineær støydemper = <i>Nonlinear flutter rejection filter</i>	Et filter for å dempe raske variasjoner i pådraget. Hensikten er å holde pådraget til pådragsorganet stabilt.	Figur ??
Motorfunksjon = <i>Motor function</i>	Den naturlige måten å betegne styringen av muskler og funksjoner i mennesker. Modellmessig kan det ses på som om å modellere én frihetsgrad eller flere. Ved én frihetsgrad består motorfunksjonen av én type bevegelse og dens antagonist-bevegelse, mens ved flere frihetsgrader fås en mer kompleks bevegelse.	Pronasjon/supinasjon av underarm <i>Kompleks bevegelse: Åpning/lukking av hånd</i>
Bevegelsesklasse = <i>Motion class</i>	Forskjellen mellom bevegelsesklasse og motorfunksjon, er at disse betegner en enkelt bevegelse, uavhengig av frihetsgrad. Sett i forhold til motorfunksjon, har disse bare én retning på pådraget eller null pådrag.	Pronasjon av underarm, åpning av hånd, hvile-tilstand
Brukertrening = <i>User training</i>	Trening av brukerens evne til å styre protesen. Dette vil være øvings- og erfaringsbasert trening.	
Justering av forsterkning og terskelverdier = <i>Gain and threshold adjustments</i>	Her justeres forsterkning, offset- og terskelverdier. Hensikten bak justeringene, er å få definerte operasjonssområder for hver bevegelsesklasse.	Seksjon ??
Styringsalgoritme = <i>Control algorithm</i>	Post-prosesserings-algoritme for å optimalisere utvelgelsen av pådrag til pådragsorganet. Målet er å redusere antallet feiltolkninger fra intensjonsestimeringen.	<i>Majority vote</i> og <i>Buffer output</i>
Beregning av signalegenskaper = <i>Feature extraction</i>	Selvforklarende navn, men se del 3 i figur 9 for plassering i protesesystemet.	<i>Zero-crossing</i> og <i>Waveform length</i>
Av/på-klassifisering = <i>Crisp classification</i>	Klassifiseringen som gir klassene av/påverdier. Det er mulig å bruke algoritmer for proporsjonalstyring også til av/på-klassifisering.	<i>KNN</i> og <i>LDA</i>

Begrep = <i>original ord</i>	Beskrivelse	Eksempler
Analog frontenhet = <i>Analog front-end device</i>	En analog front er den analoge delen av kretsen før analog til digital omformeren. En analog front enhet er en enhet som inneholder en krets med denne delen.	ADS1298 og RHA2216
Pådragsorgan = <i>Output device</i>	En generalisering for enhetene som blir påsatt signaler og pådrag fra intensjons-estimeringen.	Protese, VRE

Tabell 3: Tabell med oversikt over den foreslåtte norske terminologien. Det meste av terminologien er begreper definert i tabell 1 hos (Fougner et al., 2012, s. 665), men det gis en norsk beskrivelse i denne tabellen. Tabellen inkludere begreper brukt i denne og Alf Bertnum sin oppgave Bertnum (2013)

Funksjon	Bidrag	Versjoner
<i>TrignoSession</i>	Denne er skrevet for å kunne benytte DTWS under <i>Recordings</i> i BioPatRec	4
<i>ReadEMG</i>	Denne er skrevet for å hente data i en <i>TrignoSession</i> . Den benyttes derfor under <i>Recordings</i> i BioPatRec	2
<i>updatePlot</i>	Denne er skrevet for å plote data i en <i>TrignoSession</i> . Den benyttes derfor under <i>Recordings</i> i BioPatRec	2
<i>TrignoInit</i>	Denne er skrevet for å Initialisere DTWS i BioPatRec. Den brukes både i <i>Recordings</i> og <i>RecordingSession</i>	1
<i>GUI-Recordings</i>	Denne er modifisert for å kunne velge DTWS under <i>Recordings</i> i BioPatRec	1
<i>ReadAndPlotEMG</i>	Denne er skrevet for å plote og hente data i en <i>TrignoSession</i> . Den benyttes derfor under <i>Recordings</i> i BioPatRec	1
<i>GUI-AFE-selection</i>	Denne er modifisert for å kunne velge DTWS under <i>RecordingSession</i> i BioPatRec	1
<i>RecordingSession</i>	Denne er modifisert for å kunne benytte DTWS under <i>RecordingSession</i> i BioPatRec	2
<i>ReadAndPlotAdvancedEMG</i>	Denne er skrevet for å plote og hente data i en <i>RecordingSession</i> . Funksjonen oppdaterer alle funksjonaliteter som er relatert til EMG signaler under en <i>RecordingSession</i> i BioPatRec	1
<i>ReadACC</i>	Denne er skrevet for å plote akselerometerdata i en <i>RecordingSession</i> i BioPatRec	1
<i>TrignoSessionACC</i>	Denne er skrevet for å teste akselerometerdataen under <i>Recordings</i> i BioPatRec	1

Tabell 4: Her er bidragene gjort i BioPatRec. Hvis en funksjon er **skrevet** er det en ny funksjon for BioPatRec, er den **modifisert** har den eksistert i BioPatRec før denne oppgaven men den inneholder nå ny funksjonalitet

Del IV

Diskusjon

9 Diskusjon

BioPatRec har nå flere funksjonaliteter som støtter bruk ved det nevrologiske laboratoriet ved NTNU. Det er nå på tide å drøfte programvaren. For selv om oppgaven har resultert i en fungerende implementasjon har det vært problemer underveis. Noen av disse problemene var forutsett under programvarevurderingen i kapittel 6, men det utviklet seg også problemer med deler som var forventet å gå smertefritt. BioPatRec ble vurdert på forhånd men det er viktig å vurdere BioPatRec etter implementasjonene som er gjort.

9.1 Implementasjonshåndtering

Som nevnt var det et par problemer underveis i BioPatRec. Det ble brukt mye tid på å komme på løsninger for å håndtere disse på en fornuftig måte. Når man skriver programvare kommer det alltid til å oppstå problemer. Det er sjeldent man klarer å forutse alle utfall og konsekvenser av koden som blir skrevet og feil kan oppstå. Etersom koden blir mer kompleks øker muligheten for at noe problemer kan bli oversett.

Når det er snakk om programvare som er skrevet av andre kan det bli meget vanskelig å forutse problemer som kan oppstå ved egne modifikasjoner. Under denne oppgaven ble BioPatRec modifisert til bruk hos NTNU. BioPatRec er et stort prosjekt og det er derfor mye å sette seg inn i. Kompleksiteten i programvaren fører til at man trenger å sette opp en funksjonsspesifikasjon, for så å gjøre en vurdering av BioPatRec. Dette krever mye arbeid med tanke på at en grundig vurdering av BioPatRec kun er mulig etter at man har god forståelse av hvordan en slik programvare fungerer. I tillegg til dette skulle DTWS implementeres. DTWS har en liten sensortest som den eneste dokumenterte koden i Matlab. På grunn av dette og store forskjeller i hvordan BioPatRec og Delsys henter data fra sensorer tok implementasjonen en del lengre tid enn det som var forventet. Arbeidet i denne oppgavene endte til slutt med en fungerende implementasjon. Men for å drøfte fremtiden til BioPatRec etter implementasjonen, er det viktig å ta en nærmere kikk på problemene som oppstod underveis.

9.1.1 Tidsstempler

Tidsstemplene som BioPatRec bruker ble tatt opp i programvarevurderingen i kapittel 6.1. Det ble nevnt at dette kunne gi problemer i en implementasjon. Dette viste seg å stemme. Dette er først og fremst på grunn av kompatibilitetsproblemer mellom de forskjellige signalopptaksmetodene. Dette fordi Delsys har en implementasjon som benytter en funksjon som henter data når tilstrekkelig datamengde er tilgjengelig, og en funksjon som plotter data når en valgt tid har passert. Problemet med dette er at det er vanskelig og sette tidsstempelet for når mengden data som er hentet ikke er gitt i samme funksjon. Dette ble løst ved å legge funksjonaliteten for plotting inn i samme funksjon som henter data. Ettersom plottingen vil skje mye oftere er det naturlig å forvente en økning i ressursbruk. Derfor ble mengden data som hentes økt slik at funksjonen kalles med større mellomrom. Dette førte til at tidsstempler var mulig å benytte uten videre problemer. Ressursbruken økte på grunn av denne implementasjonen, men etter at datahenting skjedde med større mellomrom gikk ressursbruken til et akseptabelt nivå.

Den nye implementasjonen førte til en løsning mer lik BioPatRec. Dette kan gjøre at implementasjoner gjort under denne oppgaven kan lettere forstås av andre i fremtiden. Nå som en fungerende implementasjon er gjort blir det også lettere for andre å ta DTWS videre i koden. For eksempel burde det ikke være like vanskelig å bruke DTWS i BioPatRec til sanntids styring av protese. Dette krever en ny implementasjon, som kan baseres på denne oppgaven. Det er ikke nødvendig og starte helt fra begynnelsen.

9.1.2 Ressursbruk

Ressursbruk kan utvikle seg til å bli et stort problem. SDK serveren DTCU (Delsys Trigno Utility Control) krever alene mye ressurser og den må kjøres i bakgrunnen for at det skal være mulig å bruke DTWS. Det er derfor viktig at BioPatRec ikke bruker mer enn nødvendig data. Under *Recording Session* ble rommene mellom henting av data økt en gang til. Første gang var oppdateringen i seg selv veldig liten, men det er ikke nødvendig med data så ofte som funksjonaliteten i utgangspunktet gav og det gjorde bare at ressursbruken steg.

Det bør også nevnes at en feil av mellomlagringen av datasettene gjorde at programmet tok opp enda mer ressurser. *allData* viste seg å lagre mye mer data enn det som faktisk var tilgjengelig. Dette var mulig ettersom den lagret data fra tidligere datasamplere om igjen. Dette førte til ressursbruk som fikk hele systemet til falle etter for så å låse seg.

Feilen ble funnet og fikset. SDK serveren DTCU bruker opp hele 50 prosent

av systemressursene tilgjengelige på datamaskinene på det nevro-motoriske laboratoriet. Når BioPatRec i tillegg bruker rundt 20 prosent på et fullstendig signalopptak i *Recording Session* havner det totale forbruket på rett over 70 prosent. For å ha noe og sammenlikne med ble DTWS tatt med til en annen datamaskin for og monitorer ressursforbruket. Maskinen som da ble testet har kraftig maskinvare som for det meste var helt nytt i 2012. På denne maskinen ble ressursbruken sammenliknbar. Det viser seg at det totale ressursforbruket havnet på rundt 18 prosent, der 15 prosent var serveren og de resterende tre var BioPatRec.

Dette er en ekstrem forbedring, og ut ifra dette kan det være naturlig å gjøre en vurdering på om det bør investeres i nye datamaskiner ved det nevro-motoriske laboratoriet. Foreløpig klarer maskinene akkurat å henge med, men for videre bruk av BioPatRec kan nok nye datamaskiner være nødvendig. Et eksempel på dette kan være VRE delen av BioPatRec. Her blir en virtuell hånd simulert i henhold til styringssignale. Dette er meget ressurskrevende.

9.1.3 Ikke kompatible datasett

Kompatibiliteten i datasettene ble ikke utpekt som et problem i programvarevurderingen. At datasettene ville skape problemer var ikke forutsett. Problemet her ligger i hvordan BioPatRec foretrekker data. I BioPatRec spesifiserer du frekvensen på sensoren og hvor lenge bevegelsene du utfører vil ta. Ut i fra denne informasjonen lager BioPatRec en matrise med et eksakt antall sampler som den forventer. Om ikke dataen den får inn stemmer overens med beregningen vil ikke BioPatRec fungere. Dette er fordi alle bevegelsene ligger i samme matrise. Hvis samlemengden variere vil ikke dimensjonene samsvare i matrisen. Videre vil BioPatRec bruke lagret data i forhold til den satte frekvensen. Hvis ikke dette går opp vil BioPatRec ikke kunne fungere. Dette går igjen på at *recSession* må ha riktig data. Hvis alt lagres i ?? etter BioPatRec spesifikasjoner vil ingen nye feil kunne inntreffe.

BioPatRec selv har god støtte for å generere eksakt mengde med data. Dette er mye på grunn av ventefunksjonalitet fra DAQ-driver implementasjonen. Den gjør at BioPatRec kan forvente en eksakt mengde hver gang. Etter tiden som skulle brukes til en bevegelse blir BioPatRec holdt igjen en liten periode slik at all data blir tatt med. Om mengden data blir funnet før tiden er ute vil DAQ-driveren selv terminere.

DTWS har ikke samme funksjonalitet. Det er meget viktig at SEMG dataen synkroniserer med akselerometerdataen. Forholdet mellom dem er 13.5:1. Sensordata for SEMG må hentes tidligst hvert 27. sample, noe som gjør det vanskelig å synkronisere opp mot en ønsket verdi satt av BioPatRec. Dette fører

til matriselengder som ikke samsvarer. Det ble til slutt løst ved en null utfylling. Dette betyr at de manglende samplene blir satt til null. Dette er ikke en optimal løsning, men det ble den eneste som lot seg gjennomføre i BioPatRec. Null utfylling er ikke ønskelig å bruke under disse omstendighetene, men avgjørelsen kan forsvares ettersom SEMG og akselerometeret er klokket med samme verdi som BioPatRec bruker for å beregne antall sampler. Dette betyr at selv om samplene aldri vil treffe den ønskede verdien vil det være en minimal del av datasettet som må fylles. I tillegg er det alltid hviletilstanden som blir fylt og ikke selve bevegelsen.

Til ettertanke er det ingen problemer som ikke har latt seg løse med implementeringen av DTWS i BioPatRec. Visse løsninger er ikke helt optimale, og DTWS har en mye mer omfattende implementasjon enn for eksempel en DAQ-enhet. Dette kan observeres ved å se mengden kode som må til for å bruke DTWS i BioPatRec. Selv om en implementasjon er mulig, er ikke DTWS det mest kompatible sensorsystemet for BioPatRec. På tross av dette kan signalene som lagres fortsatt anvendes til videre protesestyring. Det hadde vært ønskelig å slippe SDK serveravhengigheten og videre sette antall sampler som kommer til å bli tatt opp med større nøyaktighet. Videre er det mulig å se på de analoge koblingene som er tilgjengelig i DTWS. Disse inngangen gjør det mulig å aksessere dataen via en DAQ-enhet og derfor kan man benytte DAQ-driver egenskapene i Matlab. Dette kommer da på bekostning av at det ikke lenger kan benyttes en USB tilkobling, som var et krav i funksjonsspesifikasjonen.

9.2 BioPatRec

Etter å ha studert og implementert funksjonalitet i BioPatRec kan programvaren drøftes. BioPatRec er laget modulært og dette kommer godt frem når koden studeres. De forskjellige datastrukturene er et godt fundament for modulariteten. Implementasjonen av nye algoritmer er veldig enkelt. Den største endringen som må utføres for å benytte egne algoritmer blir en omstrukturering av inngang- og utgangsvariablene.

Opgaven sitt hovedfokus er signalopptak under preprosesseringen. Her kommer modulariteten frem i *if* sjekken. Disse blir som en veiledning til hvor i koden funksjonalitet må inkluderes for å implementere sensorsystemer.

Det er benyttet pausefunksjonalitet for å holde igjen koden under de forskjellige opptakene. Pausefunksjonalitet er ikke alltid like bra å benytte. Slik kode kan føre til problemer som det ikke lenger kan håndteres. Et eksempel på dette skjedde i denne oppgaven. Problemene med ressursbruk førte til at koden aldri klarte å starte opp igjen etter den var satt til pause. For å utbedre ressursproblematikken ble det foretatt en effektivisering av koden, og problemer med

datahenting ble løst. Dersom koden ikke hadde låst seg på grunn av pausen, kunne koden terminert og det hadde mest sannsynlig tatt kortere tid å finne feilene som oppstod.

9.2.1 BioPatRec i forhold til ITK-programvaren

Prosjekt i ITK-programvaren dokumentert i (Berrum, 2012) og denne oppgaven har gitt innblikk i de forskjellige programvarene, spesielt på tanke på sensorer. Det er mange forskjeller i programvarene. Den største forskjellen er utviklingsverktøyet. ITK-programvaren bruker Labview og BioPatRec bruker Matlab. Her er det meste opp til hva brukeren foretrekker, men Matlab er ikke like omfattende og kan derfor fortsettes opp og tas i bruk enn det Labview kan. Grensesnittet i BioPatRec er også mer oversiktlig, men vinduene som er satt opp for plottig av signaler er ikke helt i samsvar med hva som hadde vært ønsket. Det hadde vært praktisk med 10 vinduer, da dette hadde muliggjort plottning akselerometerdata. Viktigste er riktignok muligheten til å observere SEMG signalet.

Når det kommer til modulariteten i programvarene foreligger det gode argumenter for å benytte BioPatRec. Inkludering av nye sensorer i BioPatRec viste seg å være lettere enn ved bruk av ITK-programvaren. Selv om ITK-programvaren også prøver å holde seg modulært var det fortsatt avhengigheter som gjorde det vanskelig med nye implementasjoner. Det var ikke uvanlig at feil slo igjennom helt andre steder i koden når små implementasjoner ble utført. Det bør nevnes at Labview kan gjøre det vanskeligere å skape høy modularitet. Programmeringsformen kan til tider gjøre det vanskelig å unngå avhengigheter, og ettersom programmet vokser i størrelse kan den grafiske programmeringen bli uoversiktlig med mindre utviklerne er meget forsiktig med hvordan de setter opp programvaremodulene. ITK-programvaren hadde også lite dokumentasjon, der BioPatRec har en egen *wiki*. Her har utviklere lagt ned gode beskrivelser av funksjonaliteter. Koden er også generelt kommentert. Dette var til stor hjelp under implementasjonen.

BioPatRec er en internasjonal programvare med åpen kildekode. Dette åpner en helt ny dør for NTNU. Signalopptak som er gjort på det nevro-motoriske laboratoriet i forbindelse med denne oppgaven kan nå benyttes av alle andre BioPatRec brukere. Før var laboratoriet begrenset til egen utvikling, men BioPatRec tilgjengelighet gjør at arbeid enklere kan deles i et forskningsmiljø. Dette betyr at arbeid andre har gjort ikke må gjøres om igjen hos NTNU.

10 Konklusjon

Det har vært ønsket et alternativ til ITK-programvaren som nå er i bruk ved det nevromotoriske laboratoriet ved NTNU. Oppgaven er et resultat av dette ønsket. BioPatRec er et av de bedre alternativene. Denne oppgaven omfatter en evaluering av BioPatRec i forhold til laboratoriets behov og arbeid med å tilpasse BioPatRec etter disse. I kapittel 6 ble det konkludert med at programvaren var egnet for bruk på det nevromotoriske laboratoriet til NTNU. Modifikasjonene gjort i denne oppgaven er sensorrelaterte, og resulterte i full funksjonalitet av *Recordings* og *Recording Session* modulene i BioPatRec, til tross for problemer underveis med implementasjonen. Sensorsystemet viste seg å fungere i BioPatRec, men har en litt mer omfattende implementasjon enn det andre sensorsystemer har i programvaren.

Etter å ha jobbet over lengre tid med både BioPatRec og ITK-programvaren er det tydelig at BioPatRec i sin helhet er meget godt egnet for bruk ved det nevromotoriske laboratoriet til NTNU. Det fremstår som et godt alternativ til ITK-programvaren. En videreutvikling og ytterligere tilpasninger av BioPatRec vil kunne gi muligheten for å erstatte ITK-programvaren. BioPatRec er generelt mer modulær og NTNU kan nyte godt av godene som kommer med programvaren som et internasjonalt åpen kildekode prosjekt.

11 Fremtidig Arbeid

Denne oppgaven har vært en introduksjon i BioPatRec, med spesielt fokus på aspekter til inngangssignaler programvaren. Mye har blitt gjort i løpet av denne oppgavene, men for at BioPatRec skal ha en fremtid på det nevromotoriske laboratoriet til NTNU er det fortsatt flere oppgaver som må utføres.

11.1 Videre implementasjoner

Her er det mye som kan nevnes, helt sentralt er det at BioPatRec blir modifisert slik at den får den samme funksjonaliteten som ITK-programvaren har i dag.

Til nå har det ikke blitt gjort noen implementasjon av MC protesen. Dette bør være en prioritet, da det vil gjøre det mulig å kjøre protesen i sanntid med DTWS gjennom *PatRec* modulen fra figur 4. MC protesen kjøres i dag gjennom en DAQ-enhet. Dette svært positivt med tanke på at det er god støtte for DAQ-enheter i BioPatRec.

Denne oppgaven, og oppgaven til Alf Bertnum kan videreføres til en fullstendig implementasjon av et proporsjonalt styringssystem. Men for at BioPatRec skal kunne utnytte en slik løsning godt, kreves det også her støtte for nye datasett som gir bedre resultater med proporsjonalstyring.

11.1.1 Delsys Trigno Wireless System

DTWS fungerer nå med modulene av BioPatRec som generer datastrukturer som inneholder signalopptak. For å få sensorene til å være kompatible med sanntids kjøring av proteser kreves en implementasjon av disse sensorene i tillegg til en implementasjon av MC protesen. BioPatRec funksjonen *Realtime-PatRec* kan kjøre en protese i sanntid med signalopptak ved å bruke treningssett som er generert på forhånd. Det bør derfor legges inn støtte for DTWS her. Arbeidet gjort i denne oppgaven kan benyttes til å skrive en funksjonalitet som tar sensordata fra DTWS og gjør denne tilgjengelig for sanntidsstyring.

I fremtiden kan det også vurderes å bruke DTWS med en DAQ-enhet istedet. Dette vil kunne føre til at man slipper SDK serveren til Delsys og at man kan løse problemene med ikke kompatible datasett på en bedre måte.

Ved USB bruk av DTWS i BioPatRec kan null utfyllingen kanskje byttes ut. Det bør her vurderes om man kan ta den siste dataen som er lagret og fortsette og fylle ut med den, eller man kan vente til SEMG datasettet har mer data enn forventet og så fjerne de aller siste samplene.

11.1.2 Recording Session

Fremover kan også *Recording Session* videreutvikles. Det kan i første omgang legges inn støtte slik at MC protesen kan brukes under *dummy repetition* til å vise bevegelsen som skal utføres av brukeren. Dette kan følges opp med fullstendig PGT funksjonalitet i *Recording Session*. PGT er en viktig del av ITK-programvaren og det er viktig at den kan videreføres til BioPatRec. Dette medfører at flere enn bare brukere ved NTNU kan dra nytte av en slik implementasjon. PGT er ikke veldig utbredt og derfor er det ikke så mange som driver forskning på det. Ved å inkludere PGT i BioPatRec kan flere som ellers ikke hadde hatt muligheten, teste og videreutvikle PGT.

11.1.3 Kraft sensorer

Det bør også vurderes om kraftsensorer skal implementeres i BioPatRec. Får å gjøre dette kan dokumentasjonen til denne oppgaven benyttes. Det bør skrives en funksjon som implementerer kraftmålinger i *Recordings* og det bør skrives en funksjon som implementerer kraftmålinger i *Recording Session*. Dette vil bli en mindre omfattende oppgave siden kraftsensoren på det nevrologiske laboratoriet allerede bruker en DAQ-enhet for og kommunisere med en datamaskin. *recSession* vil da også trenge en oppdatering som inkluderer kraftdata.

11.2 Norsk terminologi

Ettersom arbeidet fortsetter på laboratoriet med BioPatRec er det mulig å utvide terminologien. Den norske terminologien utarbeidet i denne oppgaven er relatert til arbeidet. Når nye forskning og implementasjon blir utført vil det komme godt med og videreutvikle den norske terminologien til begreper som brukes innen andre temaer under proteseforskning.

11.3 Laboratoriet

Til slutt er det greit å ta med at regnekraften til datamaskinene på laboratoriet ble et problem under deler av implementasjonen. Det er ikke nødvendigvis det at programvaren tar opp for mye kraft, ettersom systemet ble testet på en annen datamaskin med alarmerende forbedringer i ressursbruk.

Når man driver med operasjoner i sanntid vil alltid et fokus være å holde ressursbruken i programmet nede, men på et tidspunkt må man også passe på at utstyret er kraftig nok. I dette tilfellet bør datamaskinene byttes ettersom

resultater på andre datamaskiner viser at BioPatRec i seg selv ikke bruker store mengder ressurser etter dagens standard. Fremtidige prosjekter vil være godt tjent med et oppgradert sett med datamaskiner.

Referanser

Berrum, K. (2012), Inclusion of a multimodal sensor unit in a prosthesis control system, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Berrum, K. and Bertnum, A. (n.d.), 'Dokumentasjon av arbeid'.

URL: <https://code.google.com/p/biopatrec/wiki/NTNU>

Bertnum, A. B. (2013), Biopatrec som forskningsplattform med fokus på intensjonsestimering og systemtrening, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Delsys (n.d.), 'Delsys inc'.

URL: <http://www.delsys.com/>

Duda, R. O., Hart, P. E. and Stork, D. G. (2001), *Pattern Classification*, 2 edn, John Wiley & Sons.

Fougner, A., Stavadahl, Ø. and Kyberd, P., J. (utgis senere), 'System training and assessment in simultaneous proportional myoelectric prosthesis control', *IEEE Transactions on biomedical Engineering* .

Fougner, A., Stavadahl, O., Kyberd, P., Losier, Y. and Parker, P. (2012), 'Control of upper limb prostheses: Terminology and proportional myoelectric control: A review', *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* **20**(5), 663–677.

Linnerud, Å. S. (2012), Lab-oppsett for proteseforskning, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

MC (n.d.), 'Motion control inc'.

URL: <http://www.utaharm.com/>

Ortiz-Catalan, M., Brånemark, R. and Håkansson, B. (2013), 'Biopatrec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms', *Source code for biology and medicine* **8**, 11.

Ortiz-Catalan, M. J. (n.d.), 'Biopatrec'.

URL: <http://code.google.com/p/biopatrec>

Sæther, M. (2008), Practical artifact cancellation for myoelectric prosthesis control, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Suleng, T. (2012), Design av multimodal inputsensor for armproteser, Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Appendiks A CD

Her er en kort forklaring av hva som ligger på CD.

A.1 Rapporten Masteroppgaven er her vedlagt som en enkel pdf.

A.2 Kildekode til rapporten Den fullstendige kildekoden til rapporten er vedlagt

A.3 Referanser Her ligger mange av referansene i pdf format.

A.4 BioPatRec En fullstendig versjon av BioPatRec ligger vedlagt.

A.5 Arbeidet i denne oppgaven En mappe med de mest relevante filene fra BioPatRec er vedlagt. Her ligger bare filer denne oppgaven har brukt eller skrevet.

A.6 Matlab eksempler Her ligger Matlab filer som ble brukt som veiledning under oppgaven.

A.7 Figurer Alle figurer brukt i rapporten som .eps, .pdf, .jpg og .png.