

Stemmestyring av armortose for ALS-pasient

Anniken Olsen

Master i kybernetikk og robotikk

Innlevert: mai 2018

Hovedveileder: Øyvind Stavdahl, ITK

Medveileder: Terje Kristoffer Lien, MTP

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

Forord

Denne masteroppgaven utgjør siste del av min 5-årige mastergrad i kybernetikk og robotikk med hovedprofil i biomedisinsk kybernetikk. Oppgaven er skrevet etter henvendelse fra professor Terje K. Lien, som ønsket en styringsenhet til en motorisert armortose han bygger i samarbeid med St. Olavs Hospital i Trondheim. Ortosen lages til en pasient som er rammet av sykdommen Amyotrofisk lateral sklerose.

I denne masteroppgaven er det utviklet en stemmestyringsenhet for den motoriserte ortosen, oppbygningen av denne er beskrevet i underkapittel 4.2. Undertegnede valgte ut implementeringsteknologi og dataprogrammer benyttet til å utarbeide styringsenheten, utviklet stemmestyringsprogrammet og arrangerte testing av systemet. Grensesnittet mellom ortosen og styringsenheten, det vil si formen på styringssignalet og signaloverføringen, ble avgjort gjennom samarbeid og kommunikasjon med oppdragsgiver. Lien anskaffet de nødvendige komponentene til styringsenheten, gjorde fysiske justeringer på enheten som viste seg nødvendig underveis og satte sammen styringsenheten. Utgiftene for komponentene til styringsenheten og lisens til dataprogrammet QuickT2SI, ble dekket med støtte fra Helse Midt-Norge til ortoseprosjektet.

Jeg vil takke mine veiledere Øyvind Stavdahl og Terje K. Lien for nyttige innspill og gode veiledningssamtaler. En ekstra takk til Stavdahl som sa seg villig til å være min veileder på tross av han for øyeblikket er bosatt på New Zealand. Skype har gitt mulighet for mange gode veiledningssamtaler med begge veiledere. I tillegg har jeg hatt jevnlig møter med veileder og oppdragsgiver, Lien. Jeg har satt stor pris på all inspirasjon og gode råd dere har gitt meg. Jeg er takknemlig for å ha fått jobbe med en så lærerik og givende masteroppgave.

Jeg vil også takke familie og Simen for god støtte og motivasjon. En stor takk rettes også til de som testet det ferdige stemmestyringssystemet.

Anniken Olsen
Trondheim, 16. Mai 2018

Abstract

This thesis is written as a continuation of the paper “Control principles for orthosis for ALS patient”. As concluded in the paper, the work in this thesis will be developing a voice control unit for controlling the movement of a powered orthotic device. The orthotic device is custom made for a patient with Amyotrophic Lateral Sclerosis (ALS). It consists of a support structure for the arm and two motors providing movement in the patient’s shoulder and elbow joint. The voice control unit listens for voice commands which leads to a control signal sent to the motors on the orthotic device. The control unit is built with an Arduino Uno card with ATmega168 microcontroller combined with a EasyVR voice recognition shield. The unit is connected to the orthotic device with one cord, the cord is both supplying the control unit with power and transferring the control signals to the actuators. The voice control unit send control signals for the speed and the direction of movement for the two motors. The production of the unit was finished within this thesis, and the control unit is currently in use by the patient. The unit is tested by 5 individuals, including the patient, to recognize commands correctly in 89,61% of cases while testing the movement control modes.

Sammen drag

Denne masteroppgaven skrives som en videreføring av arbeidet i prosjektoppgaven “Styringsprinsipper for ortose til ALS-pasient”. Som konkludert i prosjektoppgaven, vil det i denne masteroppgaven utvikles en stemmestyringsenhet for en motorisert armortose til en pasient med Amyotrofisk lateral sklerose (ALS). Armortosen er spesialtilpasset for pasienten, den består av en støtteskinne som stabiliserer armen og to motorer som gir bevegelse i pasientens skulder og albue. Stemmestyringsenheten er bygget av et Arduino Uno kretskort med ATmega168 mikrokontroller, som er slått sammen med en kompatibel EasyVR-talegjenkjenningsmodull. Stemmestyringsenheten lytter etter kommandoer, som fører til styringssignal for motorene. Enheten er koblet til ortosen med én ledning, denne forsyner stemmestyringsenheten med strøm og sender styringssignaler til ortosens motorer. Styringssignalene kontrollerer hastighet og retning av pådraget på ortosens to motorer. Ved innlevering av masteroppgaven var stemmestyringsenheten tatt i bruk av pasienten. Stemmestyringsenheten er testet av 5 personer, inkludert pasienten. Resultatene av testingen viser at stemmestyringsenheten gjenkjenner en avgitt kommando korrekt i 89,61% av tilfellene under testing av stemmestyringsenhetes bevegelsesmoduser.



Masteroppgave

Studentens navn: Anniken Olsen

Fag: Teknisk kybernetikk

Tittel (norsk): Stemmestyring av armortose for ALS-pasient

Tittel (English): Voice Control of Upper-Limb Orthosis for ALS Patient

Beskrivelse:

Det foreligger et langsiktig mål om å bygge en prototyp av en armortose for en person med motornevronsykdom (ALS) med svikt i armfunksjon. ALS er en tilstand der de motoriske funksjonene i ulike deler av kroppen svekkes i ulikt tempo. Det betyr at enkelte funksjoner kan brukes til styring av ortosens motorer i en tid etter at andre funksjoner begynner å svikte.

Et tidligere studentprosjekt har utredet alternative styringsstrategier. I denne oppgaven skal du implementere og evaluere én av disse strategiene.

1. Gi en kort oppsummering og revurdering av resultatene fra det foregående prosjektet i lys av dagens situasjon og fremtidige behov for den aktuelle brukeren.
2. Med utgangspunkt i at stemmestyring synes å være den mest aktuelle styringsmetoden, gjennomfør en detaljert analyse av utfordringer og praktiske løsninger for stemmestyring av motoriserte armortoser.
3. Gi en kortfattet vurdering av hvordan stemmestyring fungerer i forhold til tidligere prøvde styringsmetoder for armortosen som er grunnlaget for oppgaven.
4. Foreta et begrunnet valg av implementeringsteknologi for en praktisk implementering av en stemmestyring for den foreliggende ortosen, der du tar hensyn til foreliggende komponenter, annen tilgjengelig teknologi og prosjektets tidsramme.
5. I den grad tiden tillater det skal det gjennomføres praktisk utprøving av stemmestyringen, samt en vurdering av denne teknologiens egnethet for motoriske hjelpemidler for ALS-pasienter generelt.

Veileder(e): I. amanuensis Øyvind Stavdahl, Institutt for teknisk kybernetikk, NTNU
Prof. Emeritus Terje K. Lien, Institutt for maskinteknikk og produksjon, NTNU

Trondheim/Auckland, 12.10.2017

Øyvind Stavdahl
Faglærer

Innhold

| | | |
|----------|--|-----------|
| 1 | Introduksjon | 1 |
| 1.1 | Oppgavens relevans | 1 |
| 1.2 | Kartlegging og avgrensing av oppgaven | 3 |
| 1.3 | Arbeidsmetodikk | 3 |
| 2 | Videreføring av prosjektoppgave | 5 |
| 2.1 | Kroppens signalsystem og ALS | 5 |
| 2.2 | Revurdering av prosjektoppgavens konklusjon | 6 |
| 3 | Den motoriserte ortosen | 9 |
| 3.1 | Utvikling av den motoriserte ortosen | 9 |
| 3.2 | Den motoriserte ortosens komponenter | 10 |
| 3.2.1 | Motorstyreren | 12 |
| 3.2.2 | Ortosens aktuatorer | 12 |
| 3.3 | Tidligere styringsmetoder for den motoriserte ortsen | 13 |
| 3.3.1 | Joystick | 13 |
| 3.3.2 | Trykksensorer | 14 |
| 3.3.3 | Fotstyring | 15 |
| 4 | Utvikling av stemmestyringsystemet | 17 |
| 4.1 | Produktspesifikasjon | 18 |

| | | |
|----------|---|-----------|
| 4.2 | Valg av implementeringsteknologi | 19 |
| 4.2.1 | Talegjenkjenningmodulen EasyVR | 20 |
| 4.2.2 | Talegjenkjenningsprogrammet på EasyVR-modulen | 21 |
| 4.3 | Arkitekturdesign ved analyse av utfordringer for stemmestyring av armortose | 22 |
| 4.3.1 | Stemmestyringsprogrammets moduser | 25 |
| 4.3.2 | Brukereksempel av norskmodus | 27 |
| 4.3.3 | Planlegging av system testing | 30 |
| 4.4 | Moduldesign | 30 |
| 4.4.1 | Stemmestyringsprogrammets moduler | 30 |
| 4.4.2 | Lydklipp og talegjenkjenningskommandoer | 31 |
| 4.5 | Modulspesifikasjon og koding | 32 |
| 5 | Grensesnitt mellom ortose og stemmestyringsenhet | 37 |
| 5.1 | Styringssignal sendt ved pulsbreddemodulasjon | 37 |
| 5.2 | Filtrering av styringssignalet | 38 |
| 5.2.1 | Valg av filterkonstanter | 38 |
| 5.2.2 | Det ferdige filteret | 42 |
| 6 | Stemmestyring i praksis | 45 |
| 6.1 | Testing | 46 |
| 6.1.1 | Testing av engelskmodus med ulik sensitivitetsinnstilling | 51 |
| 6.2 | Tilbakemeldinger med utviklers kommentarer | 52 |
| 6.2.1 | Tester 1 | 53 |
| 6.2.2 | Tester 2 | 53 |
| 6.2.3 | Tester 3 | 53 |
| 6.2.4 | Tester 4 | 53 |

| | | |
|----------|--|-----------|
| 6.2.5 | Tester: Pasienten | 54 |
| 6.3 | Stemmestyring i forhold til tidligere styringssystemer | 55 |
| 7 | Diskusjon og konklusjon | 57 |
| 7.1 | Vurdering av egnethet | 58 |
| 7.2 | Fremtidig arbeid | 59 |
| A | Bruksanvisning for stemmestyringsenhet til armortose | 61 |
| A.1 | Om stemmestyringsenheten | 61 |
| A.2 | Stemmestyringsenhetens innganger og utganger | 62 |
| A.3 | Stemmestyringsenheten i bruk | 62 |
| A.4 | Engelskmodus | 64 |
| A.5 | Norskmodus | 64 |
| A.6 | Programmeringsmodus | 65 |
| A.7 | Talegjenkjenningssensitivitet | 66 |
| A.8 | Datatilkobling til EasyVR commander | 66 |
| | Referanser | 74 |

Figurer

| | | |
|------|---|----|
| 2.1 | Blokkdiagram som viser veien fra motorisk intensjon til motorisk aksjon hos en pasient med ALS og motorisert ortose | 6 |
| 3.1 | Den motoriserte ortosen med fotspaker til styring og batteripakke | 10 |
| 3.2 | Konstruksjonsskisse av den motoriserte ortosen med overkroppssele | 11 |
| 3.3 | Delsystemene som utformer den stemmestyrte ortosen. | 11 |
| 3.4 | Systemskisse over ortosens motorstyrer og aktuatorer | 12 |
| 4.1 | Fremstilling av V-modellen. | 18 |
| 4.2 | EasyVR modul slått sammen med Arduino Uno, og komponentene innkapslet i en boks | 20 |
| 4.3 | Delene som utformer den stemmestyrte ortosen. | 21 |
| 4.4 | Oversikt over stemmestyringsprogrammets overordnede arkitektur | 26 |
| 4.5 | Oversikt over kommandoene mellom norskmodusens ulike tilstander | 28 |
| 4.6 | Brukereksempel av stemmestyringsenhetens ordinære norskmodus | 29 |
| 4.7 | Brukereksempel av stemmestyringsenhetens norske hurtigmodus | 29 |
| 4.8 | Pseudokode som forklarer innholdet i filen SetupEasyVR.ino. | 33 |
| 4.9 | Kodeutdrag fra filen StateMachine.ino. | 34 |
| 4.10 | Forenklet kode fra funksjonen som endrer talegjenkjenningssensitivitet. | 35 |
| 4.11 | Forenklet kode fra funksjonen som trener angitt kommando i commandTraining.ino. | 36 |
| 5.1 | Skisse av RC-filteret som glatter PWM-styringssignalet | 39 |

| | | |
|------|--|----|
| 5.2 | Styringssignal målt med oscilloskop | 42 |
| 5.3 | Forstørret styringssignal målt med oscilloskop | 43 |
| 6.1 | Bilde av den ferdige motoriserte ortosen i bruk med stemmestyringen. | 45 |
| 6.2 | Samlet gjennomsnitt fra testresultatene | 47 |
| 6.3 | Testresultater for stemmestyringsenhetens engelskmodus | 48 |
| 6.4 | Testresultater av modus for å justere talegjenkjenningssensitivitet. | 49 |
| 6.5 | Testresultater av programmeringsmodus. | 49 |
| 6.6 | Testresultater for stemmestyringsenhetens norskmodus | 50 |
| 6.7 | Testresultater av engelskmodus med samme testperson og ulik talegjenkjenningssensitivitet. | 52 |
| A.1 | Stemmestyringsenheten. | 61 |
| A.2 | Oversikt over modusene i stemmestyringsprogrammet | 63 |
| A.3 | Oversikt over kommandoene mellom engelskmodusens ulike tilstander. | 67 |
| A.4 | Oversikt over kommandoene mellom norskmodusens ulike tilstander. | 68 |
| A.5 | Fremgangen for trening av norske kommandoer i programmeringsmodusen. | 69 |
| A.6 | Alle de norske kommandoene med tilhørende gruppe- og indeksnummer. | 70 |
| A.7 | Oversikt over modusen hvor talegjenkjenningssensitiviteten stilles. | 71 |
| A.8 | Stemmestyringsenheten åpnet | 72 |
| A.9 | Skjerm bilde fra dataprogrammet EasyVR commander | 72 |
| A.10 | Skjerm bilde fra dataprogrammet QuickSynthesis | 73 |
| A.11 | Skjerm bilde fra innlasting av lydfiler og engelske ordsett med EasyVR commander | 73 |

Forkortelser og oversettelser

ALS Amyotrofisk Lateral Sklerose

EMG Elektromyografi

EEG Elektroencefalografi

FAT Factory acceptance test

SAT Site acceptance test

AD Analog til digital

Sampling Tasting

Aliasing Nedfolding

SI Speaker independent, ikke personavhengig

SD Speaker dependent, personavhengig

TP True positive, sann positiv

FP False positive, falskt positiv

FN False negative, falskt negativ

1 | Introduksjon

Denne masteroppgaven skrives som en oppfølging av min prosjektoppgave, “Styringsprinsipper for ortose til ALS-pasient”. Prosjektoppgaven ga en kartlegging av den forestående styringsutfordringen med å lage et styringssystem for en motorisert armortose som bygges til en pasient med Amyotrofisk lateral sklerose (ALS). Oppgavens fokus var å lære om sykdommen og vurdere alternative styringsprinsipper. Nå skal det utvikles et styringssystem til ortosen basert på det styringsprinsippet som er utredet som best egnet. Jeg antar leseren av denne masteroppgaven kjenner til grunnleggende terminologi innen kybernetikk og menneskelig anatomi, og vil derfor ikke utdype vanlige faguttrykk.

Den motoriserte armortosen bygges av Terje K. Lien i samarbeid med ALS-teamet på St. Olavs hospital. Etter oppdrag fra Lien skal det i denne masteroppgaven utvikles et styringssystem basert på informasjonen samlet i prosjektoppgaven. I prosjektoppgaven ble stemmestyring utredet som det mest gunstige styringsprinsippet for pasienten ortosen utvikles til. Etter avtale med pasienten vil vedkommende bli anonymisert ved å omtales som “pasienten”.

1.1 Oppgavens relevans

ALS er en medisinsk utfordring, foreløpig finnes ingen medisin for å stanse denne neurodegenerative sykdommen.[1] Det finnes lite funksjonelle hjelpemidler tilgjengelig for å bistå ALS-pasienter sin bevegelse, derfor undersøkte pasienten mulighetene for å få utviklet et hjelpemiddel for å øke bevegelse i armene. En rekke hjelpemidler forutsetter håndfunksjon. Pasienten, som mange andre med neurodegenerative sykdommer, kan reise seg og gå, men har utfordringer med å håndtere brytere, dørhåndtak og lignende.

Mulighet for bevegelse gir selvstendighet og frihet, dette er viktig for livskvalitet og trivsel. En ortose vil ikke kunne erstatte den tapte funksjonen, men det kan være en viktig hjelp for å holde livskvaliteten oppe. Teknologien for å utvikle en ortose med tilrettelagt styringssystem er tilgjengelig, og dette vil være til stor nytte for pasienten i hverdagen. Med tilpassede hjelpemidler kan pasienter med ALS få økt livskvalitet og funksjonalitet i flere år etter sykdommens utbrudd. Å opprettholde bevegelse er blant annet viktig for sirkulasjonssystemet, ledd og det kan hjelpe å motvirke smerter og stivhet. Med motorisk assistanse fra en ortose vil behovet for assistert bevegelsestrening og assistanse i hverdagslige gjøremål reduseres, dette vil både være til fordel for brukerens selvstendighet og det vil være gunstig for samfunnet å bevare en person sin helse og selvhjulpenhet så lenge som mulig.

Den motoriserte ortosen har fått sitt navn basert på at ortosen i første omgang var en ortopedisk støtteskinne for underarmen. Ettersom ortosen ble utviklet med motorer plassert ved armens albue og skulderledd og er festet til kroppen, kan det argumenteres for at enheten kan kategoriseres som et overarms-eksoskjelett. Forskjellen mellom dette og en protese, er at en protese benyttes som erstatning for en manglende kroppsdel. Eksoskjeletter og den motoriserte ortosen benyttes for å støtte og assistere bevegelse av en kroppsdel. Forskningen utført på eksoskjeletter er omfattende, en slik ytre støttestruktur på kroppen kan benyttes både å assistere og styrke bevegelse. Det mangler imidlertid tilgjengelighet på dette for vanlige norske brukere. Årsaken til dette kan komme av at en del av disse eksoskjelettene fortsatt er i forskningsfasen, de er svært kostbare og ofte store og tunge.[2] [3] For pasienten var det ønskelig med noe lett og enkelt som dekket dagens behov og var tilgjengelig raskt.

I samtale med to norske aktører som utvikler hjelpemidler for mennesker med funksjonshemming, kom det frem at de ikke tilbyr eller har kjennskap til hjelpemidler for stemmestyring av assistert fysisk bevegelse. [4] [5]

Utviklingen innen stemmestyring har vært økende de siste årene og stemmestyring kan synes å være et fornuftig satsingsområde. Store aktører har tatt i bruk talegjenkjenning for å utvikle produkter med stemmestyring til å forenkle hverdagen, slik som Apple home kit og Google home. Disse løsningene kan gi stemmestyring av en rekke kompatible produkter. I denne oppgaven vil stemmestyring brukes innen rehabiliteringsteknologi. ALS pasienter sitt hjelpebehov kan være stort og når personer skal bistå er det kostnads og ressurskrevende. Hjelpemidler som kan styres uten muskelkraft, vil kunne gjøre denne gruppen mye mer selvstendig. Denne pasientgruppen kan ha store fysiske hjelpebehov, men de er intellektuelt intakt. Hjelpemidler som fører til økt

bevegelse vil kunne gi helsefordeler som bedret sirkulasjon og respirasjon, samt mestring og velvære.

1.2 Kartlegging og avgrensing av oppgaven

I denne oppgaven vil konklusjonen fra prosjektoppgaven bli revurdert i lys av sakens utvikling etter prosjektoppgavens slutt. Systemet som skal styres, den motoriserte ortosen, vil bli kartlagt. Oppgaven viser utviklingen og oppbygningen av en stemmestyringsenhet, derunder talegjenkjenningsteori, implementeringsteknologi, systemutvikling og testing.

Kapittel 2 har en kortfattet innføring i tematikken til prosjektoppgaven som denne masteroppgaven bygger på. Kapitlet 3 vil inneholde en forklaring av bestanddelene til den motoriserte ortosen, og veien som er gått for å produsere den. Kapittel 4 viser utviklingen av stemmestyringssystemet fra systemspesifikasjon, valg av implementeringsteknologi til programkode. I kapittel 5 beskrives grensesnittet mellom styringsenheten og den motoriserte ortosen. Resultater fra testing av det ferdige stemmestyringssystemet vises i kapittel 6, sammen med tilbakemeldinger fra testerne og en vurdering av stemmestyrings egnethet for brukere med ALS. I Kapittel 7 er diskusjon, konklusjon og planer for fremtidig arbeid. I vedlegg A er en bruksanvisning for stemmestyringsenheten.

1.3 Arbeidsmetodikk

Ved oppstart av masteroppgaven, revurderte jeg konklusjonen fra prosjektoppgaven i lys av prosjektets progresjon de siste månedene og pasienten sin motoriske funksjon. Som konkludert med i prosjektoppgaven, synes stemmestyring fortsatt å være det beste alternativet. Arbeidet fortsatte med litteratursøk for å finne en implementeringsteknologi som ville passe til ortosen og prosjektets rammer, og samtidig var gjennomførbart innenfor tidsbegrensningene til masteroppgaven. Under byggingen av styringssystemet ble det gjennomført gjentatte samtaler med byggeren av ortosen, Lien. Dette for å avgjøre de ønskede funksjonene og spesifikasjonene til stemmestyringen. Deretter samtaler med veiledere, og spesielt pasienten, rundt brukergrensesnittet for stemmestyringen. Målet var å ferdigstille stemmestyringen for testing en periode før endt master, slik at pasienten fikk tid til å teste den ut og komme med tilbakemeldinger om eventuelle problemer og innvendinger til brukervennligheten.

KAPITTEL 1. INTRODUKSJON

Testresultatene er presentert som totalt snitt av sanne positive kommando i figur 6.1. Snittet for hver av testerne er også inkludert for å vise de individuelle avvikene fra det totale snittet.

I utviklingsfasen er det utført tester av systemet, gjennomført flere samtaler med oppdragsgiver og pasienten. I møte på St. Olavs hospital ble fremdriften for byggingen av ortosen og stemmestyringssystemet presentert og drøftet. Underveis har jeg søkt kunnskap om systemet som skal styres av stemmestyringsenheten og reflektert rundt hvem systemet skal kunne virke for og hvilke behov som skal dekkes.

Fokuset gjennom denne oppgaven har vært å utvikle et system som kan hjelpe pasienten i hverdagen. Et system som er brukervennlig og dekker hans ønsker og behov. Det har vært nødvendig å gjøre styringssystemet så komplett som mulig mellom hver testing, da ortosen enten var under utvikling eller har vært hos pasienten et annet sted i landet. Det har vært jevnlig kontakt med pasienten per telefon og mail for tilbakemeldinger og oppdateringer, to ganger har vi møttes for utprøving og testing. Siden systemets funksjoner ble til underveis og var vanskelig å få testet i praksis, var det sannsynlig at noen ideer ikke ville treffe. Det var derfor viktig å utvikle et system som enkelt kunne endres, dette ble oppnådd ved å modularisere systemet.

2 | Videreføring av prosjektoppgave

Fordypningsprosjektet “Styringsprinsipper for ortose til ALS-pasient” var starten på veien til å utvikle et styringssystem til en motorisk armortose etter oppdrag fra Lien. Prosjektet var et litteraturstudie som kartla den forestående styringsutfordringen og tok et velbegrunnet valg for styringsprinsippet som skulle utvikles til den motoriserte ortosen i en følgende masteroppgave. Prosjektoppgaven inneholdt en introduksjon til kroppens signalsystem, nervesystemet, og forklarte sykdomsgangen ved ALS. Deretter ble egnede styringsprinsipper for den motoriserte ortosen utforsket, deriblant muskelaktivitet målt med EMG, hjerneaktivitet målt med EEG, blikksporing og stemmestyring. Stemmestyring ble konkludert som den mest egnede styringsmetoden basert på pasienten sitt tilfelle. Dette kapittelet vil inneholde en kort innføring i prosjektoppgaven innhold, og en revurdering av oppgavens konklusjon.

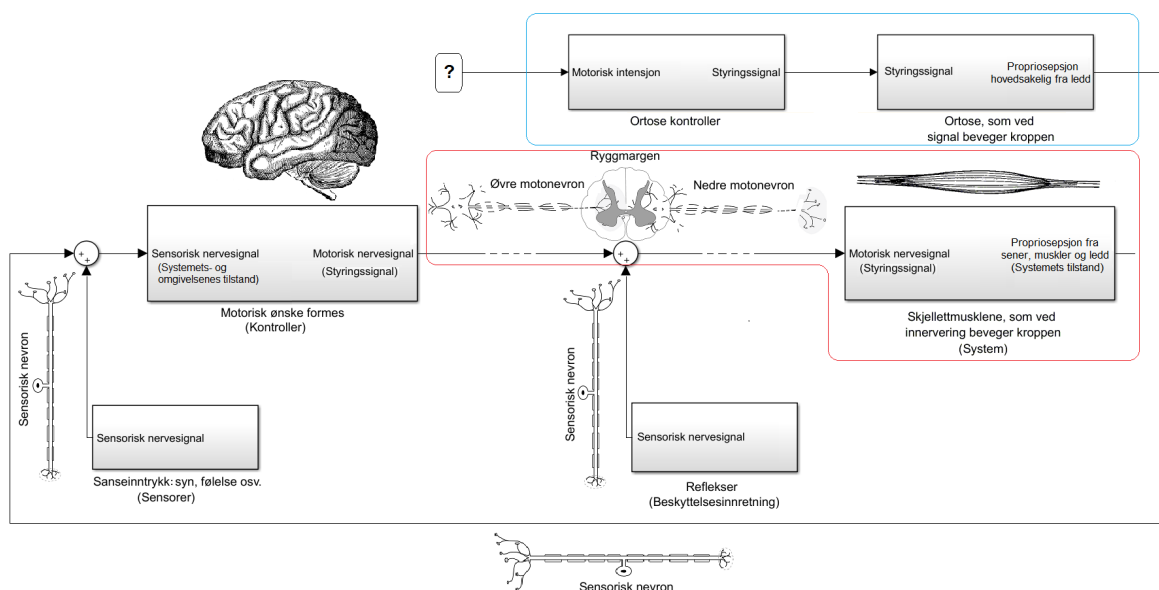
2.1 Kroppens signalsystem og ALS

For bevegelse av en frisk kropp, sendes en motorisk kommando fra hjernebarken til musklene. Den motoriske kommandoen sendes i form av nerveimpulser via motonevroner ned til musklene som er involvert i den ønskede bevegelsen. Når nerveimpulsene når frem til musklene, fører det til at musklene trekker seg sammen og dermed beveger skjelettet.

ALS er en progressiv nevrodegenerativ sykdom, som fører til en økende funksjonsnedsettelse. Sykdommen ALS degenererer kroppens motonevroner, dette fører til at motoriske kommandoer i stadig minkende grad når frem til musklene. Derfor mister ALS pasienter gradvis kontroll over de viljestyrte motoriske kroppsfunksjonene. [1]

I figur 2.1 er et blokkdiagram som viser en kropp med ALS sitt styringssystem, assistert med en motorisert ortose. Delene av kroppens styringssystem som er affisert av ALS er omkran-

set av en rød linje og den motoriske ortosen er omkranset av en blå linje. Diagrammet viser styringssystemet i et kybernetisk perspektiv, isolert fra omverdenen. Blokkdiagrammet gir et forenklet fysiologisk oversiktsbilde, mens bildene viser forenklet anatomisk struktur. I en frisk kropp er hjernen kontrolleren i systemet, og den styrer kroppen til ønsket tilstand ved bruk av skjelettmusklene og med tilbakekobling fra sansene. I figuren illustreres hvilke funksjoner ortosen og ortosens styringssystem spiller ved bevegelse av en kropp hvor signalsystemet ikke lenger virker. Kontrolleren, hjernen, og tilbakekoblingsystemet, sansene, er hovedsakelig intakt, men kontrollen over skjelettmusklene som er pådragsorganene minker i stadig økende grad.[1] I denne masteroppgaven skal blokken “ortosens kontroller” fra figuren utvikles. Spørsmålsteget i figuren, signalet som angir motorisk intensjon, vil være inngangssignalet til styringsenheten.



Figur 2.1: Blokkdiagram som viser veien fra motorisk intensjon til motorisk aksjon hos en pasient med ALS og motorisert ortose. [6]

2.2 Revurdering av prosjektoppgavens konklusjon

Prosjektoppgaven konkluderte med at stemmestyring var den mest velegnede styringsmetoden for ortosen. Stemmestyring syntes å passe best på det sykdomsstadiet pasienten befant seg på da, og var den løsningen som passet best med utvikler og bruker sine ønsker og behov.

Ved avslutning av prosjektoppgaven var trykksensorstyring under utarbeiding. Trykksensorsty-

ring blir nærmere forklart i underkapittel 3.3.2. Tanken var at stemmestyringen kunne fungere som et supplement til trykksensorstyringen, eller eventuelt senere som en erstatning for den. Noen måneder etter prosjektoppgavens slutt, viste trykksensorstyringen å ikke fungere.

I lys av at trykksensorstyringen ikke kunne brukes og at pasienten sitt sykdomsbilde hadde endret seg, var det passende med en revurdering av prosjektoppgavens konklusjon før et nytt styringssystem ble utarbeidet. I prosjektoppgavens konklusjon ble stemmestyring anbefalt på grunnlag av at det er raskt realiserbart, rimelig, det vil ikke vil være i veien eller ukomfortabelt og det kan utvikles uten jevnlig testing med pasienten. Utover dette var det viktig for brukeren at styringssystemet skulle være enkelt å ta på, noe stemmestyring vil kunne være.

Dersom trykksensorstyring ikke viste seg å fungere, ble det foreslått å revurdere styring ved bruk av signaler fra muskelsammentrekninger (EMG-signaler). Etter samtale med pasienten, ble det klart at førligheten i armene hadde blitt svekket i den grad at det var vanskelig med kontrollerte muskelsammentrekninger i områdene vi undersøkte for styring med EMG-signaler i prosjektoppgaven. Det kan hentes EMG-signaler fra andre områder på kroppen, men det vil ikke gi en like god brukeropplevelse. Elektroder for å måle EMG vil være krevende å ta på, og det vil være et mye større behov for jevnlig testing med pasienten. Et stemmestyringssystem vil kunne gjøres nærmest vedlikeholdsfritt, mens et EMG-styringssystem vil kreve kontinuerlige tilpasninger ettersom muskelaktivitet svekkes.

I samtale med pasienten og oppdragsgiver ble det enighet om at stemmestyring fortsatt var det best egnede styringsalternativet for dette prosjektet. Stemmestyring vil kunne realiseres innenfor prosjektets økonomiske- og masterens rammer og det vil møte pasientens ønsker for et styringssystem.

3 | Den motoriserte ortosen

En ortose er betegnelse på en støtteanretning som assisterer bevegelse.[1] Den motoriserte ortosen består av en støtteskinne for brukerens høyre arm og to motorer som fører til bevegelse i skulder- og albueledd. De to motorene gir ortosen to frihetsgrader. Utover dette har ortosen et passivt ledd ved skulderen, som gir ytterligere to frihetsgrader, se figur 3.1. Leddet gjør det mulig med abduksjon og internal- og eksternal rotasjon i skulderleddet mens ortosen er på. Dette vil si at armen kan beveges ut fra kroppen og at den kan roteres noe, enten ved muskelkraft eller ved å støtte albuen mot for eksempel et armlene. Ortosen gir brukeren en bevegelsesrekkevidde fra nøytral utstrakt posisjon og opp til ansiktet, slik kan brukeren nå munn, nese, øyne, ører og panne. Denne ortosen er utviklet for en spesifikk pasient, ut fra pasientens ønsker og behov. Informasjon om ortosen er innhentet fra utvikleren av ortosen, Terje K. Lien.

3.1 Utvikling av den motoriserte ortosen

Under utarbeidingen av ortosen var det av hensyn til brukervennlighet og komfort, viktig å minimere vekt og volum. Ønsket fra pasienten var at ortosen skulle være lett å ta på og av, og helst at pasienten skulle kunne gjøre dette uten assistanse.

Tidligere har det vært laget en enklere ortose til pasientens høyre arm, bestående av underarmsortose beveget av én motor ved albueleddet. Motoren ble styrt av en joystick med venstre hånd. Underarmsortosen ble utviklet av Norsk Teknisk Ortopedi AS på Hamar, og den ble gjort motorisert av Lien. Etersom pasienten sin tilstand ble forverret, viste det seg nødvendig å videreutvikle ortosen med en motor for å bevege skulderleddet. Det var også behov for et annet styringsystem til den nye ortosen, siden finmotorikken i pasienten sine hender var blitt svekket. Den nye ortosen ble utviklet av Lien parallelt med utviklingen av stemmestyringssystemet.



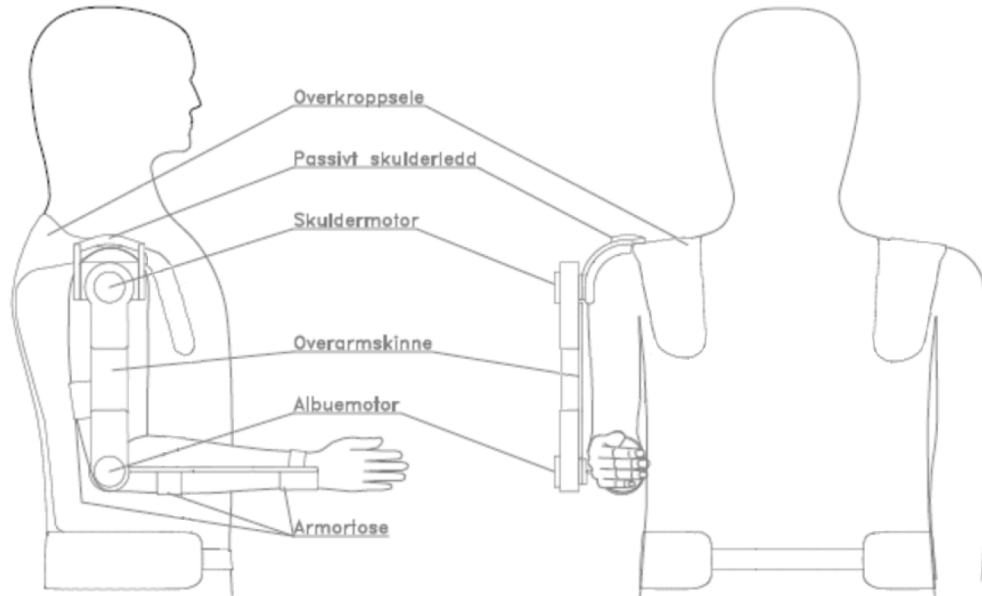
Figur 3.1: Den motoriserte ortosen med fotspaker til styring og batteripakke.[7]

For å feste den videreutviklede ortosen til kroppen og fordele vekten på en komfortabel og ergonomisk riktig måte, ble det laget et spesielt tilpasset overkroppssele. I figur 3.2 er en skisse av den videreutviklede motoriserte ortosen med overkroppsselen. Overkroppsselen ble produsert av Ortopediteknikk AS i Tønsberg. Den er utformet etter mål fra en overkroppsmoell laget etter 3D-skanning av brukerens overkropp.

Utforming av ortosen ble avgjort etter nøye beregninger og mål av pasienten. I samarbeid med fysioterapeuter og leger ved St. Olavs hospital ble den motoriserte ortosen utformet for å kunne bevege brukerens arm komfortabelt og ikke føre til slitasje eller skade. For å estimere rotasjonsentrene i brukerens arm ble det tatt røntgenbilder med mynter på huden som referanser. Festeanordningen av ortosen til overkroppsselen ble laget etter måling på overkroppsmoellen og røntgenbildene. Det lyktes ikke å få til en løsning hvor pasienten selv kunne feste ortosen uten assistanse. Ortoosen må løftes på brukeren og festes med to reimer.

3.2 Den motoriserte ortosens komponenter

Delsystemene som sammen danner den stemmestyrte motoriserte ortosen, vises i figur 3.3 i et “deployment diagram” i Unified Modelling Language (UML). Et deployment diagram fremstiller



Figur 3.2: Konstruksjonsskisse av den motoriserte ortosen med overkroppssele.[8]

et system sin fysiske oppbygning, hvor komponentene er vist som bokser og knyttet sammen av kommunikasjonslinjer. [9] I diagrammet er tredimensjonale bokser brukt for å symbolisere hardware som inngår i systemet. Noen enheter kjører software, dette blir illustrert med en firkant inne i den tredimensjonale hardware-boksen. Piler er lagt til på kommunikasjonslinjene for å vise retningen av informasjonsflyt mellom de ulike delsystemene. Oppbygningen til stemmestyringsenheten beskrives i underkapittelet 4.2 og vises i figur 4.3. De følgende delkapitlene vil omhandle motorstyreren og ortosens aktuatorer.

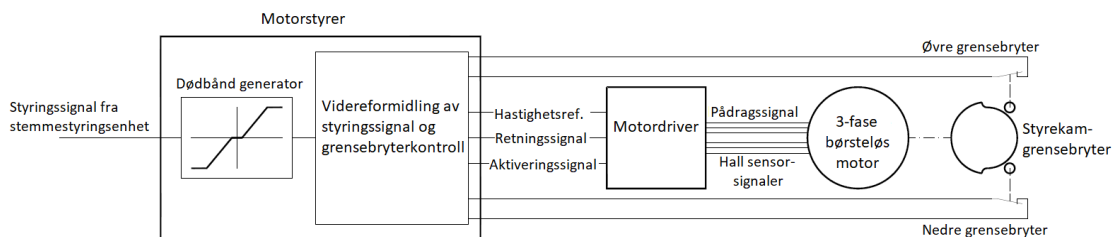


Figur 3.3: Delsystemene som utformer den stemmestyrte ortosen.

3.2.1 Motorstyreren

Motorstyreren er utviklet av Lien, og består av et spesialutviklet kretskort med ATmega168 mikrokontroller programmert med programmeringsspråket Assembly. Motorstyreren mottar styringssignal fra stemmestyringsenheten, filtrerer og omformulerer signalet, og sender det videre til motordriveren. Se figur 3.4. Grensebryterne er illustrert som en forlengelse av motoren helt til høyre i figuren. Styringssignalet fra stemmestyringsenheten mottas i form av to spenningsverdier, en for hver motor, mellom 1,5 V til 3,5 V. Motorstyreren filtrerer dette spenningssignalet med et dødbånd og metningsgrenser, deretter omformuleres spenningsverdiene til separate retningssignal og hastighetsreferanse for albue- og skuldermotor. Styringssignal i dødbåndet rundt 2,5 V, fører ikke til bevegelse. Styringssignal mellom 1,5-2,3 V og 2,7-3,5 V gir retningssignal som henholdsvis gir en ekstensjon og fleksjon i skulderleddet. For albue er det omvendt, verdier mellom 2,7-3,5 V gir retningssignal ned og verdier mellom 1,5-2,3 V gir retningssignal opp, som fører til henholdsvis fleksjon og ekstensjon i albueleddet.

Absoluttverdien av avviktet fra spenningssignalet og 2,5 V angir referansehastigheten. Retningssignal og hastighetsreferanse sendes videre til motordriveren, sammen med et aktiveringssignal. Aktiveringssignalet sendes dersom styringssignalet ligger utenfor dødbåndet.



Figur 3.4: Systemskisse over ortosens motorstyrer og aktuatorer.[10]

3.2.2 Ortosens aktuatorer

Motorer med motordrivere og nedgiring, utgjør ortosens aktuatorer. Retningssignalet og hastighetsreferansen fra motorstyreren videreformidles til motoren via motordriveren pådragssignal. Motordriveren har tilbakekobling fra motoren, og gir pådrag for at den faktiske hastigheten skal tilsvare referansehastigheten. Tilbakekoblingen er i form av "Hall sensorsignaler. Hall sensor signalene gir informasjon om motorens hastighet ved at det sendes en impuls til motordriveren

hver gang motoren gjør en $\frac{1}{16}$ omdreining. Ved å telle pulser og måle pulsfrekvensen kan ortosens posisjon og motorenes hastighet estimeres.

For å gjøre motorene så lette som mulig er ortosen bygget med børsteløse motorer i flat utførelse. Motorene har stor diameter, dette gir høyt dreiemoment i forhold til volum og vekt. En utfordring med motorene er at de ikke leveres med integrert bremse, dette skyldes at motorene har roterende hus. Det viste seg å være behov for bremses på motorene, siden armens vekt førte til at ortosen mistet høyde når det ikke var pådrag. Lien utviklet en skivebremse til ortosen, denne gjorde at ortosen kunne holde en posisjon uten motorpådrag.

For å unngå at ortosen skulle kjøre til en anatomisk skadelig posisjon, ble det laget grensebrytere som sikkerhetsmekanisme. Disse sperret for et videre pådrag i samme retning når ortosen hadde nådd øvre eller nedre grensebryter. For ekstra sikring ble det også plassert fysiske sperrer etter grensebryterne, disse vil stanse ortosens bevegelse dersom grensebryterne ikke har lyktes i å stanse motoren.

Kravet til ortosens løfteevne er å løfte sin egen vekt, vekten av brukerens arm og i tillegg en liten nyttelast. Lasten kan være en kopp med drikke eller liknende. For å utføre slike oppgaver ble motorenes dreiemoment økt ved nedgiring med Harmonic Drive enheter. Harmonic Drive er et reduksjonsgir som gir stor utveksling.

3.3 Tidligere styringsmetoder for den motoriserte ortsen

Lien har utviklet flere styringsmuligheter av den motoriserte ortosen. Noen har blitt byttet ut underveis grunnet endring i pasienten sitt sykdomsbilde. Noen idéer har ikke latt seg bruke i praksis. I dette delkapittelet beskrives og vurderes tidligere styringsmekanismene som er utviklet til ortosen.

3.3.1 Joystick

Den første utgaven av den motoriserte ortosen, hadde én motor og ble styrt med en joystick. På dette tidspunktet i sykdomsutviklingen var dette en god og funksjonell løsning, med svært nøyaktig og trinnløs styring. Ved ALS er det individuelt om finmotorikken eller de store muskelgruppene svekkes først. For brukere med tilstrekkelig finmotorikk synes joystick å være et

godt styringsalternativ. En ulempe ved dette styringsprinsippet kan være at joysticken okkuperer én hånd til å bevege den andre. Det kan være ressurskrevende å manøvrere for brukere med svekket finmotorikk, og manøvrering av joystick er en monoton bevegelse som kan føre til spenningssmerter.

3.3.2 Trykksensorer

Ved avslutningen av prosjektoppgaven var trykksensorstyring under utarbeiding. Dette var en idé med et godt potensiale, ettersom det var tilpasset pasientens daværende førlighetsgrad. Trykksensorstyring ble laget med tre trykksensorer plassert rundt skulderleddet. Disse skulle aktiveres ved at brukeren beveget skulderen mot den enkelte trykksensor. Enda pasienten hadde mistet kontroll over musklene for å løfte armene, hadde han fortsatt kontroll over skulderbevegelse. Tanken var å utnytte denne førligheten til å gi pådrag på ortosens motorer. Én trykksensor ble plassert foran skulderen, én over og én bak skulderleddet. Trykk mot kun den fremre sensoren ga albuemotoren signal om å løfte underarmen, trykk mot den fremre og den øvre samtidig ga signal om løft til skuldermotoren. Trykk mot den bakre sensoren senket armen.

Å bevege ortosen ved å aktivere muskler relatert til den ønskede bevegelsen, kan lette innlæring og gi brukeren en følelse av å selv bevege armen. Dette kan oppleves intuitivt for brukeren, og kan gi en følelse av kontroll over bevegelsene. Ved bruk av dette styringsprinsippet vil brukeren bevege armen i den grad det er mulig. Bevegelse er helsebringende, men kan også være ressurskrevende for brukeren.

Dessverre viste det seg svært vanskelig for brukeren å trykke inn den bakre trykksensoren med ortosen på. En annen utfordring viste seg når ortosen løftet armen, det førte til kontinuerlig press mot den fremre og øvre sensoren, og låste armen i øverste posisjon. Det ble ikke funnet noen løsninger på disse hindringene, dermed ble trykksensorprosjektet avsluttet. Problemene ved trykksensorene i praksis, gjorde at det ikke ble mulig å stanse løftebevegelse eller å senke armen. Det kunne vært mulig å finne en løsning på dette med software eller flere sensorer et annet sted, men det ville ført til at systemet ikke lenger hadde enkelheten og brukervennligheten som gjorde styringsprinsippet attraktivt til å begynne med.

3.3.3 Fotstyring

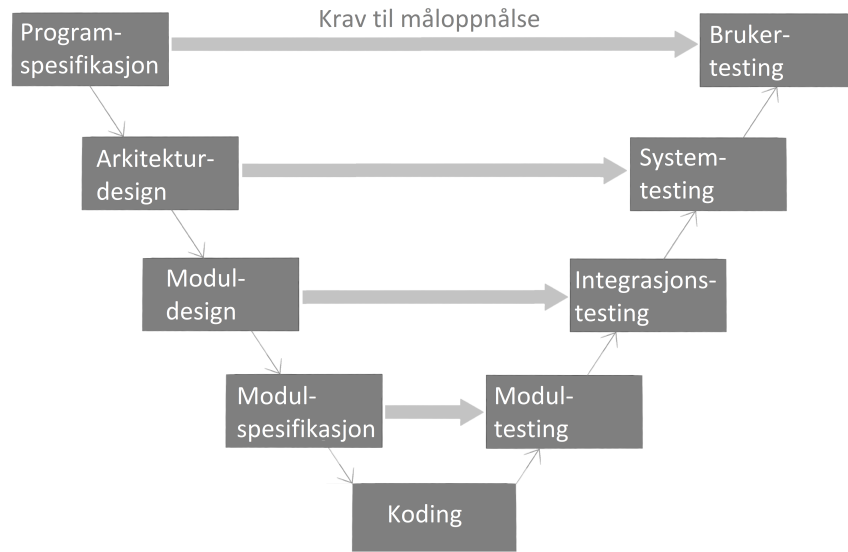
Etter trykksensorene viste seg å ikke fungere, utviklet Lien fotstyring til ortosen. Fotstyringen har to potensiometere med små styrespaker som styrer pådraget på ortosens motorer. Styrespakene gir en trinnløs styring av motorpådraget ved at de beveges opp eller ned med tærne. Styringsenheten er koblet til ortosen med en ledning. Fotstyringsenheten er avbildet sammen med ortosen i figur 3.1.

Pasienten har fortsatt god førlighet i tærne, så det var en god løsning å utvikle fotstyring. Pasienten kan gå og ortosen er utviklet for å kunne forflytte seg med. I pasienten sitt tilfelle viste det seg derfor å være en ulempe at fotstyringsenheten ikke er portabel. Han klarer ikke flytte fotstyringen på egenhånd og siden styringsenheten henger fast i ortosen med en ledning, kan han ikke forflytte seg når fotstyringen er koblet på. For en bruker som ikke kan gå, men har kontroll over finmotorikken i føttene, kan fotbryter være en god idé å ha festet til en rullestol. Fotstyringen er i bruk i dag og pasienten er godt fornøyd med den, bortsett fra at den er begrensende for hans mobilitet.

4 | Utvikling av stemmestyringsystemet

Utviklingen av stemmestyringssystemet vil ha et stort element av instrumentering, systemutvikling og embedded-programmering. Organiseringen av utviklingen er gjort etter inspirasjon fra V-modellen. Denne modellen kartlegger trinnene som bør inngå i systemutvikling fra en idé til et ferdig produkt. V-modellen sikrer en bevisst og systematisk utvikling, dette er viktig for å oppnå en implementasjon som er robust og mulig å utvide og gjenbruke. Ved å formulere testkrav til systemet i hvert ledd i utviklingsprosessen sikres det at produktet som utvikles står til produktspesifikasjonene, samt at det blir det lettere å oppdage feil og mangler underveis. Når testene utføres etter produktet er utviklet, gir det en pekepinn på om målet med det utviklede systemet er nådd og om det presterer som forventet. [11] [12]

V-modellen tilpasses til produktutviklingen den skal fungere for, det finnes mange ulike versjoner av V-modellen. I figur 4.1 er en modell tilpasset til steg som var relevante for utviklingen av stemmestyringsystemet. På venstre side i figuren er stegene for utvikling av systemet. Stegene på høyre side utgjør testing og validering av systemets moduler og det sammensatte produktet. Pilene som går horisontalt mellom boksene illustrerer at det ved hvert steg i utviklingen burde settes opp krav til måloppnåelse for å sikre at det ferdige produktet møter produktkravene formulert under det første steget, produktspesifikasjon. I løpet av systemutviklingen ble vurdering av måloppnåelse enten formulert som kriterier som skulle innfris eller ved planlegging av testing som ville gi en indikasjon på systemets prestasjon.



Figur 4.1: Fremstilling av V-modellen.

4.1 Produktspesifikasjon

Første steg i V-modellen er å kartlegge de ønskede spesifikasjonene til det ferdige produktet. For et vellykket system er det viktig med kommunikasjon og forståelse mellom oppdragsgiver, produktutvikler og produktets målgruppe. Som det fremkommer i delkapittel 2.2 som revurderer prosjektoppgavens konklusjon, var både oppdragsgiver og pasienten enig i at en stemmestyringsenhet var den best egnede styringsenheten å utvikle for ortosen på dette tidspunktet. Ønsket til oppdragsgiver og pasienten var at enheten skulle bli ferdig i løpet av masteroppgaven, at den skulle være kompatibel med ortosen og gjerne at den kunne brukes på norsk. Den skulle fortrinnsvis være kompakt og rimelig. Det ble vurdert om det var hensiktsmessig at stemmestyringsenheten skulle være personavhengig, så den kun ville reagere på pasientens stemme, men dette ble ikke ytret som et spesifikasjonskrav.

Som brukertesting, det siste steget i figur 4.1, vil det utføres en Site Acceptance Test (SAT) av stemmestyringsenheten. SAT er produkttesting som utføres i omgivelsene produktet skal arbeide i.[13] I dette tilfelle vil det si at pasienten tester styringsenheten for styring av ortosen hjemme hos seg selv. Testing ble etterfulgt av samtaler med pasienten for tilbakemeldinger om systemets egnethet. Resultatene av testingen er vist i delkapittel 6.1.

4.2 Valg av implementeringsteknologi

I neste steg i V-modellen omformes produktkravene til tekniske krav og det foretas valg av komponenter. Under denne systemutviklingen ble valg av implementeringsteknologi gjort før systemets ønskede arkitektur ble designet, dette steget deles derfor over to delkapitler, “Valg av implementeringsteknologi” og “Arkitekturdesign og analyse av utfordringer ved stemmestyring av armortose”. Før implementasjonsteknologi er valgt er det ikke sikkert hvilke muligheter og begrensninger implementeringsplattformen ville gi, derfor starter planleggingen av stemmestyringssystemets arkitektur etter dette steget.

Ved valg av implementeringsteknologi ble det søkt etter muligheter som ville kunne realiseres innenfor den gitte tidsrammen og som var økonomisk forsvarlig. Det var ønskelig at den ferdige stemmestyringsenheten skulle ha støtte for norsk og engelsk språk og kunne sende kompatibelt utsignal til ortosens motorer. Målet var å produsere en stemmestyringsenhet som var robust, prisgunstig, like viktig var det at enheten ble raskt ferdig for å kunne utprøves og tas i bruk av pasienten. Den opprinnelige planen var at stemmestyringsprogrammet skulle ligge på motorstyren. Etterhvert ble det isteden valgt at det skulle være en egen enhet. Dette gjorde at ortose og styringsenhet kunne utvikles separat og virke uavhengig av hverandre.

Å utvikle et talegjenkjenningsprogram er en stor oppgave, derfor var første tanke var å finne et open source talegjenkjenningsprogram eller å kjøpe et ferdiglaget program som kunne kjøre på en enhet som tilfredsstilte kravene ovenfor. Da det ikke lyktes å finne en slik løsning som innfridde kravene, gikk søket videre til ferdige enheter med talegjenkjenning. Flere enheter ble funnet, deriblant Geeetech Voice Recognition Module. Denne som støtter engelsk og kinesisk språk, og kan lagre totalt 15 kommandoer som må trenes opp før de kan gjenkjennes. [14] Voice Recognition Module V3 er et annet alternativ hvor kommandoer også må trenes før de kan gjenkjennes og maksimalt 7 kommandoer kan lyttes til om gangen. [15]

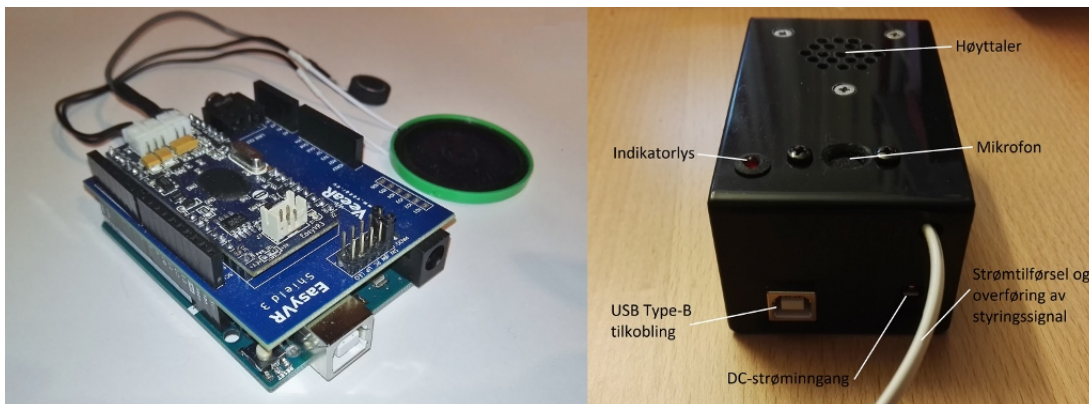
Valget falt på “EasyVR Shield 3.0” som er en modul kompatibel med Arduino kretskort, siden denne enheten var rimelig, godt dokumentert, hadde gode tilbakemeldinger og kunne kjøpes fra en pålitelig forhandler. EasyVR kan trenes opp til å støtte norske kommandoer og støtter engelske kommandoer uten å måtte trenes. Enheten hadde liten nok størrelse til å passe til formålet og skal kombineres med et Arduino kort som gir mange muligheter for tilkobling til ortosen. Tid er en viktig faktor for pasienter med ALS. I stedet for å finne de optimale løsningene, er det viktig å finne funksjonelle løsninger raskt. Da det ble funnet en enhet som så ut til å imøtekomme alle

krav, ble den umiddelbart bestilt for å starte utviklingsarbeidet.

4.2.1 Talegjenkjenningsmodulen EasyVR

EasyVR utvikles av Robotech. Robotech står også bak dataprogrammene EasyVR Commander og QuickT2SI som har vært nødvendig ved utviklingen av talegjenkjenningsenheten. EasyVR-modulen har ikke støtte for norske ikke-personavhengige (SI) kommandoer. Den har 26 inkluderte SI-kommandoer på enten amerikansk-engelsk, italiensk, tysk, fransk, spansk eller japansk. Det må kjøpes lisens til dataprogrammet QuickT2SI for å legge til nye SI-kommandoer, der er de overnevnte språkene støttet i tillegg til mandarin. Med QuickT2SI kan totalt 336 nye kommandoer legges til i grupper på maks 12 kommandoer i hver.

Personavhengige (SD) kommandoer kan lages på norsk og legges til inn på EasyVR-enheten med gratisprogrammet EasyVR Commander. Et skjermbilde som gir veiledning til bruk av dataprogrammet er vist i vedlegg A, figur A.9. EasyVR Commander benyttes også til å legge inn SI kommandoer opprettet med QuickT2SI. Totalt 32 SD kommandoer kan opprettes og fordeles over maksimalt 16 grupper. SD-kommandoene er ment å bare fungere med personen som har trent opp kommandoene.

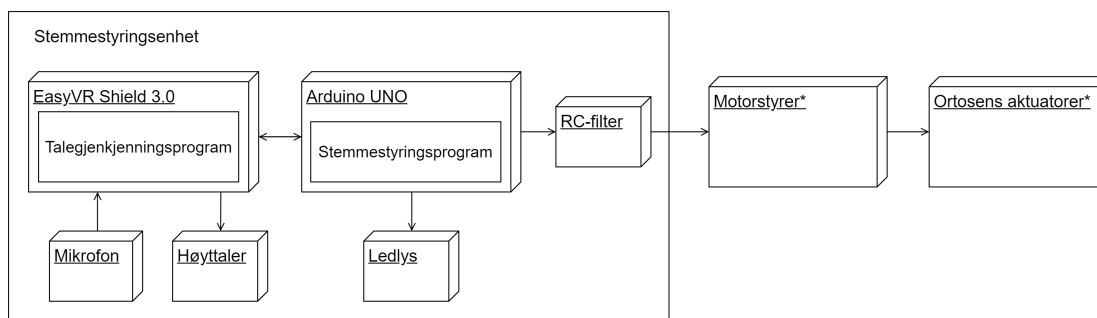


Figur 4.2: EasyVR-modul slått sammen med Arduino Uno, med mikrofon og høyttaler. Til er komponentene innkapslet i en boks.

EasyVR-modulen kobles til et Arduino kort, kompatible kort er Arduino Duemilanove, - Due, - Mega, - Leonardo og - Uno. [16] Arduino Uno kort er et av de rimeligste kortene og det har de nødvendige spesifikasjonene for å møte produktkravene. Kortet kan sende styringssignal til

ortosen, har mulighet for USB-tilkobling og tilstrekkelig med minne for et omfattende stemmestyringssystem. På dette grunnlaget ble Arduino Uno anskaffet. En liten mikrofon fulgte med EasyVR-modulen. For å ha mulighet til å gi tilbakemeldinger til brukeren av stemmestyringsenheten ble det også kjøpt en liten 0.25W høyttaler til å spille av lydklipp. EasyVR har også en 3,5 mm jack utgang dersom det er ønskelig på et senere tidspunkt å gi tilbakemeldinger fra enheten via et headset. Komponentene, og en boks til å innkapsle disse, er kjøpt fra www.sparkfun.com.

Figur 4.2 viser EasyVR-modulen slått sammen med et Arduino Uno kretskort og koblet til høyttaler og mikrofon. I figur 4.3 utvides deployment diagrammet fra figur 3.3 til å vise de indre komponentene til stemmestyringsenheten. De to programmene som utgjør programvaren på implementeringsplattformen, talegjenkjenningsprogrammet og stemmestyringsprogrammet, vil beskrives i henholdsvis delkapitlene 4.2.2 og 4.5. Grensersnittet mellom stemmestyringsenheten til motorstyreren, og derunder RC-filtrering av styringssignalet, beskrives i kapittel 5.



Figur 4.3: Delene som utformer den stemmestyrte ortosen.

4.2.2 Talegjenkjenningsprogrammet på EasyVR-modulen

Talegjenkjenningsprogrammet består av en rekke funksjoner som kan utføre oppgaver slik som gjenkjenning av tale og avspilling av lydklipp. Disse funksjonene er lagt inn av produsenten og kan ikke endres, men de kan kalles av stemmestyringsprogrammet når det er ønskelig at talegjenkjenningsenheten utfører en oppgave. [17]

Talegjenkjenningsteknologien brukt i EasyVR-modulen er en RSC4128-prosessor fra Sensory Inc. Sensory Inc. er verdensledende innen embedded-talegjenkjenningsteknologier. SI-kommandoer gjenkjennes med en algoritme som er en hybrid mellom “Hidden Markov Model” og nevraltnett. En slik hybrid kan utnytte neuralnett sin evne til og klassifisering, og Hidden Markov Model sin evne for sekvensiell modellering. SI-kommandoer opprettes med dataprogrammet QuickT2SI,

hvor tekst gjøres om til sett av SI-kommandoer.[18] [19] [20]

SD-talegjenkjenningen krever at kommandoer trenes opp med opptak av personen som skal bruke enheten, hvor vedkomne gjentar hver kommando helst 2-3 ganger. SD-talegjenkjenningen benytter algoritmen “dynamic time warping” for å finne likheter mellom de trente kommandoene og en kommando som sies. Dette er en mønstersammenlignings-teknologi som kan gjenkjenne samme kommando i to lydklipp selv om de har ulik hastighet. Metoden fordreier unlineært lyd-signalet til å ligne på malen og deretter kalkulerer likheten mellom dem. [19] [20]

4.3 Arkitekturdesign ved analyse av utfordringer for stemmestyring av armortose

I dette steget skal systemets arkitektur designes. Under planlegging av hvilke funksjonaliteter som skulle inngå i stemmestyringssystemet for den motoriserte ortosen, var det viktig å prøve å forutse hvilke utfordringer som kunne oppstå ved bruk.

Det var det usikkert hvor pålitelig talegjenkjenningen ville være, i et krisemaksimert tilfelle kunne pasienten sagt “stopp” og enheten gjenkjent “full fart opp”. Pålitelighet er en utfordring ved styring av motoriserte hjelpemidler med talegjenkjenning.[21][22] Pasienten ytret et ønske om styring hvor bevegelse kunne startes med en kommando og stoppes ved en annen. Tanken var at pasienten kunne stanse bevegelse ved å si “stopp” når armen var i ønsket posisjon eller ikke gi mer enn startkommando for å la armen fortsette til endeposisjon. Et slikt styringssystem ville satt svært store krav til påliteligheten til talegjenkjenningen. Min vurdering var at dette ikke ville være forsvarlig basert på de innledende testene av talegjenkjenningsenheten. I stedet ble det foreslått for pasienten at ønsket bevegelse skulle angis til enheten før bevegelse ble startet. Dersom pasienten ville stanse bevegelsen før den kom til angitt posisjonen, skulle dette kunne gjøres med kommandoen “stopp”.

En svakhet som tidlig viste seg med talegjenkjenningsenheten var at den hadde mange falskt positive kommandogjenkjennelser. Den reagerte på prat eller lyder og registrerte det som en kommando. Det var viktig å forhindre at bevegelse startet uten hensikt, derfor ble det planlagt å ha flere lag med kommandogjenkjennelse før bevegelse ble startet. Dette reduserer sannsynligheten for en utilsiktet bevegelse, da det vil kreve flere falskt positive gjenkjennelser på rad uten

at brukeren legger merke til det for å starte en uønsket bevegelse. Brukeren har også mulighet for å si en kommando som vil stanse en påbegynt bevegelsesbeskrivelse når det har forekommet en falsk registrering. Falske registreringer var et vesentlig problem ved SI-talegjenkjenning, men det forekom svært sjeldent ved SD-talegjenkjenning. Dette ble erfart gjennom utviklingen av systemet, og kommer også tydelig frem i testresultatene i delkapittel 6.1.

En utfordring ved stemmestyring er at den presterer best i rolige omgivelser og kan ha variabel prestasjon i omgivelser med støy.[23] [22] Dette gjør det utfordrende å bevege ortosen under en samtale, med TV-en på eller i andre omgivelser med støy. Ortosen kan også være vanskelig å bruke til å spise eller pusse tenner da det er vanskelig å gi kommandoer med noe i munnen. I de sistnevnte tilfellene kan SI-talegjenkjenning være en fordel, da brukeren kan gi kommando om å heve gaffelen og en som sitter ved siden av kan gi kommando om å senke armen. For å gjøre enheten bedre egnet til ulike omgivelser ble det planlagt en modus for å stille talegjenkjenningssensitivitet. På den måten kan sensitiviteten justeres ned når enheten har vanskelig for å gjenkjenne kommandoer og justeres opp hvis det er mange falskt positive registreringer.

Å velge kommandoene som skulle inngå i til stemmestyringssystemet viste seg å være en utfordring. For et intuitivt og brukervennlig system er det viktig at kommandoene er passende og naturlige. For at talegjenkjennings algoritmen skal prestere best burde kommandoene være ulike og ha flere enn én stavelse. “Stopp” og engelsk “stop” er enstavelsesord som viste seg å være relativt vanskelig for enheten å gjenkjenne, samtidig er dette kommandoer som er naturlig å si når en ønsker å stanse en bevegelse. Alternativer som “avbryt” og “abort” ble utprøvd, men det ble enighet mellom utvikler og pasient å beholde “stopp” og “stop”. For å løse dette problemet ble det i tillegg lagt til en ekstra kommando som også stoppet bevegelse, “ferdig” og “finish”. I tillegg skulle denne kommandoen navigere ut av bevegelsesmodus.

SD-talegjenkjenning trenger å trenes opp før det kan benyttes første gang, dette kan gjøres i dataprogrammet EasyVR Commander. Det vil ikke være mulig for pasienten å koble enheten til pc og trene opp kommandoer. Det hadde vært mulig å reise ned til pasienten og trene opp kommandoene, men det ville vært vanskelig å vedlikeholde. For at pasienten skulle kunne trene norske kommandoer på egenhånd ble det planlagt en programmeringsmodus. På den måte ville pasienten selv kunne vedlikeholde SD-kommandoene og trene kommandoer opp på nytt dersom de vanskelig ble gjenkjent. Dette kan vise seg nyttig ved en forkjølelse eller gradvis nedsettelse av artikulasjon.

KAPITTEL 4. UTVIKLING AV STEMMESTYRINGSSYSTEMET

Stemmestyling er en tidkrevende styringsmetode, dette gjør prinsippet mindre egnet for små justeringer i hastighet og posisjon. I pasientens tilfelle er det heller ikke nødvendig med korrekt posisjon på millimeteren, da overkroppen og hodet kan beveges i forhold til ortosen. For å bevege ortosen til posisjoner som pasienten bruker ofte, ble det lagt til en hurtigmodus hvor det var tre forhåndsinnstilte posisjoner.

Talegjenkjenningen fra EasyVR fungerer ved at enheten lytter etter én gruppe kommandoer om gangen. For SI-kommandoer er maksimal størrelse på en gruppe 12 kommandoer. Det er viktig at dette blir forståelig for pasienten. Det kan være forvirrende for en bruker at enheten responderer på ett sett kommandoer når den er innstilt på gjenkjenning i én gruppe, og responderer på helt andre kommandoer når den lytter etter kommandoer i en annen gruppe. Da talegjenkjenningen oppdeler etter kommandogrupper virket det naturlig å utvikle stemmestyriingsenheten som en tilstandsmaskin, hvor kommando-gruppene er tilstandene. I hver tilstand er det kun et knippe gyldige kommandoer som systemet vil reagere på. For å gjøre det enkelt for brukeren, vil oppbygningen til engelskmodus og norskmodus gjøres identisk, dermed har begge språk maks ha 12 kommandoer i hver gruppe. Det vil også bli forsøkt å gjøre navigasjonskommandoene til de to modusene til naturlige oversettelser, så det ikke skal være krevende å lære utenat.

Stemmestyriingsenheten skulle utvikles med foroverkobling, hvor det ble gitt pådrag på motorene så lenge et styringssignal ble sendt fra stemmestyriingsenheten. Selv om styringssystemet ikke har tilbakekobling, kan ortosens posisjon beregnes utfra styringssignalet som er sendt. Ortosens pådrag varieres etter en hastighetsreferanse, dermed kan tiden ortosen bruker fra minimalt til maksimalt utslag benyttes for en posisjonsstyring med en absolutt posisjonsreferanse. En slik posisjonsstyring vil ikke være helt nøyaktig, men i dette tilfellet er nøyaktigheten det tilstrekkelig. Ortosens to motorer bruker 7 sekunder fra maksimalt til minimalt utslag ved rask hastighet (styringssignal 3,5 V eller 1,5 V). Ved sakte hastighet bruker ortosen omtrent tre ganger så lang tid. En styring med relativ posisjonsreferanse vil si at brukeren angir den ønskede ortoseposisjonen i forhold til den nåværende posisjonen, slik som å gi beskjed om pådrag oppover i 3 sekunder for albueledd. Ved styring med absolutt posisjonsreferanse kan brukeren angi en spesifikk posisjon som er ønsket basert på en fast posisjonsreferanse, også blir det korresponderende pådraget sendt til motorene, slik som å gi beskjed om at ortosen skal flytte armen til nøytral utstrakt posisjon.

I prosjektoppgaven ble det foreslått å ta utgangspunkt i antall grader det var ønskelig at ortosen skulle forflytte seg. Dette ville vært mulig å benytte både ved en absolutt og en relativ posisjonsreferanse. Da grader kunne synes å være litt for teknisk, ble det foreslått å heller angi pådrag

med sekunder. Det vil si å angi ønsket styringssignal basert på antallet sekunder motoren skal kjøre, hvor syv sekunder gir maskimalt utslag ved rask hastighet. For å sende styringssignal i korrekt tidsperiode ble en innebygget timer på Arduinoen brukt. Pasienten forteller hvor mange sekunder det er ønsket at motoren kjører, så starter timeren og teller ned til bevegelsen skal stanses.

Arduino Uno har en 16 MHz keramisk resonator og tre “timere” (Timer0, timer1 og timer2) som kan brukes som tidtakere da de er koblet til resonatorens svingninger. Timer 0 benyttes av EasyVR sitt talegjenkjenningsprogram og timer 1 er koblet til utgangene 9 og 10 som benyttes til å sende styringssignalet, mens Timer 2 er koblet til utgangene 11 og 12 som ikke benyttes. dermed ble Timer2 valgt til å telle ned tiden før en bevegelse skal stanses. Arduinoen har en klokkehastighet på 16 MHz som vil si en svingning per 62,5 ns. Telleren ble nedskalert med 64 bits, som er maksimal nedskalering for timer 2. Telleren ble nedskalert for å kunne telle til 7 før telleren fikk overflyt, altså at telleren overgår sin kapasitet og begynner på 0 igjen. Når telleren når angitt en satt verdi kjøres en avbruddsfunksjon (interrupt routine). Det er ikke ønskelig at en slik avbruddsfunksjon kjører for ofte, da det vil gi et tregere program. Samtidig er det ønskelig i dette tilfellet at avbruddsfunksjonen blir kjørt ofte nok til at styringen blir presis.

Nedskalering med 64 gjør at timer2 teller med en frekvens på 250 KHz. Ved å sette en avbruddsfunksjon hver gang timer2 når 250, kjøres det en avbrytelse 1000 ganger i sekundet. Det gjør at vi kan få et styringssignal med en nøyaktighet +- ett millisekund. Sammenligningsregisteret som anngir verdien timer2 må nå før avbruddsfunksjonen startes, må settes lavere enn 256, for runde tall er den satt til 250. Nøyaktighet på millisekundet er mer presist enn hva som er nødvendig i dette systemet, men da det ikke fører til noen merkbar treghet i systemet og det ville være tidkrevende å bytte til timer1 ble det avgjort å beholde timer2 som tidtaker.

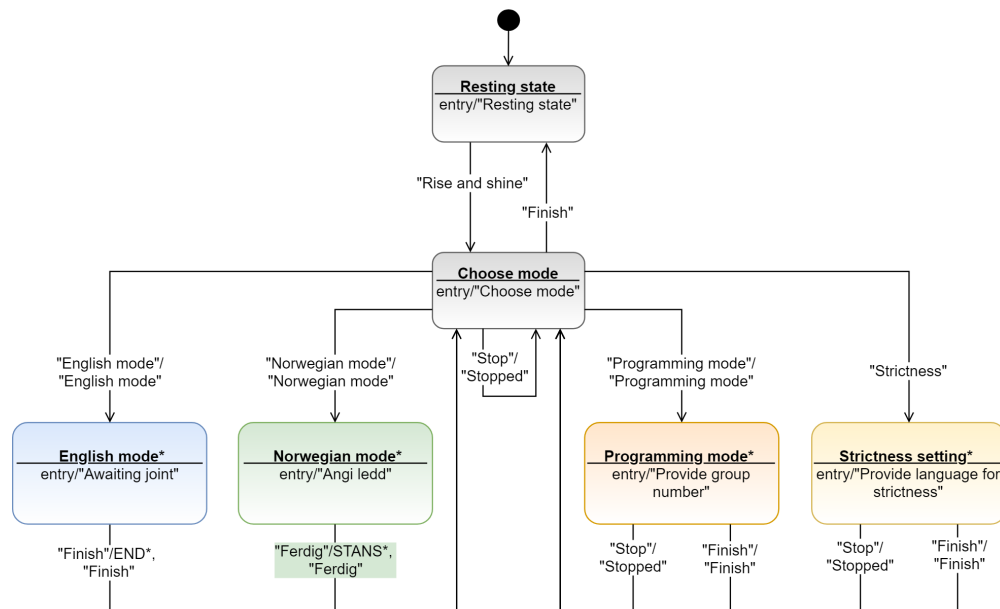
4.3.1 Stemmestyringsprogrammets moduser

Basert på analyse av utfordringer og praktiske løsninger ved stemmestyring av en motorisert armortose, ble enhetens fire moduser planlagt. En hvilemodus hvor triggerkommandoen for å vekke enheten har flere stavelser og er noe en sjelden sier i dagligtale. Engelskmodus med SI-kommandoer for å styre ortosen uten å behøve opptrening av kommandoer. En norskmodus i tilsvarende oppbygning som engelskmodusen, men med SD-kommandoer på norsk som må trenes opp. En programmeringsmodus for at pasienten selv skal kunne trene opp norskmodusen

KAPITTEL 4. UTVIKLING AV STEMMESTYRINGSSYSTEMET

uten å måtte koble til datamaskin. “Strictness setting” for å la pasienten kunne justere talegjennkjennings sensitiviteten basert på omgivelsene ortosen brukes i og egne brukererfaringer om hva som er optimal sensitivitetsinnstilling for pasienten.

Den endelige arkitekturen til stemmestyringssystemets brukergrensesnitt, med to starttilstander og fire moduser er vist i UML-tilstandsdiagrammet i figur 4.4. Stemmestyringens bruksanvisning finnes i vedlegg A. For å gjøre det enkelt å finne frem i bruksanvisning og testresultater er modusene fargekodet. Den indre strukturen til norskmodus vises i figur 4.5. Alle boksene med stjerne har en indre struktur. De resterende modusene sine tilstandsdiagrammer finnes bruksanvisningen, tilstandsdiagram for “english mode” i figur A.3, “programming mode” i figur A.5 og “strictness setting” i figur A.7.



Figur 4.4: Oversikt over stemmestyringsprogrammets overordnede arkitektur. Boksene med stjerner har en indre struktur.

På transisjonspilene mellom tilstandene står kommandoer som fører til ny tilstand. I tilstandsboksene står tilstandens indre aktiviteter. Teksten i diagrammet har formatet

“hendelse[transisjonsbetingelse]/aksjon”. [9] En aksjon kan være å starte motorene eller å gi en tilbakemelding over høyttaleren. Ofte vil en av de tre, “hendelse[transisjonsbetingelse]/aksjon”, være oppført alene. Slik som [timeout] eller “Albue”, men en aksjon vil aldri inntreffe uten en hendelse. En transisjonsbetingelse på en transisjonslinje, forhindrer at transisjonen kan gjøres før betingelsen er sann. Når aksjonen er en auditiv tilbakemelding og når hendelsen er en kom-

mando som skal sies, vil det stå i anførselstegn. Eksempelvis **“Stop”/“Stopped”**, hvor “Stop” er hendelsen og “Stopped” er aksjonen. Hendelsen **entry**/ optrer når enheten går inn i en ny tilstand, denne hendelsen fører ofte til en tilbakemelding over høyttaleren, slik som **entry/“Angi ledd”**.

Figur 4.5 viser den indre strukturen til “Norwegian mode” som vist i det overordnede arkitekturbilde i figur 4.4. Figuren viser brukergrensersnittet til stemmestyriingsenhetens norskmodus i et tilstandsdiagram, inkludert er starttilstandene som viser hvordan det navigeres frem til norskmodus fra hviletilstand. Under diagrammet vises den motoriserte ortosens tilstander og hvordan de korresponderer med navigasjonen av stemmestyriingsprogrammet. I figuren er det noen transisjonslinjer som går fra norskmodus som helhet, til starttilstandene. Dette vil si at kommandoen på transisjonslinjen kan sies fra alle tilstandene i norskmodus og føre til den samme transisjonen. I bruksanvisningen i vedlegg A er tilsvarende tilstandsdiagramer for stemmestyriingsprogrammets andre moduser med samsvarende farger som i oversiktsbildet, “english mode” i figur A.3, “programming mode” i figur A.5 og “strictness setting” i figur A.7.

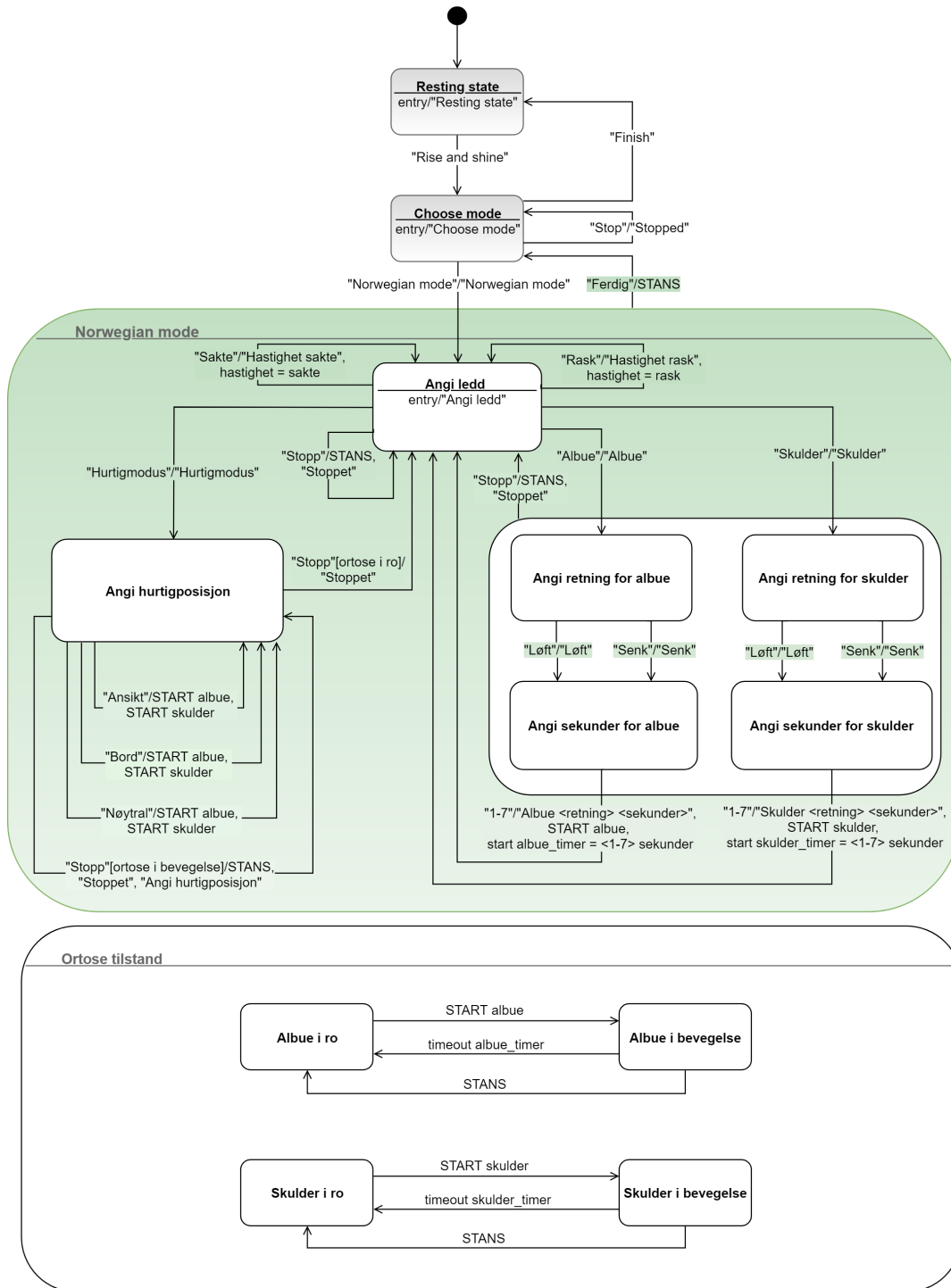
4.3.2 Brukereksempel av norskmodus

Brukereksempel av norskmodus er vist i figur 4.6 og 4.7. Brukereksemplene viser bruk av norskmodus sine to bevegelsesmuligheter, hurtigmodus og ordinær bevegelse. Det kan være forvirrende at hurtigmodus er en modus i norskmodus. Den opprinnelige planen var å kunne navigere til hurtigmodus fra “choose mode” tilstanden. Da enheten viste sårbarhet for falskt positive gjenkjennelser, ble hurtigmodus flyttet for å øke sikkerheten mot at falskt positive gjenkjennelser skulle kunne starte en hurtigbevegelse.

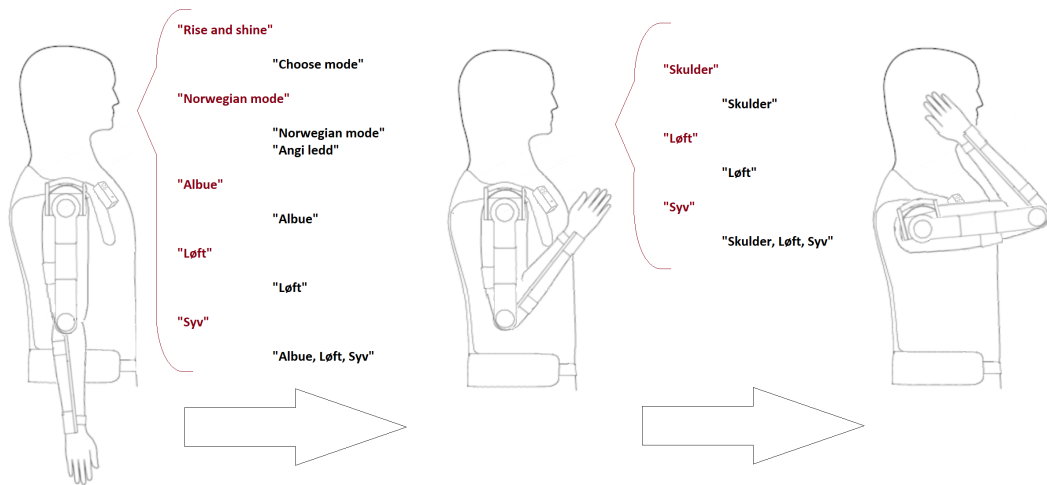
Ved start av brukereksemplene er stemmestyriingsenheten i hvilemodus. Kommandoer sagt av brukeren er i rød skrift og tilbakemeldinger over stemmestyriingsenhetens høyttaler er i svart skrift. Hurtigmodus krever kun én kommando for å starte bevegelse, mens ordinær bevegelse krever tre kommandoer for å starte bevegelse. I brukereksempelet ønsker brukeren å heve armen fra nøytral utstrakt posisjon, til ansiktet.

I figur 4.6 vekker brukeren stemmestyriingsystemet ved å si “Rise and shine”. Deretter gis kommandoene “norwegian mode”, “albue”, “løft” og “syv”. For hver kommando som sies responderer enheten ved å spille et passende lydklipp over enhetens høyttaler. Før bevegelsen startes gjentar enheten den valgte bevegelsen, deretter gir et pådrag på albuemotoren slik at underarmen løftes.

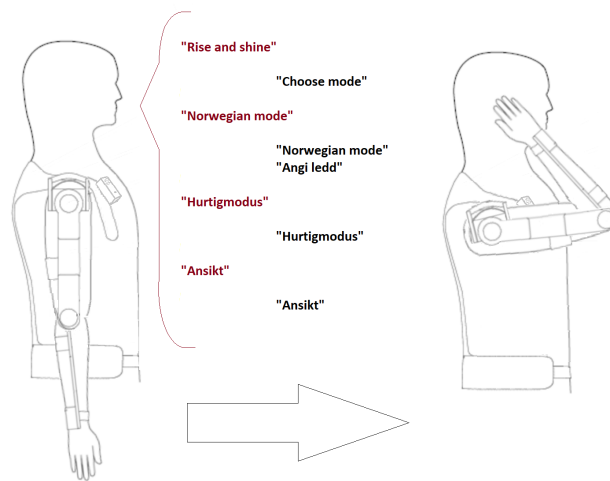
KAPITTEL 4. UTVIKLING AV STEMMESTYRINGSYSTEMET



Figur 4.5: Oversikt over kommandoene mellom norskmodusens ulike tilstander.



Figur 4.6: Brukereksempel for bevegelse av arm fra nøytral utstrakt posisjon til ansikt ved bruk av ordinær norskmodus.



Figur 4.7: Brukereksempel for bevegelse av arm fra nøytral utstrakt posisjon til ansikt ved bruk av hurtigmodus i norskmodus.

Samtidig som motoren startes, starter en teller som etter 7 sekunder vil stanse motorene ved en avbruddsfunksjon. Pådrag på skuldermotoren kan startes før albuemotoren er stanset, styring av de to motorene foregår uavhengig av hverandre. For å bevege skulderen sier brukeren kommandoene "skulder", "løft" og "syv". Etter pådrag på skuldermotoren i syv sekunder er armen ført til ansiket.

I figur 4.7 er brukereksempel fra norsk hurtigmodus. Enheten vekkes med “Rise and shine”, deretter sies kommandoene “norwegian mode”, “hurtigmodus” og “Face”. Både skulder og albuemotor startes og beveger armen mot ansiktet.

4.3.3 Planlegging av system testing

Korresponderende testing til arkitekturdesign i V-modellen er kalt system testing, det vil gjøres i form av en Factory Acceptance Test (FAT).[13] En FAT utføres i kontrollerte omgivelser der produktet er utviklet. I denne oppgaven vil FAT-testing av stemmestyriingsenheten utføres ved at 4 personer tester stemmestyriingsenhetens pålitelighet. Resultatene av FAT er vist sammen med resultatene av SAT i delkapittel 6.1.

4.4 Moduldesign

På dette punktet er produktets ønskede spesifikasjoner kartlagt, implementeringsteknologi er valgt og stemmestyriingsprogrammets arkitektur er planlagt. I dette steget planlegges modulenes design. Modulene som trengs for å bygge stemmestyriingsprogrammet er en fil med setup av kommunikasjon mellom stemmestyriingsporgrammet og talegjenkjenningsprogrammet, en fil med tilstandsmaskin hvor den essensielle logikken er, og en modull for stemmetyriingsenhetens utgangssignaler. Utover det vil det lages separate funksjoner som programmerer SD-kommandoer, og som endrer talegjenkjennings-sensitivitet. To andre viktige komponenter som må utarbeides for stemmestyriingsenheten, er lydklipp for tilbakemeldinger og talegjenkjenningskommandoene.

4.4.1 Stemmestyriingsprogrammets moduler

Setup og tilstandsmaskin skal utforme programmets essensielle logikk og utgangssignal-modulen skal kunne endres uten at det påvirker den essensielle logikken. Det kan være ønskelig å endre utgangssignal-modulen i fremtiden for eksempel ved å sende tilbakemeldinger til et display i stedet for over høyttaler. Det kan også være ønskelig å sende et annet styringssignal.

Inngangssignalet er kommandoer registrert med mikrofonen og gjenkjent av talegjenkjenningsprogrammet. Den gjenkjente kommandoens indeksnummer hentes med en innebygget EasyVR funksjon i setup filen og sendes til tilstandsmaskin-filen. Ved å motta en indeksverdier som til-

svarer den samme hensikten fra andre kilder, er målet at styringssystemet skal kunne fungere på tilsvarende måte.

Enheten støtter, som forklart i delkapittel 4.2.1, både personavhengige (SI) og -uavhengige (SD) kommandoer. SD-kommandoene læres opp av systemet ved lydklipp, så det er vilkårlig hvilket språk disse kommandoene lages i. SI-kommando kan produseres syntetisk med “tekst til SI-kommando” dataprogrammet QuickT2SI. Enheten har et knippe SI-kommandoer allerede lagt inn på EasyVR-modulen. For å lage nye kommandoer som ikke er personavhengige ble det kjøpt lisens til dataprogrammet QuickT2SI.

4.4.2 Lydklipp og talegjennkjenningskommandoer

Kommandoene brukt i stemmestyringssystemet ble utviklet parallellt med at systemet ble utviklet. Testing underveis viste hvilke kommandoer som presterte bedre enn andre. Tilsvarende ble også lydklippene brukt til tilbakemeldinger endret underveis. Lydklippene benyttet for å gi tilbakemeldinger over høyttaleren, består av 86 opptak av min stemme, dette utgjør hele stemmestyringsenheten sitt vokabular. Opptakene er gjort med dataprogrammet Audacity. For å minimere antallet filer, er setninger med små variasjoner delt opp i deler som settes sammen for å imitere flytende tekst. Lydklippene som skal legges inn på stemmestyringsenheten må være monofone lydfiler (mono) og av filtypen WAV, og komprimeres med Sensory Inc. sitt dataprogram “Quick Synthesis 5”. Deretter kan lydfilene overføres til EasyVR-modulen med EasyVR Commander. [17]

Stemmestyringssystemet inneholder 83 kommandoer fordelt på 14 grupper. 58 engelske kommandoer og 25 norske kommandoer. Enheten lytter kun etter kommandoene i én gruppe om gangen. For å begrense oppgaven med å trene opp systemets norske kommandoer, er det blitt forsøkt å holde antallet norske kommandoer lavt. Antallet norske kommandoer er holdt lavt ved at ønsket bevegelse velges ved å først velge ledd, så retning og til slutt varighet. Dette reduserer antallet nødvendige kommandoer drastisk fra å ha én lang kommando for hver bevegelse. I stedet for å lytte etter 28 alternative kommandoer på en gang, lytter enheten etter 11 kommandoer fordelt på tre grupper. Denne løsningen øker også ulikheten mellom kommandoene og dermed reduserer risikoen for falskt positive registreringer. Kommandoene “Løft” og “Senk” har færre likheter enn “Løft, albue, syv” og “Senk, albue, syv”. Under innledende testing viste det seg at kommandoer som var for like ble vanskelige å trene opp i norskmodus. Enheten ga feilmelding

om at kommandoen som ble sagt var for lik en annen kommando og at den måtte trenes på nytt. For like kommandoer viste seg også å være et problem i engelskmodus da kommandoene fort ble forvekslet av talegjenkjenningseenheten.

En begrensning for systemet er at stemmegjenkjenningen kun kan lytte etter 12 ord om gangen på engelsk og kun 32 ord totalt på norsk. Ettersom programmet ble utviklet var det nødvendig å fjerne kommandoer eller slå kommandoer sammen for å oppnå samme funksjonalitet innenfor systemets begrensning. “Hjelp” eller “assistanse” som tidligere ble brukt for å gi brukeren muligheten til å høre alle alternative kommandoer i den gitte tilstanden ble slått sammen med stopp. I den reviderte versjonen av koden sies systemets tilstand over høyttaleren etter kommandoen stopp er gitt, i tillegg til å stoppe all bevegelse og gi tilbakemeldingen “stoppet”.

Å velge gode kommandoer er en krevende prosess, egnethet og funksjonalitet må sjekkes ut for hvert ord. Kommandoene har blitt forandret og tilpasset basert på observasjoner av stemmestyringen i bruk. Kommandoer ble endret enten fordi de ble forvekslet av talegjenkjenningen, at de var vanskelig å trene opp, at de var unaturlige å si eller at de sjeldent ble gjenkjent. Systemets endelige kommandoer er valgt i kompromiss mellom å gjøre ordene ulike og lett for talegjenkjenningsprogrammet å tolke, og samtidig ord som er naturlige å bruke og enkle å huske.

4.5 Modulspesifikasjon og koding

EasyVR Commander er dataprogrammet som benyttes for å legge inn SI- og SD-kommandoer som skal kunne gjenkjennes av enheten. Dataprogrammet kan, utfra kommandoer som er lastet over til EasyVR-kortet, opprette en fil for Arduinoen med filutvidelsen “.ino”. Filen inneholder et skall i programmeringsspråket C for talegjenkjenning av kommandoer lastet over til EasyVR-modulen. Arduino-filen inneholder oppsett av kommunikasjon til talegjenkjenningseenheten og en loop hvor det lyttes etter kommandoer. Når en kommando blir gjenkjent kalles en funksjon hvor ønsket innhold kan legges inn. Denne filen dannet starten på stemmestyringsprogrammet og ble navngitt “setupEasyVR.ino”. Programmeringsspråket valgt til å utvikle stemmestyringssystemet ble “C”. C er et populært språk for profesjonell programmering og har alle funksjonene nødvendig for oppgaven. C har trekk av høynivåprogrammering, som gir kompakt og lettlest kode og samtidig har det lavnivåkomponenter som gir mulighet til å jobbe med registre og tellere.

Stemmestyringsprogrammet kan oppstykket i fem deler. En del er filen **SetupEasyVR.ino**, som inneholder setup og talegjenkjenningsloop. Her er også deklarerering av variabler og navngivning

med data typen enum av tilstander, kommandoer, variabler og lydklipp. Filen inneholder også avbruddsfunksjonen som stanser motorbevegelse etter tidperioden angitt med talekommando av brukeren av stemmestyringsenheten. Denne avbruddsfunksjonen måtte stå utenfor loopen, da avspilling av lydklipp og gjenkjenning av tale okkuperer prosessoren. Denne avbruddsfunksjonen kan ikke kalles noen sekund for sent, derfor er den plassert utenfor loopen hvor den vil kjøre akkurat når timeren når 250. I loopen lyttes det etter en kommando frem til en kommando er registrert. Som anbefalt i databladet, har lytting etter kommando timeout hvert 5. sekund. Når en kommando registreres settes variabelen “registredCommand_index” til indexnummeret for kommandoen som gjenkjennes. Dersom ingen kommando gjenkjennes eller det oppstår en feil, settes “registredCommand_index” til -1 og loopen kjører på ny. Pseudokode for den ferdig utviklede setupEasyVR.ino er vist i figur 4.8 hvor omtrent 500 linjer kode er destillert til 30.

```

SetupEasyVR.ino

Declarations and initialisations of variables.

Names given to relevant states, commands, variables and soundclips. Example:

enum EnglishWordsets{
  PROVIDE_JOINT_STATE = -6,
  PROVIDE_DIRECTION_STATE = -7,
  PROVIDE_SECONDS_STATE = -8, |
};

enum ProvideJoint_Wordset6 {
  S6_STOP = 0,
  S6_FINISH = 1,
  S6_ELBOW = 2,
  S6_SHOULDER = 3,
  S6_RAPIDMODE = 4,
  S6_QUICKSPEED = 5,
  S6_SLOWSPEED = 6,
};

void setup() {
  setup of board
}

// timer compare interrupt service routine, interrupt runs when Counter2 reaches OCR2A
ISR(TIMER2_COMPA_vect) {
  counterShoulder++;
  counterElbow++;
  if (counterShoulder == secondsShoulder) setOutputsignal(STOP, ELBOW);
  if (counterElbow == secondsElbow) setOutputsignal(STOP, SHOULDER);
}

void loop(){
  -Listen for command
  if command heard{
    registeredCommand_index = index number of the command heard
    stateMachine(); //run state machine events for the current state and command index
  }
}

```

Figur 4.8: Pseudokode som forklarer innholdet i filen SetupEasyVR.ino.

I kodeeksemplene inngår det funksjoner fra EasyVR-modulen. Disse har funksjonsnavn som starter med “easyvr.”. Funksjonenes virkemåte og informasjonen som de enkelte funksjonene

KAPITTEL 4. UTVIKLING AV STEMMESTYRINGSYSTEMET

trenger og informasjonen som gis tilbake er kjent fra databladet. Den strukturelle oppbygningen av funksjonene er imidlertid ikke kjent for programmereren.

En annen del er filen **stateMachine.ino**, inneholder funksjonen “void stateMachine();”, her finnes den essensielle logikken som utføres når en kommando blir gjenkjent. Funksjonen består av en overordnet switch som håndterer tilstandene, switch(state), hvor hver tilstand er en case. I hver tilstand er det en underordnet switch, switch (registeredCommand_index), som har en case for hver de mulige kommandoene i tilstanden. Når en kommando i enhetens tilstand registreres, settes “registeredCommand_index” til kommandoens indeksnummer og tilhørende case i “stateMachine();” utføres, før programmet går videre i loopen i SetupEasyVR.ino og lytter etter en ny kommando. Et utdrag av koden “stateMachine.ino” er vist i figur 4.9. Inngangssignalet til stemmestyringsprogrammet er i form av indeksnummeret til en kommando. Kilden for indeksnummeret kan endres uten å behøve strukturelle endringer i stemmestyringskoden, annet enn i “SetupEasyVR.ino”-filen.

```
stateMachine.ino

void stateMachine() {
  switch (state) { //The states correspond to the command groups
    -There is one case for each state. Example:
    //States in norwegian mode and english mode are combined, messages are translated to norwegian if norwegian mode is chosen.
    case PROVIDE_JOINT_STATE: case ANGL_LEDD_TILSTAND:
      switch (registeredCommand_index) {
        case S6_STOP: //og norsk: STOPP //There is a case for every possible command in the command group.
          setOutputSignal(ELBOW, STOP); //Stop movement
          setOutputSignal(SHOULDER, STOP);
          playMessage(STOPPED); //Play suited feedback message over speaker
          playMessage(AWAITING_JOINT);
          break;

        case S6_FINISH: //og norsk: FERDIG
          setOutputSignal(SHOULDER, STOP);
          setOutputSignal(ELBOW, STOP);
          playMessage(FINISH);
          playMessage(CHOOSE_MODE);
          state = CHOOSE_MODE_STATE; //Transition to another state
          break;

        case S6_ELBOW: //og norsk: ALBUE
          playMessage(ELBOW);
          Joint=ELBOW;
          state = PROVIDE_DIRECTION_STATE;
          break;

        case S6_SHOULDER: //og norsk: SKULDER
          playMessage(SHOULDER);
          Joint=SHOULDER;
```

Figur 4.9: Kodeutdrag fra filen StateMachine.ino.

En tredje del er filene for utsignaler, **setOutputSignal.ino** og **playMessage.ino**. Førstnevnte sender angitt hastighets og retningssignal på utgang 9 for albue eller 10 for skulder med funksjonen “analogWrite(outputPin, outputValue);”. Formen på utgangssignalet utdypes i neste kapittel, 5 om grensersnittet mellom stemmestyringsenheten og den motoriserte ortosen. Funksjonene i “playMessage.ino” sender lydklipp til enhetens høyttaler. Begge disse funksjonene kan endre formen på utsignalet uten at det vil påvirke tilstandsmaskinen.

Fjerde del består av filen som endrer talegjenkjenningssensitivitet, **setStrictness.ino**. Denne delen justerer talegjenkjenningssensitiviteten etter brukerens kommando og lagrer verdien frem til sensitivitet blir endret på ny. Om enheten kobles fra strøm vil dette allikevel være sensitivitetsverdien enheten er satt til når den kobles til igjen. Sensitivitet angis separat for SD og SI kommandoer, eller norske og engelske kommandoer som det er blitt i praksis. Forenklet kode er vist i figur 4.10

```
bool setStrictness(the chosen language, the chosen value){
    switch (language){
        case ENGLISH_LANGUAGE:
            //Knob is zero indexed, therefore value-1. bool easyvr.setKnob(value) returns true if new value is set
            if (easyvr.setKnob(value-1)){
                EEPROM.write(addr_english, value);
                strictness_ENG = EEPROM.read(addr_english);
                return true;
            }
            else return false;
            break;

        case NORWEGIAN_LANGUAGE:
            if (easyvr.setLevel(value)){ // bool easyvr.setLevel(int strictnes_setting)
                EEPROM.write(addr_norwegian, value);
                strictness_NOR = EEPROM.read(addr_norwegian);
                return true;
            }
            else return false;
            break;

        default:
            return false;
    }
}
```

Figur 4.10: Forenklet kode fra funksjonen som endrer talegjenkjenningssensitivitet.

KAPITTEL 4. UTVIKLING AV STEMMESTYRINGSYSTEMET

Sistnevnte del består av funksjonene som trener norskmodusens SD-kommandoer. “void setGroupIndexForTraining(int input)” og “void runCommandTraining()”. Førstnevnte funksjon setter gruppe og indeksnummer for kommandoen som skal trenes, utfra tallkommandoer som gies av brukeren. Så avspilles et lydklipp for å bekrefte om den riktige kommandoen er valgt. Slik som “Train command for” “Ferdig”. Hvis brukeren sier ja, kalles den sistnevnte funksjonen, som trener den angitte kommandoen. Denne delen er navngitt **commandTraining.ino**. Forenklet kode for denne funksjonen kan leses i figur 4.11.

```
void runCommandTraining(){
    easyvr.eraseCommand(groupNumber, indexNumber);
    Label_Start_of_Training: //This will prevent ending of training prematurely (when train < 3, even though trainingCount is < 6)
    playMessage(SAY_NEW_COMMAND);

    while (number of successful trainings < 3 and number of trials < 6){ //Keep training until 3 succesfull trainings or untill 6 tries.
        easyvr.trainCommand(groupNumber, indexNumber);

        while(easyvr.hasFinished() has not finished and wait < 10){ //Keep listening until a phrase is registered or timeout
            delay(250);
            digitalWrite(pinLED, HIGH); // Flash led when listening for a phrase
            delay(250);
            digitalWrite(pinLED, LOW);
            wait++;
        }

        if (conflict occurs){ //Saved trainings should be erased if the recognized phrase is too similar to another command
            easyvr.stop();
            easyvr.eraseCommand(groupNumber, indexNumber);
            playMessage(FAILED);
            -reset number of succesfull trainings
        }

        else if (easyvr.getError()==-1){ // if no error occured a new training is saved. (easyvr.getError()==-1 if no error occured)
            -increment number of succesfull trainings
        }
        wait=0;
        -increment number of trials

        if (number of succesfull trainings < 3 and number of trials < 6){
            playMessage(SAY_NEW_COMMAND_ONCE_MORE);
        }
        else if (number of succesfull trainings < 3 and number of trials == 6){
            playMessage(TRAINING_FAILED);
        }
    } // End While

    if (easyvr.dumpCommand(groupNumber, indexNumber, name, train)){ //check if training indeed is complete
        if (number of succesfull trainings <= 2 and number of trials < 6){ //if training not complete
            number of succesfull trainings = train;
            goto Label_Start_of_Training; //Tran some more
        } //end if training not complete

        else if (number of succesfull trainings > 1 and number of trials == 6, or number of succesfull trainings == 3){
            playMessage(NEW_COMMAND_SAVED);
            keyWordLetter = numberToKeyword(train); //Simple function to translate number into the corresponding voiceclip
            playMessage(keyWordLetter);
            playMessage(SUCCESSFUL_TRAININGS);
        }
    } //end check if training complete

    playMessage(PROVIDE_INDEX_NUMBER);
    state = PROGRAMMING_STATE;
} // End function CommandProgramming
```

Figur 4.11: Forenklet kode fra funksjonen som trener angitt kommando i commandTraining.ino.

5 Grensesnitt mellom ortose og stemmestyringsenhet

Grensesnittet mellom stemmestyringsenheten og ortosen ble utarbeidet etter samtaler med oppdragsgiver om hvert delsystem sine muligheter og begrensninger. Stemmestyringsenheten ble utviklet med foroverkobling til motorstyreren. Mangel på tilbakekobling gir noen begrensninger for stemmestyringen, samtidig gir foroverkobling et enkelt system som er tilstrekkelig for styring av dette systemet. Mulighet for tilbakekobling ble vurdert, men det viste seg for tidkrevende, da det ville kreve både en strukturell endring og en reprogrammering av motorstyreren. Tilbakekobling kan imidlertid legges inn på et senere tidspunkt, hvor informasjon om ortosens stilling og motorenes hastighet kan hentes fra motorenes Hall signaler.

5.1 Styringssignal sendt ved pulsbreddemodulasjon

Ortosens motorer styres av et spenningssignal sendt fra stemmestyringsenheten til motorstyreren, som beskrevet i delkapittel 3.2.1 om motorstyreren. Stemmestyringsenheten ble programmert til å sende samme type styringssignal til ortosen som fotspakene, for informasjon om fotstyring se delkapittel 3.3.3. Med denne løsningen kan man etter behov skifte styringsmetode ved å koble ledningen som overfører styringssignalet fra fotpedal til stemmestyring.

For å styre den motoriserte ortosen sendes to styringssignaler fra stemmestyringsenheten. Signalene angir ønsket pådrag og pådragsretning for hver av motorene. Arduino Uno kan kun sende digitale signaler og motorstyreren trenger å motta analoge spenningssignaler. Dette fordi inngangssignalet til motorstyreren samples med en analog til digital omformer (AD-omformer).

Styringssignalet sendes fra stemmestyringsenheten som to digitale pulsbreddemodulerte (PWM) signaler, som gattes med et lavpassfilter for å imitere et analogt signal.

PWM-signalene sendes fra Arduinoens digitale utganger 9 og 10. Digitale utganger kan enten sende høyt eller lavt signal. Arduino Uno sine utsignaler har en spenningsverdi på 5 V, det betyr at lavt signal gir 0 volt og høyt signal gir 5 volt.[24] Ved bruk av PWM kan ulike spenningsverdier imiteres med digitalt signal ved å veksle mellom å sende høyt og lavt signal med ulik ratio mellom 0 og 5 V. Når kontinuerlig høyt signal tilsvarer 5 V, så vil et signal som veksler mellom å være 50% på og 50% av, ha en tilsynelatende spenningsverdi på 2,5 V.

Utgang 9 og 10 er 8 bits utganger, som vil si at spenningsnivået kan varieres med $2^8 - 1 = 255$ nivåer. På Arduinoen sendes et PWM signal med den innebygde funksjonen *analogOut(bitValue)*; med bit-verdi mellom 0 og 255, som fører til et utsignal tilsvarende 0-5 V. 255 nivåer hvor $255 = 5V$, gir konverteringsformel $XV_{olt} \cdot \frac{2^8 bits - 1}{5V_{olt}} = Ybits$.

5.2 Filtrering av styringssignalet

For å videreformidle den korrekte spenningsverdien til motorstyreren, må PWM-signalet gattes tilstrekkelig før det når motorstyrerens AD-omformer. Et første ordens RC-filter er en enkel og rimelig måte å lavpassfiltrere PWM signalet. En skisse av et RC-filter koblet mellom stemmestyringsenheten og motorstyreren er vist i figur 5.1. Signalets spenning vil deles over motstandenes impedans og kondensatorens reaktans. Motstanden i kondensatoren, reaktansen, avtar med økende frekvens. Når det er brå endringer i signalet, høye frekvenser, vil kondensatorens reaktans avta. Sett fra motorstyrerens signalinngang vil dette dempe de høye frekvensene, ved at mer av signalets spenning står over kondensatoren ved høye frekvenser. Når PWM signalet er høyt vil kondensatoren lades opp og når PWM signalet er lavt vil kondensatoren lades ut, slik demper lavpassfilteret spenningsverdien til PWM signalets topper og hever signalets bunner.

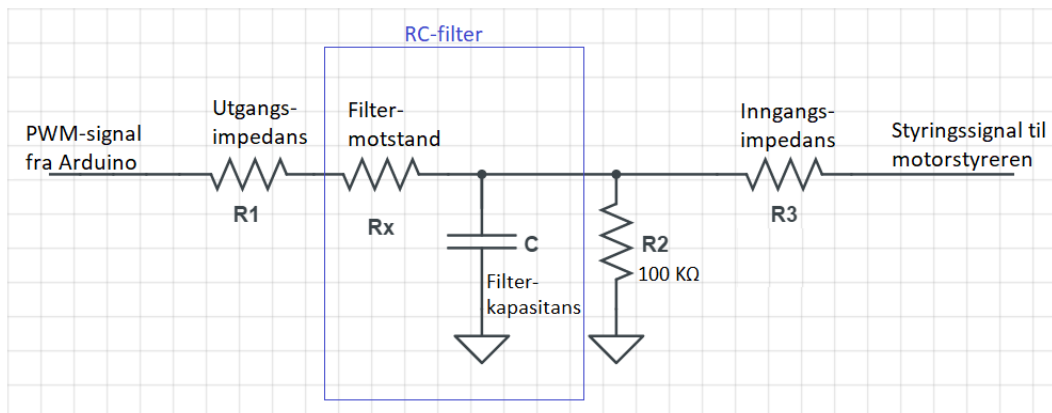
5.2.1 Valg av filterkonstanter

Filterkonstanter må velges etter systemets krav for responstid og glatthet på styringssignal. Her må det gjøres et kompromiss da større resistans og kapasitans i filteret gir tregere respons, men et jevnere signal. Både avsender og mottaker av styringssignalet er en ATmega168 mikrokontroller. Utgangsimpedansen fra ATmega168 er omtrent $25 - 40\Omega$ og inngangsimpedansen til

AD-omformer er omtrent $100M\Omega$. [25] Motorstyrerkrets-kortet har i tillegg en ledningsbrudd-kring mot jord på $100k\Omega$, navngitt R2 i filterskissen. ATmega168 er designet for at et analogt innsignal skal ha en utgangsimpedans på mindre eller lik $10k\Omega$. [25] Utgangsimpedansen sett fra motorstyreren i figur 5.1 er gitt ved formel for parallellkoblede motstander som ligning 5.1.

$$\frac{(R_1 + R_x) \cdot R_2}{R_1 + R_x + R_2} \leq 10k\Omega \quad (5.1)$$

Filterverdiene $1\mu F$ og $10k\Omega$ er angitt som tilstrekkelige verdier til glatting av PWM signal sendt fra Arduino Uno. [26] Utfra filterkomponentene som var tilgjengelig synes $10\mu F$ og $10k\Omega$ å være tilfredsstillende verdier for filteret som skulle bygges. $10\mu F$ er en høy kondensatorverdi som vil føre til et filter som glatter signalet i stor grad, men som vil være litt tregt. Denne tregheten vil være akseptabel i dette systemet da stemmestyring er et styringsprinsipp som ikke krever umiddelbar respons. Samtidig har motordriverne i systemet en tidskonstant på ca. 1 sekund for å øke pådragssignal til motoren fra stillestående til maksimal hastighet, så filterets forsinkelse ved disse verdiene vil ikke være av betydning.



Figur 5.1: Skisse av RC-filteret som glatter PWM-styringssignalet fra stemmestyringsenheten

Når filtermotstanden skal velges i størrelsesorden $10k\Omega$, vil utgangsimpedansen som står i serie med filtermotstanden være så mye lavere at den kan neglisjeres ved utregning av utgangsimpedans. En filtermotstand på $10k\Omega$ vil tilsvare en utgangsimpedans sett fra motorstyreren på omlag $9,09k\Omega$ som vist i ligning 5.2. Denne utgangsimpedansen er innenfor anbefalt verdi for innsignal til mikrokontrolleren.

$$\frac{10k\Omega \cdot 100k\Omega}{10k\Omega + 100k\Omega} = 9,09k\Omega \quad (5.2)$$

Motorstyrerens ekseveringstid er målt til 0,144 ms ved drift av begge motorene. AD-omformerens i motorstyreren vil taste det glattede PWM signalet ved samplingsintervall på $T = 0,144ms$, dette tilsvarer en samplingsfrekvens gitt ved ligning 5.3. Denne samplingsfrekvensen er regnet ut etter ekseveringstid ved maksimal belastning av programmet, dermed kan dette regnes som AD-omformerens sin minimale samplingsfrekvensen.

$$f_s = \frac{1}{0,144ms} = 6,94kHz \quad (5.3)$$

Nyquist sitt samplingsteorem sier at samplingsfrekvensen må være mer enn det dobbelte av den høyeste frekvens i signalet som skal samples for å kunne gjenskape signalet og unngå aliasing.[27] I praksis blir dette å legge filterets knekkfrekvens lavere enn halvparten av minimal samplingsfrekvens.

$$f_{max} \leq \frac{6,94kHz}{2} \quad (5.4)$$

Et ideelt lavpassfilter vil filtrere ut alle frekvenser over filterets knekkfrekvens. I praksis er ikke en så brå overgang mulig, i stedet angir knekkfrekvensen der amplituderresponsen dempes til $\frac{1}{\sqrt{2}}$ av maksimal amplitude. Frekvenser høyere enn knekkfrekvensen vil dempes i stadig større grad. Filterets knekkfrekvens er gitt av ligning 5.5, hvor τ er filterets tidskonstant. Tidskonstanten angir hvor lang tid det tar fra et pådrag sendes som PWM fra stemmestyringsenheten til det filtrerte signalet når 63% av stasjonærverdien til pådraget.[28] I en RC-krets er tidskonstanten gitt ved ligning 5.6. R og C er kretsens ekvivalentmotstand og kondensatorens kapasitans.

$$f_c = \frac{1}{2 \cdot \pi \cdot \tau} \quad (5.5)$$

$$\tau = R \cdot C \quad (5.6)$$

Når alle kretsens motstander slås sammen til en ekvivalentmotstand, vil den variere svært lite i verdi om inngangs- og utgangsimpedansen neglisjeres. Dette er fordi inngangsimpedansen på $100M\Omega$, som står i parallell med filtermotstanden, er så drastisk mye høyere enn filtermotstanden. Utgangsimpedansen på sin side, står i serie med filtermotstanden og er så mye lavere enn filtermotstanden at den heller ikke vil utgjøre nevneverdig forskjell i verdien til kretsens ekvivalentmotstand. Å se bort fra inngangs- og utgangsimpedansen vil ikke føre til at filterkarakteristikken endrer seg i vesentlig grad, derfor vil disse motstandene for enkelhetsskyld neglisjeres. Ledningsbruddsikringen på $100k\Omega$ vil regnes med da denne vil ha en innvirkning på ekvivalensmotstanden i vesentlig skala.

Ved å sette inn verdien for kapasitans og kretsens ekvivalentmotstand, hvor ekvivalentmotstanden er tilnærmet med utgangsimpedansen fra ligning 5.2, gir det ligning 5.7.

$$\tau = 9,09k\Omega \cdot 10\mu F = 90,90ms \quad (5.7)$$

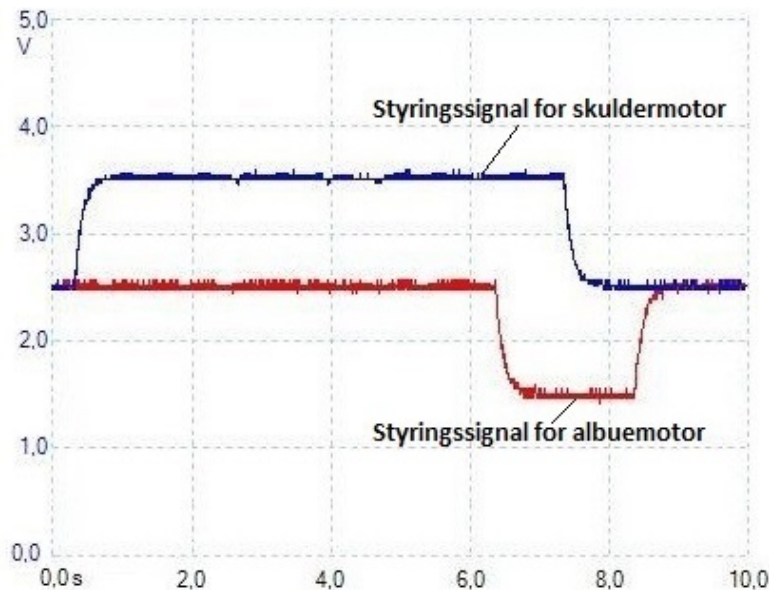
Tidskonstanten satt inn i ligning 5.5 gir knekkfrekvensen 17,51 Hz.

$$f_c = \frac{1}{2 \cdot \pi \cdot 90,90\mu s} = 1,75Hz \quad (5.8)$$

Denne knekkfrekvensen er lav, men sett i forhold til PWM signalets frekvens og foremålet til filtreringen er det passelig. PWM signal sendt fra Arduino Uno sine utganger 9 og 10 er på 490 Hz.[29] Dette svarer til en periode på omtrent 2 ms. Tidskonstanten på 90,90 ms er vesentlig høyere enn PWM signalets periode, det er en forutsetning for å filtrere PWM-signalet til et tilsynelatende analogt signal. Tidsforsinkelsen på tilnærmet ett tiendedels sekund vil ikke merkes i dette systemet. Et lavpassfilter med filterverdiene $10k\Omega$ og $10\mu F$ tilfredstiller kravene til systemet, så filteret utvikles med disse filterverdiene.

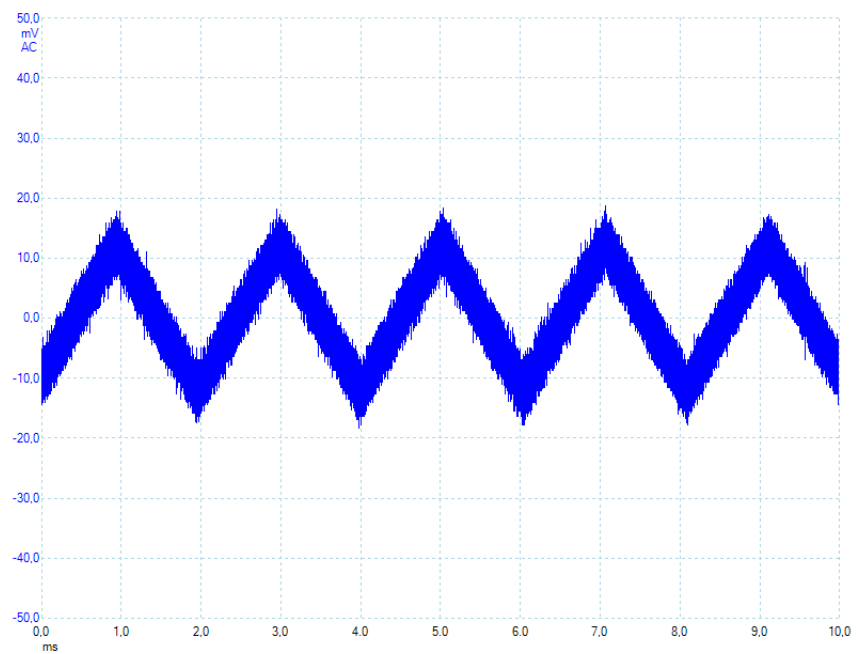
5.2.2 Det ferdige filteret

Filterkomponentene er loddet på et spesialutviklet kretskort laget av Lien. Filteret kan sees avbildet i vedlegg A figur A.8, festet som en bro over EasyVR-modulen. Det filtrerte styringssignalet målt med oscilloskop er avbildet i figur 5.2. Under målingen ble det gitt kommando til stemmestyringsenheten om et 7 sekunder langt pådragssignal med retning opp til skuldermotor, og deretter et to sekunder langt pådragssignal med retning opp til albueomotor. Som nevnt i delkapittel 3.2.1 vil signal over 2,5 V føre til oppoverbevegelse i skulderleddet og signal under 2,5 V vil føre til oppoverbevegelse i albueleddet. Som det fremgår av figuren kan motorene styres uavhengig av hverandre. Ved måling av tiden fra punktet styringssignalet starter frem til oppnådd 63% av stasjonærverdi, stemmer tidskonstanten lest av figuren godt med utregnet tidskonstant.



Figur 5.2: Styringssignal målt med oscilloskop

Ved å forstørre styringssignalet på oscilloskopet, kan vi lese av rippelen til det filtrerte signalet i figur 5.3. Signalets rippel er omtrent 32 mV, som er tilstrekkelig glatt for dette formålet. Motorstyrerens dødbånd er på 0,2 V. Signalets rippel er langt under halvparten av dødbåndet, dermed vil ikke et stoppsignal kunne feiltolkes som et pådragssignal etter sampling av AD-omformerens.



Figur 5.3: Forstørret styringssignal målt med oscilloskop

6 | Stemmestyring i praksis

Den har vært flere runder med testing av stemmestyringsenheten. De første testene kan kalles modultesting, hvor enkelte moduler ble testet til være fungerende. Deretter ble det utført tester av det komplette systemet, hvor alle moduler var integrert. Først av utvikleren av systemet og deretter av frivillige, det er ofte da mangler og feil som ikke utvikleren har tenkt på blir avdekket. Enheten ble også testet av flere individer for å finne gode kommandoer som både var intuitive og presterte godt med ulike brukere. Deretter ble enheten testet av oppdragsgiver og med den motoriserte ortosen, før den ble sendt til pasienten for utprøving.



Figur 6.1: Bilde av den ferdige motoriserte ortosen i bruk med stemmestyringen.

6.1 Testing

Etter mange uhøytidlige tester av stemmestyringsystemet, ble systemet ansett som klart for organisert testing hvor resultater ble registrert. Stemmestyringsenheten ble testet av 4 frivillige og deretter av pasienten. De frivillige testet stemmestyringsenheten uten tilkobling til ortosen, pasienten testet med. Enhetens prestasjon ved ulike innstillinger for talegjenkjenningssensitivitet i engelskmodus, ble testet av utvikleren og er presentert i 6.1.1.

Testene ble utført ved å notere prestasjonen til stemmestyringsens talegjenkjenning, mens testpersonene navigerte stemmestyringsenheten ved å si ulike kommandoer. Hver kommando ble sagt 10 ganger. Sensitivitetsinnstillingen under testingen var satt til standardverdi for 4 av 5 testere, som er 3 for SI-kommandoer og 2 for SD-kommandoer. I figur 6.3-6.7 er resultatene oppført i kolonner for hver av testerne. Engelsk- og norskmodus ble testet av fem personer, mens modusene for programmering og sensitivitetsinnstilling ble testet av tre personer. Engelsk- og norskmodus ble mest grundig testet da dette er de modusene som er ment å brukes aktivt. Startmodusene er inkludert i testingen av engelskmodus.

I figurene er det egne felter for sann positiv gjenkjenning (TP), falskt negativ gjenkjenning (FN), og falskt positiv gjenkjenning (FP1+FP2). Sann positiv vil si at kommandoen som ble sagt ble registrert. Falskt negativ vil si at en kommando ble sagt, men ingen kommando ble registrert. Falskt positiv ble delt i to poster, FP1 og FP2. FP1 vil si at ingen kommando ble sagt, men en kommando ble registrert. FP2 vil si at en kommando ble sagt, men en annen kommando enn den som ble sagt ble registrert.

Antallet falske registreringer (FP1) under testingen vil variere utfra en rekke faktorer, deriblandt hvor lang tid testingen tok for de ulike testpersonene, hvilken tilstand systemet var i de gangene testpersonen tok pause fra å gi kommandoer. Antallet falske registreringer vil også bli påvirket av omgivelsene testingen foregikk i, hvor mye støy det var i rommet, hvor mye prat det var i nærheten av mikrofonen. Allikevel vil antallet gi en grov indikasjon på hvor ofte systemet har falske aktiveringer i de ulike testomgivelsene. Derfor er resultatene inkludert, men de regnes ikke med i noen gjennomsnitt. For å minimere falske aktiveringer, anbefales det at systemet står i "Resting state" eller i norskmodus når brukeren skal ta en pause fra å gi kommandoer. Beskrivelse av omgivelsene ved hver test står under 6.2.

Resultatet av testene av engelsk og norskmodus vises som samlet gjennomsnitt for alle testerne i figur 6.2. Individuelle avvik fra det totale snittet kan sees av enkeltpersonenes gjennomsnitt nederst i hver kolonne av figurene med testresultater. Testerne sine tilbakemeldinger om systemet etter testingen er samlet i underkapittel 6.2. Testingen er utført for å gi en indikasjon på stemmestyringsenhetens egnethet til å styre en motorisert ortose. Å styre et slikt system, krever at talegjenkjenningen er pålitelig. Denne testen vil kunne vise hvilke kommandoer som trekker ned gjennomsnittet av sant positive gjenkjenninger. Den vil vise hvilke kommandoer som typisk blir forvekslet og kan gi en pekepinn på hvordan systemet kan videreutvikles i fremtiden. Testen vil også indikere hvordan enheten presterer for personer med ulike dialekter og engelskferdigheter.

I figur 6.2 angis samlet resultat fra 5 personer sin testing av starttilstander, engelskmodus og norskmodus. Totalt inngår vurdering av registrering for 2846 angitte kommandoer fra de 5 personene i det samlede snittet for modusene.

| | Sann positiv (TP) | Falskt negativ (FN) | Falskt positiv 2 (FP2) |
|--|-------------------|---------------------|------------------------|
| Starttilstandene og engelskmodus (SI-talegjenkjenning) | 89,00% | 7,69% | 3,31% |
| Norskmodus (SD-talegjenkjenning) | 90,41% | 9,43% | 0,16% |
| Starttilstandene, engelsk- og norskmodus | 89,61% | 8,45% | 1,94% |

Figur 6.2: Samlet gjennomsnitt fra testresultatene til de 5 testerne av starttilstandene, engelskmodus og norskmodus.

Testresultatene som danner basis for gjennomsnittsverdiene vises i de følgende figurene. Testresultatene presenteres ved at første kolonne er stemmestyringssystemets tilstander, som tidligere nevnt tilsvarende disse tilstandene gruppene av kommandoer lastet inn på talegjenkjenningsenheten. I kolonnen til høyre for tilstandene er de eneste alternative kommandoene enheten lytter etter, når den er i de ulike tilstandene. I de følgende kolonnene vises testresultatet for hver person etter hver av kommandoene er sagt 10 ganger. Et resultat på “10-0-0”, altså verdi 10 for TP, vil si at alle gangene denne kommandoen ble sagt, ble den registrert korrekt av stemmestyringsenheten. Et resultat på “5-0-5” vil si at fem av de uttalte gangene ble kommandoen registrert riktig, og ved de andre fem gangene ble kommandoen registrert som en annen kommando. I de

KAPITTEL 6. STEMMESTYRING I PRAKSIS

fleste tilfeller står det hvilke kommandoer som ble registrert feil i kolonnen helt til høyre, kolonnen “Registrert feil som”. Denne kolonnen er felles for alle testerne og viser hva kommandoene kan misforståes som av talegjenkjenningen. Under innledende testinger ble kommandoer med et høyt resultat under “registrert feil som”. byttet ut med nye kommandoer. Spørsmålsteget i testresultatet viser manglende data.

| Tilstand | Kommando | Tester: Pasienten | | | Tester 1 (M 29) | | | Tester 2 (K 32) | | | Tester 3 (K 23) | | | Tester 4 (K 27) | | | Registrert feil som |
|--|------------------|-------------------|------|------|-----------------|------|------|-----------------|-------|------|-----------------|-------|------|-----------------|------|----------|---------------------|
| | | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | |
| Resting state | Rise and Shine | 8 | 2 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | |
| | Stop | 8 | 2 | 0 | 7 | 3 | 0 | 7 | 3 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | |
| Choose mode | Finish | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 8 | 2 | 0 | 10 | 0 | 0 | |
| | English mode | 10 | 0 | 0 | 8 | 2 | 0 | 8 | 2 | 0 | 8 | 2 | 0 | 10 | 0 | 0 | |
| | Norwegian mode | 7 | 2 | 1 | 10 | 0 | 0 | 6 | 4 | 0 | 9 | 1 | 0 | 8 | 2 | 0 | 1x? |
| | Programming mode | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Strictness | 10 | 0 | 0 | 10 | 0 | 0 | 8 | 2 | 0 | 7 | 3 | 0 | 10 | 0 | 0 | |
| Awaiting joint | Stop | 9 | 0 | 1 | 7 | 1 | 2 | 8 | 1 | 1 | 9 | 0 | 1 | 10 | 0 | 0 | 1xRM, 2xSS, 1x? |
| | Finish | 8 | 2 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Elbow | 9 | 1 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | |
| | Shoulder | 10 | 0 | 0 | 9 | 1 | 0 | 7 | 3 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Rapid mode (RM) | 9 | 1 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Quick speed (QS) | 10 | 0 | 0 | 9 | 1 | 0 | 8 | 2 | 0 | 7 | 3 | 0 | 8 | 2 | 0 | |
| Awaiting direction | Slow speed (SS) | 10 | 0 | 0 | 6 | 4 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Stop | 6 | 0 | 4 | 8 | 0 | 2 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 4xFive, 2xLower |
| | Finish | 8 | 2 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Lift | 9 | 1 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 5 | 3 | 2 | 10 | 0 | 0 | 2xFinish |
| Awaiting seconds | Lower | 8 | 2 | 0 | 10 | 0 | 0 | 7 | 2 | 1 | 8 | 2 | 0 | 10 | 0 | 0 | 1xLift |
| | Stop | 9 | 1 | 0 | 8 | 0 | 2 | 8 | 0 | 2 | 10 | 0 | 0 | 10 | 0 | 0 | 2xOne, 1xFive |
| | Finish | 8 | 2 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | One | 9 | 1 | 0 | 10 | 0 | 0 | 7 | 0 | 3 | 9 | 1 | 0 | 9 | 1 | 0 | 3xFive, 1xFour |
| | Two | 7 | 2 | 1 | 8 | 0 | 2 | 5 | 0 | 5 | 7 | 0 | 3 | 10 | 0 | 0 | 8xSeven, 3xSix |
| | Three | 9 | 0 | 1 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 1xFive |
| | Four | 10 | 0 | 0 | 10 | 0 | 0 | 8 | 0 | 2 | 4 | 2 | 4 | 10 | 0 | 0 | 5xFive, 1xOne |
| | Five | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| Six | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 6 | 0 | 4 | 9 | 1 | 0 | 4xFinish | |
| Seven | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 4 | 4 | 2 | 10 | 0 | 0 | 2xFive | |
| Awaiting rapid position | Stop | 10 | 0 | 0 | 10 | 0 | 0 | 6 | 4 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | |
| | Finish | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 4 | 1 | 5 | 8 | 0 | 2 | 7xFace |
| | Neutral | 9 | 1 | 0 | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Table | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| Snitt [%] | Face | 8 | 2 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | | 90,0 | 7,50 | 2,50 | 91,25 | 6,25 | 2,50 | 85,0 | 10,63 | 4,38 | 82,5 | 10,94 | 6,56 | 96,25 | 3,13 | 0,63 | |
| Antall falske registreringer under testingen (FP1) | | ? | | | 6 | | | 10 | | | 14 | | | 7 | | | |

Figur 6.3: Testresultater av starttilstandene og engelskmodus som bruker SI-talegjenkjenning.

Som det vises av testresultatene i figurene 6.3, 6.4 og 6.5, er kommandoen “Stop” en av de som gjenkjennes dårligst, og som sporadisk registreres som andre kommandoer. Dette var kjent fra tidligere testinger, men det ble enighet om å beholde kommandoen foreløpig da de alternative kommandoene som ble forsøkt, som stans, avbryt, avslutt og holdt, var unaturlige og flere av

de ga også høye verdier for FN og FP2 ved talegjenkjenning. “Stop” er en av de mest kritiske kommandoene å gjenkjenne riktig, så det bør finnes en løsning på dette i fremtiden.

| Tilstand | Kommando | Tester: Pasienten | | | Tester 1 (M 29) | | | Tester 3 (K23) | | | Registrert feil som: |
|--------------------------------|-----------|-------------------|-------|-------|-----------------|------|------|----------------|-------|------|----------------------|
| | | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | |
| Choose language for strictness | Stop | 6 | 3 | 1 | 9 | 1 | 0 | 9 | 1 | 0 | 1x? |
| | Finish | 6 | 4 | 0 | 10 | 0 | 0 | 7 | 3 | 0 | |
| | English | 0 | 5 | 5 | 5 | 2 | 3 | 10 | 0 | 0 | 8xFinish |
| | Norwegian | 9 | 0 | 1 | 9 | 1 | 0 | 9 | 1 | 0 | 1x? |
| Choose strictness level | Stop | 9 | 0 | 1 | 7 | 1 | 2 | 10 | 0 | 0 | 2xFive, 1xOne |
| | Finish | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | |
| | One | 9 | 0 | 1 | 9 | 0 | 1 | 10 | 0 | 0 | 2xFive |
| | Two | 9 | 1 | 0 | 10 | 0 | 0 | 7 | 2 | 1 | 1xFive |
| | Three | 6 | 0 | 4 | 10 | 0 | 0 | 9 | 0 | 1 | 4xFive, 1xFinish |
| | Four | 9 | 0 | 1 | 10 | 0 | 0 | 5 | 2 | 3 | 3xFive, 1xStop |
| Five | 9 | 0 | 1 | 8 | 2 | 0 | 8 | 2 | 0 | 1x? | |
| Snitt [%] | | 74,55 | 11,82 | 13,64 | 86,36 | 8,18 | 5,45 | 84,55 | 10,91 | 4,55 | |

Figur 6.4: Testresultater av modus for å justere talegjenkjenningssensitivitet.

| Tilstand | Kommando | Tester: Pasienten | | | Tester 1 (M 29) | | | Tester 3 (K23) | | | Registrert feil som: |
|----------------------------|----------|-------------------|------|------|-----------------|------|------|----------------|------|-------|------------------------|
| | | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | |
| Provide group/index number | Stop | 10 | 0 | 0 | 6 | 1 | 3 | 8 | 0 | 2 | 4xOne, 1xFive |
| | Finish | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Zero | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | One | 10 | 0 | 0 | 8 | 1 | 1 | 10 | 0 | 0 | 1xFive |
| | Two | 7 | 0 | 3 | 9 | 1 | 0 | 6 | 2 | 2 | 2xEight, 3x? |
| | Three | 9 | 0 | 1 | 9 | 1 | 0 | 10 | 0 | 0 | 1xEight |
| | Four | 10 | 0 | 0 | 9 | 0 | 1 | 4 | 4 | 2 | 1xFive, 1xOne, 1xEight |
| | Five | 8 | 2 | 0 | 9 | 0 | 1 | 10 | 0 | 0 | 1xOne |
| | Six | 10 | 0 | 0 | 8 | 1 | 1 | 7 | 2 | 1 | 1xOne, 1xEight |
| Seven | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | | |
| Eight | 9 | 1 | 0 | 8 | 1 | 1 | 9 | 1 | 0 | 1xSix | |
| Train chosen command? | Stop | 8 | 2 | 0 | 10 | 0 | 0 | 7 | 2 | 1 | 1x? |
| | Finish | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Yes | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | No | 8 | 2 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| Snitt [%] | | 92,00 | 5,33 | 2,67 | 90,00 | 4,67 | 5,33 | 86,00 | 8,67 | 5,33 | |

Figur 6.5: Testresultater av programmeringsmodus.

Testresultatene i fig 6.3, 6.4 og 6.5, viser at tallordene blir gjenkjent andre tall relativt ofte. Kommandoene fra disse modusene er alle SI-kommandoer. I figur 6.6 vises resultatene fra norsk SD-talegjenkjenning. Her vises tallordene å være et mindre problem. I et fremtidig revidert system bør det finnes et alternativ til tallene. En enkel løsning kunne vært å addere 10 til alle tallene, og dermed bruke “elleve” for “en” og “sytten” for syv. Dette ville fungere godt på norsk,

KAPITTEL 6. STEMMESTYRING I PRAKSIS

men på engelsk legges det bare på -teen fra thirteen og oppover, da vil problemet være det samme. Dette problemet er i all hovedsak på engelsk, det synes ikke å være nødvendig med en endring i norskmodus. En løsning for tallordene på engelsk, kan være å benytte “NATO phonetic alphabet” til å uttale tallene. I dette alfabetet er det laget uttalelser for tall fra 0-9 slik at de ikke skal kunne feiltolkes. Det kan imidlertid oppleves unaturlig og være mye å sette seg inn i for en bruker som ikke kjenner til dette fra før. Eksempel fra denne konvensjonen er One(wun), Four(fow-er).[30] Det må i tilfellet lages nye SI-kommandosett med teksten i parentesene.

| Tilstand | Kommando | Tester: Pasienten | | | Tester 1 (M 29) | | | Tester 2 (K 32) | | | Tester 3 (K 23) | | | Tester 4 (K 27) | | | Registrert feil som |
|--|-------------|-------------------|------|-----|-----------------|------|-----|-----------------|-----|-----|-----------------|-----|-----|-----------------|------|------|---------------------|
| | | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | |
| Angi ledd | Stopp | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Ferdig | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 8 | 2 | 0 | 10 | 0 | 0 | |
| | Albue | 8 | 2 | 0 | 7 | 3 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | |
| | Skulder | 9 | 1 | 0 | 7 | 3 | 0 | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | |
| | Hurtigmodus | 10 | 0 | 0 | 10 | 0 | 0 | 7 | 3 | 0 | 8 | 2 | 0 | 8 | 2 | 0 | |
| | Rask | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | ? | ? | ? | |
| | Sakte | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | ? | ? | ? | |
| Angi retning | Stopp | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 8 | 2 | 0 | 10 | 0 | 0 | |
| | Ferdig | 9 | 1 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Løft | 9 | 1 | 0 | 8 | 2 | 0 | 8 | 2 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Senk | 10 | 0 | 0 | 7 | 3 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| Angi sekunder | Stopp | 9 | 1 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 7 | 3 | 0 | |
| | Ferdig | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | En | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 8 | 2 | 0 | |
| | To | 10 | 0 | 0 | 8 | 2 | 0 | 7 | 3 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Tre | 8 | 2 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 0 | 1 | 1×En |
| | Fire | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Fem | 5 | 5 | 0 | 6 | 3 | 1 | 9 | 1 | 0 | 7 | 3 | 0 | 9 | 1 | 0 | 1×Ferdig |
| | Seks | 9 | 1 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| Syv | 8 | 2 | 0 | 9 | 1 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | | |
| Angi hurtigposisjon | Stopp | 5 | 5 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 7 | 3 | 0 | 10 | 0 | 0 | |
| | Ferdig | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Nøytral | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Bord | 9 | 1 | 0 | 10 | 0 | 0 | 8 | 2 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Ansikt | 5 | 5 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| Snitt [%] | | 88,8 | 11,2 | 0 | 88,8 | 10,8 | 0,4 | 90,4 | 9,6 | 0 | 91,6 | 8,4 | 0 | 92,61 | 6,96 | 0,43 | |
| Antall falske registreringer under testingen (FP1) | | ? | | | 0 | | | 2 | | | 1 | | | 3 | | | |

Figur 6.6: Testresultater av norskmodus som bruker SD-talegjenkjenning.

Analyse av testresultatene viser at nærmest alle kommandoer som er blir feilregistrert (FP2), er enstavelseskommandoer. Unntaket er at “English” forveksles med “Finish” i resultatene i figur 6.4. For å unngå dette er kommandoene “english” og “norwegian” byttet til “english language” og “norwegian language”. På samme måte som “english mode” ikke forveksles med “finish” løste dette problemet med denne forvekslingen. Fremfor å endre disse kommandoene, anbefales heller å justere sensitivitetsinstillingen for SI-kommandoer. Som vist i figur 6.7 minker prosentandelen for FP2 ved økning av sensitivitetsverdien.

Som det fremkommer av 6.2 presterer norskmodus noe bedre enn engelskmodus for sanne positive gjenkjennelser, men svært mye bedre enn engelskmodus for falskt positive gjenkjennelser. Dette gjelder for FP2, men også for FP1 som kan leses av nederst i testdiagrammene. Antallet falskt negative gjenkjennelser er imidlertid noe høyere for norskmodus. Det er imidlertid bedre at en kommando ikke blir gjenkjent enn at den blir gjenkjent feil. En til to av kommandoene for hver av testerne ble trent opp på ny på grunnlag av svært dårlig gjenkjennelse. Deretter ble denne kommandoen testet på ny. For de få kommandoene dette gjelder er resultatene etter en ny opptrening som står oppført i testresultatene. Dersom en kommando har lavere enn omlag 60% sann positiv gjenkjennelsesrate, vil det kunne hjelpe stort å trene den enkelte kommandoen på ny. Totalt ble det utført seks nye opptrening av de 4 testerne. Ved testing gjort av pasienten ble det ikke utført ny opptrening av enkelte kommandoer som presterte dårlig. For eksempel kan kommandoen “Stopp” i tilstanden “angi hurtigposisjon” kan antas å gi svake testresultater for pasienten på grunnlag av en svak trening. Dette kan antas da den samme kommandoen i de andre gruppene har en langt høyere gjenkjennelsesprosent.

6.1.1 Testing av engelskmodus med ulik sensitivitetsinnstilling

For å gi en pekepinn på virkningen av å justere talegjenkjenningssensitivitet, ble det utført tre tester av engelskmodus og starttilstandene av samme person. Standard verdi for sensitivitetsinnstillingen til SI-kommandoer er 3. Ved å øke verdien blir talegjenkjenningen strengere, det vil si at det kreves høyere grad av likhet mellom uttalt kommando og systemets mal av kommandoen for gjenkjennelse.

Først ble enheten testet med sensitivitetsinnstilling 5, så 4 og til slutt 3. Rekkefølgen ble valgt for å hindre at økt sensitivitetsinnstilling skulle få et falskt bedre resultat på grunnlag av øvelse. Det er naturlig at brukeren etterhvert lærer hvordan kommandoer burde uttales for at enheten skal gjenkjenne dem og tilpasser uttalen deretter. Testingen og registreringen av resultater blir også mer etter øvelse, dermed blir tidsperioden for å registrere falske aktiveringer redusert. Den mest effektive og dermed korteste økten var ved testing av strictness 3, allikevel var det en del falske aktiveringer. Resultatene viser at ved økt sensitivitetsinnstilling blir det flere falske aktiveringer, men færre falskt positive registreringer. Det vil si at enheten ikke like ofte gjenkjenner kommandoen som blir sagt, men den missforstår sjeldnere kommandoer. Dette er akkurat hva som er forventet ved å øke sensitivitetsverdien.

KAPITTEL 6. STEMMESTYRING I PRAKSIS

| Tilstand | Kommando | Tester: Anniken Olsen Strictness = 3 (default) | | | Tester: Anniken Olsen Strictness = 4 | | | Tester: Anniken Olsen Strictness = 5 | | | Registrert feil som |
|--|------------------|---|------|------|---|------|------|---|------|------|-----------------------|
| | | TP | FN | FP2 | TP | FN | FP2 | TP | FN | FP2 | |
| Resting state | Rise and Shine | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Stop | 9 | 1 | 0 | 9 | 1 | 0 | 6 | 4 | 0 | |
| Choose mode | Finish | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | English mode | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Norwegian mode | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Programming mode | 10 | 0 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Strictness | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| Awaiting joint | Stop | 10 | 0 | 0 | 9 | 1 | 0 | 9 | 1 | 0 | |
| | Finish | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Elbow | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Shoulder | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Rapid mode | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Quick speed | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Slow speed | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| Awaiting direction | Stop | 10 | 0 | 0 | 10 | 0 | 0 | 8 | 2 | 0 | |
| | Finish | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Lift | 8 | 2 | 0 | 9 | 0 | 1 | 9 | 1 | 0 | 1×Finish |
| | Lower | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| Awaiting seconds | Stop | 10 | 0 | 0 | 10 | 0 | 0 | 7 | 2 | 1 | 1×One |
| | Finish | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | One | 6 | 0 | 4 | 8 | 0 | 2 | 10 | 0 | 0 | 6×Five |
| | Two | 5 | 1 | 4 | 4 | 2 | 4 | 5 | 0 | 5 | 13×Seven |
| | Three | 9 | 1 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Four | 4 | 0 | 6 | 8 | 1 | 1 | 10 | 0 | 0 | 2×One, 3×Five, 2×Stop |
| | Five | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| | Six | 7 | 0 | 3 | 5 | 1 | 4 | 10 | 0 | 0 | 5×Seven, 2×Finish |
| Seven | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | | |
| Awaiting rapid position | Stop | 9 | 1 | 0 | 9 | 1 | 0 | 10 | 0 | 0 | |
| | Finish | 9 | 0 | 1 | 9 | 1 | 0 | 8 | 2 | 0 | |
| | Neutral | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Table | 10 | 0 | 0 | 10 | 0 | 0 | 9 | 1 | 0 | |
| | Face | 10 | 0 | 0 | 10 | 0 | 0 | 10 | 0 | 0 | |
| Snitt [%] | | 92,19 | 2,19 | 5,63 | 93,44 | 2,81 | 3,75 | 93,13 | 5,00 | 1,88 | |
| Antall falske registreringer under testingen (FP1) | | 10 | | | 11 | | | 4 | | | |

Figur 6.7: Testresultater av engelskmodus med samme testperson og ulike talegjenkjenningssensitivitet.

6.2 Tilbakemeldinger med utviklers kommentarer

Tilbakemeldinger mottatt under testing av enheten er samlet i dette delkapittelet. Alle testerne hadde norsk som morsmål og dermed er det naturlig å også ha en «norsk» aksent på engelskuttalen. SI-kommandoene er utviklet etter amerikansk uttale, at testerne har norsk som morsmål vil naturligvis spille inn på prestasjonen ved bruk av enhetens engelske moduser.

6.2.1 Tester 1

Tester 1 testet systemet i et stille rom. Han syntes programmet var lett å få oversikt over. Han likte godt indikasjonene fra enhetens ledlys. Han syntes det var oversiklig at lampen lyste når enheten lyttet etter en kommando og blinket når den lyttet ved opptrening av kommando. Tester 1 synes det krevde mer innsats for å få kommandoer gjenkjent i modusene med SI-kommandoer.

6.2.2 Tester 2

Tester 2 var en av de første til å teste systemet, hun syntes det tok litt tid å sette seg inn i systemet, men etter det var det lett forståelig. Underveis ble det pratet mye, ellers var det stille i rommet. Tester 2 ga tips og ideer til å gjøre diagrammene enda mer lettlest basert på tilbakemeldinger om systemets brukervennlighet. Tester 2 hadde problemer med kommandoene “one” og “two” og synes det var uforklarlig hvorfor det noen ganger ble registrert riktig og andre ganger ikke, da kommandoen ble forsøkt sagt på samme måte hver gang.

6.2.3 Tester 3

Testingen utført av tester 3 ble gjort med mye prat underveis og noe viftestøy i bakgrunnen. Tester 3 hadde problemer med enkelte kommandoer, som “two” og “four” og “seven”, dette kommer frem i alle SI-modusene. Testeren forsøkte å si kommandoene på mange ulike måter, men kommandoen ble stadig gjenkjent som noe annet enn hva som ble sagt.

6.2.4 Tester 4

Test av norskmodus ble gjort i et stille rom uten mye prat. Test av engelskmodus ble gjort i en folksom restaurant med musikk. Musikken og folkestøyen ble bakgrunnstøy og tester 4 sine testresultatene av engelskmodus er faktisk bedre enn snittresultatene for TP, FN og FP2. Testeren har bodd flere år i engelsktalende land og har svært god engelskuttale. Sensitivitetsinstillingen for SI-kommandoer ble justert opp til 4 ved denne testen.

Omgivelsen enheten har prestert dårligst i er ved høyfrekvent viftestøy fra klimaanlegget i et kontorbygg. Enheten kan synes sensitiv for noen typer støy, mens når den ble testet på den folksomme restauranten var det ingen tilsynelatende hindring.

Tester 4 meddelte under testingen at hun mente hun kunne fått bedre testresultater om hun hadde konsentrert seg mer. Flere ganger påpekte hun at kommandoene hun sa ble sagt for utydelig eller for raskt til å kunne forvente at talegjenkjenningen skulle registrere kommandoen.

Det stemmer at jevnt volum og tempo samt god artikulasjon av kommandoene gir vesentlig økt sjansje for at kommandoen skal bli registrert. Det er også øke suksessraten å si kommandoer rettet mot enhetensmikrofon. Pasienten var den eneste av testerne som hadde prøvd enheten før testingen. At en tester ikke kjenner til systemet fra tidligere kan gi et svakere testresultat.

Etterhvert vil en bruker lære hvilket tempo, volum og artikulasjon kommandoene bør ha for at enheten skal gjenkjenne kommandoene. Allerede i løpet av testingen la jeg merke til at testerne begynte å eksperimentere med ulike uttaler for bedre gjenkjenning. Flere av testerne påpekte selv at hvordan de sa ordene hadde betydning for gjenkjennelsen.

Tester 4 viste raskt forståelse for systemet og jobbet aktivt for å si kommandoer med tydelig artikulasjon. Det kreves mer testing for å avgjøre om tester 4 sine gode testresultater kommer av forståelsen av systemet, god artikulasjon, spesielt god engelskuttale eller om stillhet eller jevn bakgrunnsstøy er miljøer talegjenkjenningen presterer godt i.

6.2.5 Tester: Pasienten

Pasienten testet stemmestyringen med gode resultater og syntes den fungerte godt på samtlige kommandoer. En utfordring pasienten hadde møtt var at gjenkjennelse av tidligere trente kommandoer var veldig dårlig i tilfeller hvor kommandoene ble trent med en annen plassering og avstand enn det de senere ble brukt. Dette ble bemerket etter stemmestyringen var trent opp og benyttet mens den var et stykke unna og når den senere ble plassert på brystet var det ikke mulig å navigere ut av norskmodus, da enheten ikke lystret “ferdig”. Etter at enheten ble festet med borrelås-tape på samme område hver gang, som på bilde 6.1, har pasienten ikke møtt på dette problemet igjen.

For å unngå at enheten låser seg i norskmodus, kan kritiske kommandoer som “ferdig” og “stopp” trenes ved start når enheten brukes i nytt miljø eller oppsett. Dermed kan det navigeres til programmeringsmodus og andre kommandoer kan også trenes på ny dersom det viser seg nødvendig.

Pasienten har meddelt at brukeropplevelsen med stemmestyringssystemet har vært enklere enn talegjenkjenningssystemer pasienten har til andre foremål fra hjelpemiddelsentralen. Pasienten

opplever bruksanvisningen og figurene som intuitive og godt tilpasset sine behov.

Det ble brukt mye tid på å lage diagrammer som viser all informasjon en bruker trenger til å navigere en av stemmestyringens moduser på én side. For en pasient med ALS er det langt fra enkelt å bla i en bruksanvisning. Derfor er det i bruksanvisningen for hver modus laget et tilstandsdiagram som viser starttilstandene for å navigere til modusen og alle gyldige kommandoer for tilstandene i modusen.

En av tilbakemeldingene fra pasienten ved test av en tidligere versjon, var at dette ikke var tydelig nok hvilken tilstand enheten var i. For å gjøre enheten raskere og behagelig å bruke er tilbakemeldingene fra enheten begrenset til et minimum. Det er allikevel svært viktig at brukeren forstår hvor i systemet den befinner seg. Derfor ble det lagt til flere tilbakemeldinger, det tilbakemelding for hver kommando en tilbakemelding og enhetens tilstand angis der det er naturlig.

6.3 Stemmestyring i forhold til tidligere styringssystemer

De tidligere utviklede styringssystemene til den motoriserte ortosen er beskrevet og vurdert i delkapittel 3.3. Joystick og fotspaker er to styringssystemer som har fungert godt. Begge deler er godt egnet til ulike sykdomsstadier. Begrensningene disse styringssystemene har er at joystick krever håndfunksjon og vil okkuperer en hånd for å kunne løfte den andre. Fotbryter krever førlighet i føtter og krever at man sitter eller står i ro da enheten er utfordrende å flytte rundt for en bruker med svekket armfunksjon.

Stemmestyring kan sitte på skulderen, uten å være i veien eller være en motorisk distraksjon. Stemmestyring har mulighet for utvidelse med nye moduser til å kunne styre andre enheter uten å innvirke med modusen for bevegelse.

Ved finjustering av bevegelse presterer joystick og fotspaker langt bedre enn stemmestyring. Bevegelse med spaker gir umiddelbar respons og hastighet kan justeres trinnløst. Joystick og fotspaker har også fordelen av at det er pålitelig og vil fungere på tilsvarende måte i alle omgivelser, det samme kan ikke sies for stemmestyring.

Trykksensorstyring hadde i teorien hatt potensiale til å utkonkurrere alle de nevnte styringssystemene, da denne styringsmetoden ville vært portabel og gitt umiddelbart og trinnløst pådrag, i tillegg ville det kunne vært intuitivt å bruke.

7 | Diskusjon og konklusjon

Dette har vært en omfattende oppgave med mange spennende aspekter. Fra planlegging av systemet og kontakt med pasienten, til å få bygge et stemmestyringsystem som vil være viktig i pasientens hverdag. Stemmestyringssystemet er tilpasset pasienten som prototypen er utviklet til, men ikke på bekostning av mulighet for at andre skal kunne bruke systemet. Det er blitt prioritert å bruke mye tid på å utvikle et system som vil kunne være fungerende for pasienten over tid. Stemmestyringsenheten ble gitt til pasienten to måneder før avslutning av masteren, dermed var det mulig å lage en programoppdatering med små endringer etter pasientens erfaringer.

Valget av implementeringsteknologi synes å være fornuftig da komponentene var innenfor budsjettet og det var mulig å utvikle et godt produkt innen tidsrammen. Å lage et system med foroverkobling synes også være en god avgjørelse da utviklingsarbeidet viste seg å være svært tidkrevende. Når systemet var mulig å realisere med foroverkobling, var det ikke nødvendig å gjøre systemet mer komplisert enn det trengte å være.

Bruk av V-modellen for å organisere systemutviklingen gjorde fremgangen oversiktlig. De ønskede produktspesifikasjonene til det ferdige systemet ble stadig endret og utvidet basert på ideer som kom underveis, og utfordringer og muligheter som viste seg under testing og utvikling av systemet. Så stegene i V-modellen ble benyttet som inspirasjon og til ettertanke, ikke som faste krav til systemutviklingens progresjon.

Under presentasjonen av testresultatene kunne det vært nyttig med en ekstra kolonne i testresultatene som viste snittet av TP, FN og TP2 for hver av kommandoene. Dette ville ikke gjort en så stor forskjell med kun 5 testere, men det ville vært spesielt nyttig ved en større testgruppe. Dette ville gjøre det raskt å identifisere hvilke kommandoer som presterte dårlig.

7.1 Vurdering av egnethet

Stemmestyring av et motorisk hjelpemiddel byr på en rekke utfordringer, men det har også mange fordeler og muligheter. Stemmestyring er sensitivt for støy og er avhengig av sikkerhetsmekanismer, som grensebrytere, da det ikke er 100% påtellig. Stemmestyring er allikevel et godt alternativ for en kompakt og portabel styringsmekanisme.

Prestasjonen til stemmestyringsenheten under testingen anser jeg som tilstrekkelig god. En styringsmekanisme som pasienten kan bruke, og som fungerer korrekt ved 9 av 10 tilfeller ansees som tilfredstillende i dette tilfellet. Ortosen har sikkerhetsmekanismer i form av grensebrytere, som begrenser skaden en feil registrering eller en falskt negativ registrering kan gi. I et slikt utviklingsarbeid vil gode testresultater være vesentlig, men enda viktigere er det at brukeren har gitt gode tilbakemeldinger og kan bruke styringssystemet.

Pasienten bruker i dag hovedsakelig norskmodus. Pasienten har endret kommandoen “løft” til “heve” da han opplevde problemer med denne kommandoen. Tilbakemeldingen over høyttaleren vil fortsatt være “løft” frem til nye lydfiler lastes opp, men det utgjør ikke et funksjonelt problem. Etter pasienten trente opp kommandoen til “heve” har det ikke vært registrert noen problemer med denne kommandoen. Pasienten fortalte at stemmestyringsenheten er benyttet til å styre ortosen for å børste tennene. Det ble gjort ved å bevege ortosen til passende posisjon og dermed beveget hodet frem og tilbake. Dette er ikke en ideell løsning, men det fungerer.

Pasientens tilbakemeldinger tyder på at målet på å lage et funksjonelt, intuitivt og enkelt system har lyktes. Det ble underveis forsøkt å ha tre bevegelsesmoduser, i stedet for at dette ga bedre funksjonalitet ved flere muligheter, syntes det for pasienten mer vanskelig å få oversikt over systemet og det ble mer krevende å navigere. Færre moduser gir også færre kommandoer å forholde seg til og som må trenes opp.

Sammen med en metode for finjustering, anses stemmestyring for å være et godt egnet styringssystem for den motoriserte ortosen. Stemmestyring er vist seg mest passende ved store bevegelser, så uten en mulighet for finjustering blir styringen mer tungvindt, men fortsatt funksjonelt.

7.2 Fremtidig arbeid

I utviklingsarbeidet og etter behov fremmet av pasienten har det dukket det opp mange ideer for videreutvikling av stemmesstyringssystemet og nye bruksområder til styringssystemet. Stemmestyringen kan gjøres mer effektiv ved et display som viser enhetens tilstand, dette gjør at det ikke brukes tid på å spille av lydklipp.

Hurtigmodus kan forbedres med mulighet for å lagre nye hurtigposisjoner. Ved at pasienten kan kjøre armen til en posisjon som ofte brukes og deretter lagre denne posisjonen til en kommando. Dette vil kunne gjøre hurtigmodus bli mye mer anvendelig.

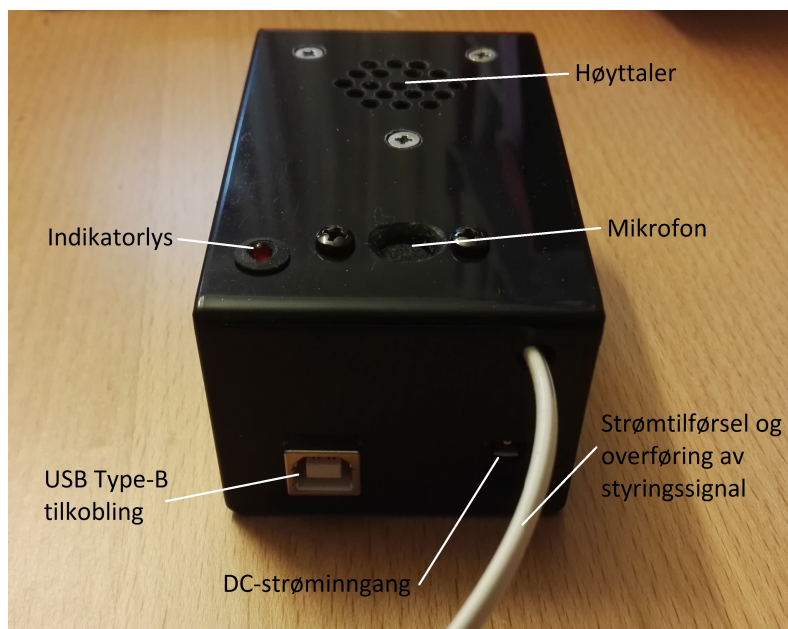
Stemmestyringsenheten kan utvides med modus hvor pasienten med kommandoer kan justere stillingen på pasientens elektriske stol. Her vil stemmestyring være godt egnet da brukeren kan forhåndsinstille noen favorittstillinger og skifte mellom dem med kommandoer. Utover dette kan modusen benyttes for å sende signal om å tenne lamper, skru på TV og åpne dører

Det er interessant å utføre videre testing av enheten for å se om den kan gjenkjenne lyder, dette vil gjøre at stemmestyringsenheten kan brukes over flere sykdomsstadier. Da at pasienten ikke behøver mulighet for artikulasjon for å lage lyder. For å kunne realisere dette må enheten kunne skille mellom lyder.

A Bruksanvisning for stemmestyringsenhet til armortose

A.1 Om stemmestyringsenheten

Stemmestyringsenheten består av en Arduino Uno med ATmega168 mikrokontroller, sammenkoblet med en EasyVR modul med mikrofon og høyttaler. Markert på figur A.1 er alle innganger og utganger til stemmestyringsenheten.



Figur A.1: Stemmestyringsenheten.

A.2 Stemmestyringsenhetens innganger og utganger

Indikatorlys: Lyser rødt når mikrofonen lytter etter kommando.

Hvit ledning: Overfører styringssignal til ortosen og gir strømforsyning til stemmestyringsenheten. Enheten trekker omtrent 30 mA i standby og 3 mA når den er aktiv.

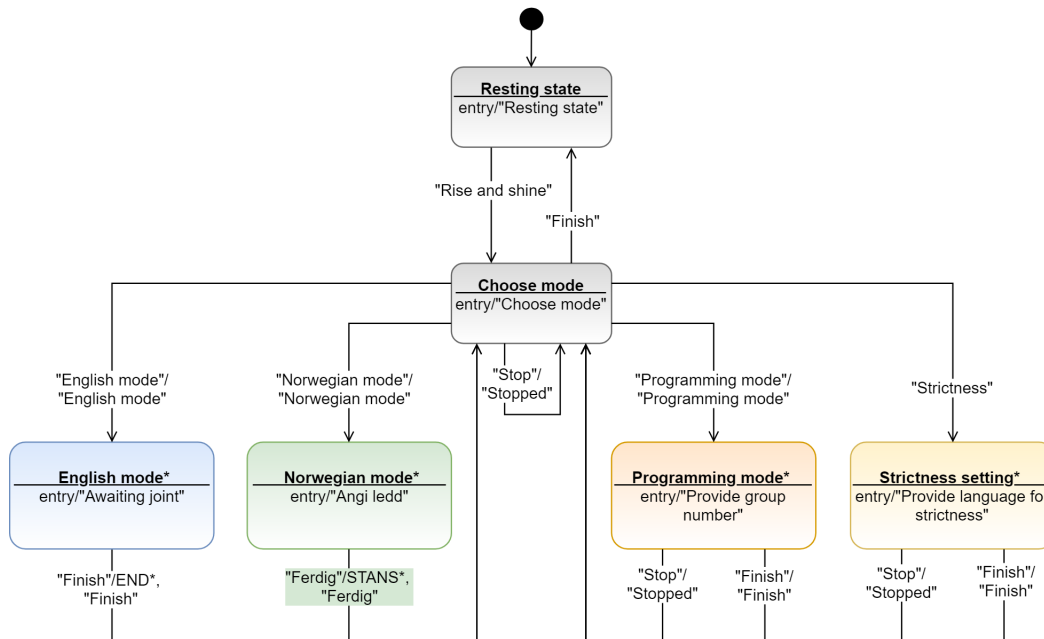
USB Type-B tilkobling: Kan benyttes for å koble stemmestyringen til PC eller forsyne boksen med strøm. Stemmestyringen har en programmeringsmodus hvor de norske kommandoene kan trenes uten å koble til datamaskin, men kommandoer kan også trenes ved å koble til datamaskin med USB, se underkapittel A.8. For å skifte ut lydfilene som utgjør tilbakemeldingene som spilles over høyttaleren, kreves det datatilkobling.

DC-strøminngang: Boksen blir forsynt med strøm gjennom den hvite ledningen, men kan også forsynes med strøm-plug for Arduino UNO-kort, med spenningstilførsel fra 7 - 12 V.

A.3 Stemmestyringsenheten i bruk

Når enheten kobles til strøm starter den i “Restingstate”, som er markert ved den sorte prikken i figur A.2. Stemmekommandoene som må angis for å gå fra en modus til en annen er angitt i hermetegn på pilene mellom modusene. Kommandoene med grønn bakgrunn er norske, alle andre kommandoer er engelske.

Den røde lampen lyser når enheten lytter etter en kommando. Når en kommando gjenkjennes, vil en passende tilbakemelding spilles over høyttaleren. Dette vil for eksempel være en gjentakelse av den registrerte kommandoen eller hendelsen kommandoen fører til. Dersom en kommando ikke blir gjenkjent, altså det ikke gis en tilbakemelding etter kommandoen er gitt, kan brukeren prøve å si kommandoen igjen etter ett sekund. Kommandoene “Stopp” på norsk eller “Stop” på engelsk vil avslutte motorisk aktivitet og enhetens tilstand vil bli annonsert over høyttaleren. For å velge en annen modus kan en når som helst si “Ferdig” eller “Finish”, da vil “Choose mode” spilles over høyttaleren og en kan velge modus. Sies istedet “Finish” i denne tilstanden, vil enheten gå i hvilemodus hvorpå kun triggerordet “Rise and shine” vil vekke enheten. “Ferdig” eller “Finish” vil også avslutte eventuell motorisk aktivitet. Dersom det er usikkert hvilken tilstand enheten er i, vil enten alltid lystre på enten “Rise and shine”, “Stopp” og “Ferdig” eller “Stop” og “Finish”.



Figur A.2: Oversikt over modusene i stemmestyringsprogrammet og deres tilknytning. Den indre strukturen til boksene med stjerne vises i figurer med samsvarende farge i figur A.3 - A.7.

I tilstanden “Choose mode” kan en velge modus ved kommandoene “Norwegian mode”, “English mode”, “Programming mode” eller “Strictness setting”. Norskmodus og engelskmodus har helt lik struktur, med kommandoene oversatt. Programmeringsmodusen benyttes til å trene norskmodusen. Den engelske modusen trenger ikke, og kan ikke, trenes. Norskmodus derimot fungerer ikke med en ny bruker før den er trent opp. Etter den er trent, er meningen at den kun vil lystre på brukeren som har trent den. Hver av de fire modusene vil forklares i de kommende delkapitlene.

I figur A.2-A.7 vises skjematisk oversikt over stemmestyringen i UML-diagrammer. På transisjonspilene mellom tilstandene står kommandoer som fører til ny tilstand. I tilstandsboksene står tilstandens indre aktiviteter. Teksten i diagrammet har formatet **hendelse[transisjonsbetingelse]/aksjon**. En aksjon kan være å starte motorene eller å gi en tilbakemelding over høyttaleren. Ofte vil en av de tre, hendelse[transisjonsbetingelse]/aksjon, være oppført alene. Slik som [timeout] eller “Albue”, men en aksjon vil aldri inntreffe uten en hendelse. En transisjonsbetingelse på en transisjonslinje vil forhindre at transisjonen kan gjøres før betingelsen er sann. Når aksjonen er en auditiv tilbakemelding og når hendelsen er en kommando som skal sies, vil det stå i anførselstegn. Eksempelvis “Stop”/“Stopped”, hvor “Stop” er hendelsen og “Stopped” er aksjonen.

Hendelsen **entry**/ opptrer når enheten går inn i en ny tilstand, denne hendelsen fører ofte til en tilbakemelding over høyttaleren, slik som **entry**/“**Angi ledd**”.

A.4 Engelskmodus

Den engelske modusen er ikke personavhengig, altså hvem som helst bruke denne modusen uten opptrening av kommandoer på forhånd. Kommandoene genereres fra skrift ved at et dataprogram lager et forventet lydbilde av den skriftlige kommandoen. Sammenlignet med norskmodus vil kommandoer gjenkjennes oftere selv om de ikke sies veldig presist, men risikoen for at enheten reagerer på generell tale eller lyder og tolker det som en kommando er også høyere enn for norskmodus. Dersom enheten gir tilbakemelding uten at en kommando er gitt betyr det at den har tolket lyd som en kommando, da vil det være hensiktsmessig å si “Stop” eller “Finish” for å unngå uønsket bevegelse. Oversikt over kommandoene for å navigere de ulike tilstandene i engelskmodus er gitt i figur A.3. Styringen er bygget opp med to forskjellige greiner. I hurtigmodus må ortosen alltid kjøres enten til “Neutral”- eller “Face”-posisjon for å kalibreres til en endeposisjon, før den kan kjøres til “Table”. Dette er nødvendig siden det foreløpig ikke er noen tilbakekobling i styringssystemet. Dersom det blir sagt “Stop” må den igjen kjøres til en endeposisjon før “Table”.

A.5 Norskmodus

Norskmodus har personavhengige stemmekommandoer, de vil bare kunne brukes av brukeren som har trent kommandoene eller andre med svært lik stemmekarakteristikk. Kommandoene trenes ved at brukeren gjentar kommandoordet noen ganger og ut fra dette dannes et forventet lydbilde av den angitte kommandoen. Den norske modusen kommer utrent, og denne modusen vil ikke være mulig å bruke uten å trene kommandoer i programmeringsmodus først. Se figur A.5 for programmeringsmodusen og figur A.6 for oversikt over kommandoenes gruppe og indeksnummer. Oversikt over norskmodusen er i figur A.4. Kommandoene for norsk modus kan trenes opp til andre ord enn de som står i diagrammet dersom det er ønskelig. Talegjenkjenningen i norskmodus presterer best når kommandoene sies i lignende toneleie, tempo og lignende avstand til mikrofonen som da kommandoen ble trent.

A.6 Programmeringsmodus

Programmeringsmodusen er utviklet for at brukeren aldri skal behøve å koble stemmestyringen til en datamaskin. Siden enheten foreløpig mangler støtte for norske kommandoer som ikke er personavhengige er programmeringsmodusen på engelsk. De engelske kommandoene i stemmestyringen er ikke personavhengige og trenger ikke å trenes. Det betyr at enhetens norske modus kan trenes opp av hvem som helst og deretter er meningen at den norske modusen kun skal lystre den stemmen den er trent med. Dette er årsaken til at de norske kommandoene trenes i en modus hvor navigeringskommandoene er på engelsk.

I programmeringsmodusen trenes en kommando med 3-6 opptaksforsøk. Først sies gruppenummer, deretter angir brukeren indeksnummer og kommandoen tilknyttet det angitte gruppe og indeksnummeret sies over høyttaleren. Brukeren får spørsmålet om denne kommandoen skal trenes og svarer “Yes” eller “No”. Etter treningen av kommandoen er ferdig spør enheten om nytt indeksnummer i samme gruppe. Når brukeren er ferdig å trene i den angitte gruppen vil kommandoen “Stop” føre til at brukeren kan velge ny gruppe. De norske kommandoenes gruppe og indeksnummer står oppført i tabellen i figur A.5.

Treningen foregår ved at brukeren blir bedt om å si den ønskede kommandoen før hvert opptak startes. Hver trening består av maksimalt seks opptak. Indikatorlyset vil blinke når kommandoen skal sies. Det vil ikke være mulig å endre tilstand før treningen er fullført og lampen tennes igjen, da enheten ikke lytter etter kommandoer under opptakene.

Når treningen av en kommando er godkjent blir brukeren bedt om å si kommandoen igjen. Treningen fortsetter frem til det er oppnådd tre godkjente treninger eller til det er gjort seks treningsforsøk. Dersom treningen ikke er tilstrekkelig lik de foregående treningene, vil den ikke bli godkjent. En kommando som er for lang eller en kommando lest inn med for mye prat eller bråk i bakgrunnen, vil heller ikke godkjennes. Når en trening ikke godkjennes får brukeren tilbakemeldingen “Failed”. Selv om en trening feiler, vil brukeren fortsatt få muligheten til å oppnå tre godkjente treninger med opp til seks forsøk totalt. Når det oppstår en feil eller en konflikt i treningen, kan det forekomme at enheten sletter tidligere treninger av kommandoordet som trenes. I slike tilfeller kan treningen avslutte før seks treningsforsøk er gjennomført og før tre godkjente treninger er oppnådd. Det bør være minimum to godkjente treninger av hver kommando for å få presis nok gjenkjennelse av kommandoen. Når 1-3 treninger blir godkjent gis tilbakemeldingen “one/two/three sucesfull trainings”. I tilfeller der ingen av de tre opptakene

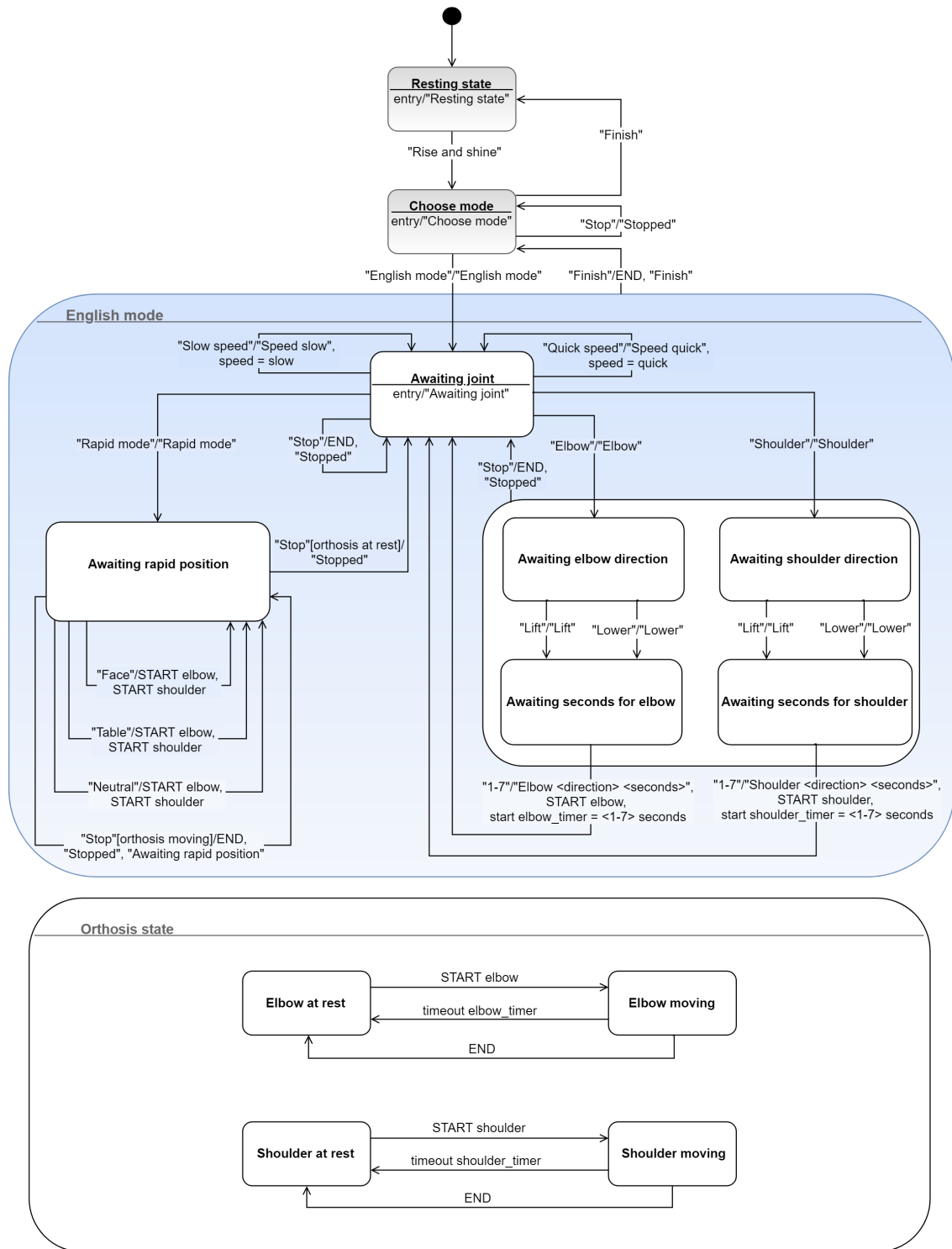
blir godkjent, får brukeren tilbakemeldingen “Training failed”. Kommandoene som får 0 eller 1 godkjente treninger bør trenes på ny. For å få en oversikt over antallet godkjente treninger alle de ulike kommandoene har, må stemmestyringsenheten kobles til dataprogrammet “EasyVR commander”, se figur A.9). Etter alle kommandoene er trent opp ved første gangs bruk, kan brukeren i senere tid gå tilbake og trene enkelte kommandoer på nytt. Dersom en kommando ofte lyster dårlig eller tolkes feil, kan det hjelpe å trene kommandoen på ny. Hvis noen ord er vanskelige å uttale eller vanskelig å få registrert, kan kommandoer trenes til et annet ord. Tilbakemeldingslydfilene vil ikke endre seg, så det kan bli forvirrende å endre kommandoer til noe som ikke passer med tilbakemeldingene som gis. Der hvor samme ord er i flere grupper i figur A.6 må ordet trenes opp for hver gruppe. Som i de andre engelske modusene vil kommandoen “Finish” returnere enheten til hvilemodus.

A.7 Talegjenkjenningssensitivitet

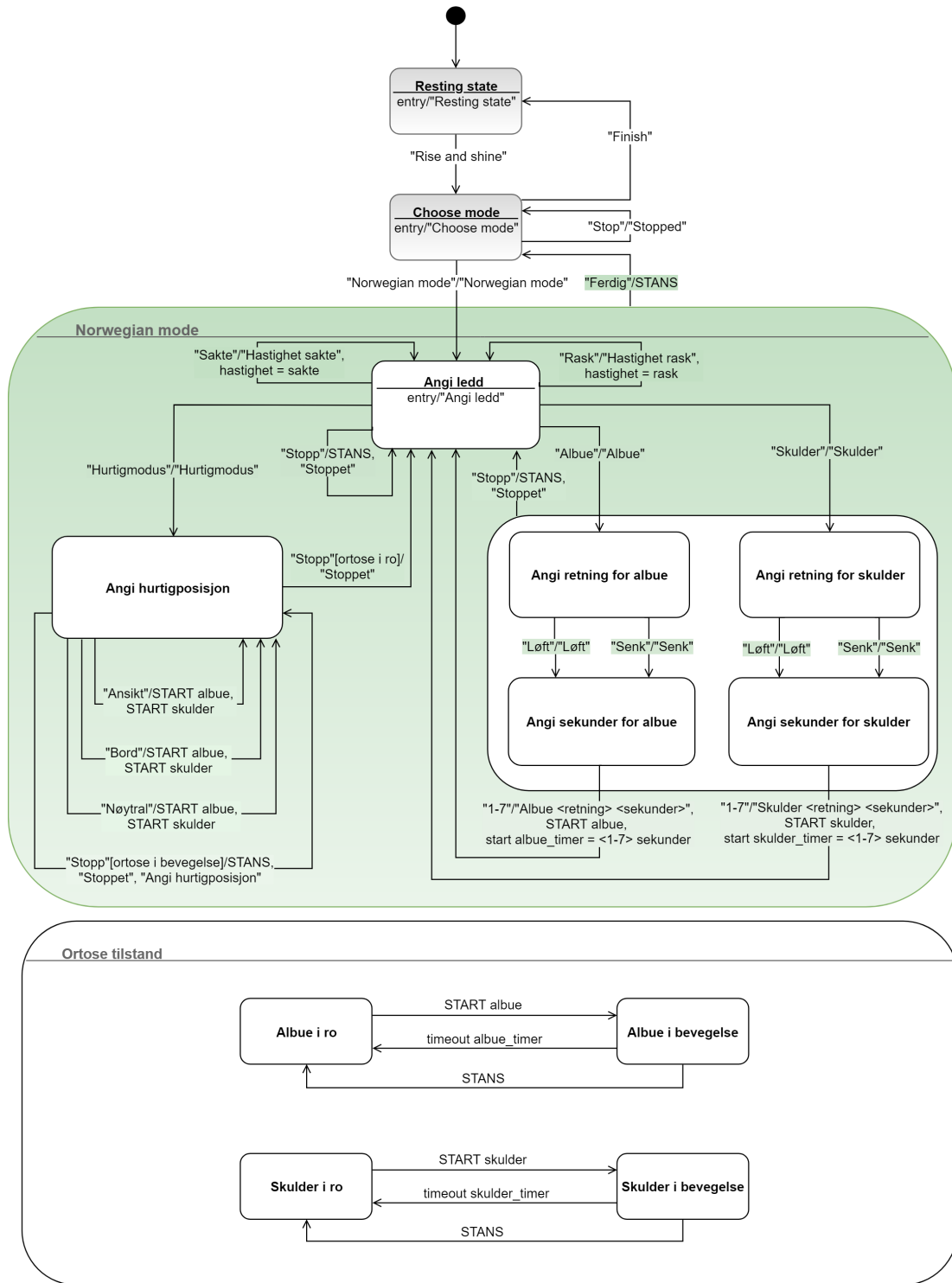
I modusen “strictness setting” kan brukeren, med et tall fra 1-5, angi hvor nøyaktig en kommando må være for at den skal registreres av enheten. Sensitiviteten stilles uavhengig for norskmodus og engelskmodus, hvor nivå 5 er mest sensitiv og nivå 1 minst sensitiv. Dersom brukeren opplever at en kommando må sies mange ganger før den registreres, kan det hjelpe å stille sensitivitetsnivået lavere. På den andre siden, om kommandoer ofte blir forvekslet med andre kommandoer eller enheten falskt aktiverer når ingen kommando ble gitt, kan det hjelpe å sette sensitivitetsnivået høyere. Sensitivitetsinstillingen lagres på enheten og vil forbli den samme frem til den blir endret, dette gjelder selv om enheten kobles fra ortosen og fra strøm.

A.8 Datatilkobling til EasyVR commander

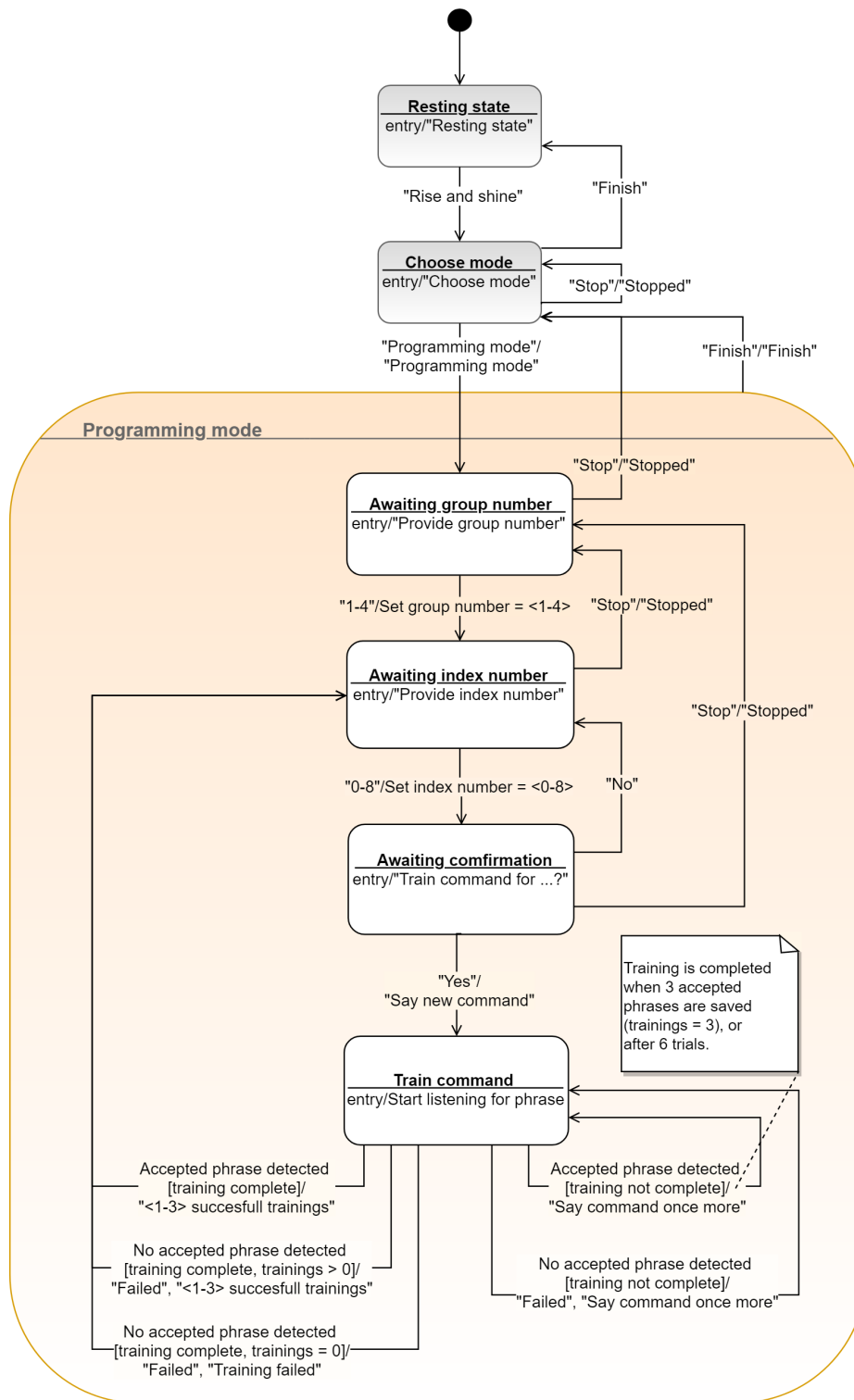
Når enheten er koblet til datamaskin, kan brukeren trene norske kommandoer og endre lydfilene på enheten med programmet “EasyVR commander”. Nye tilbakemeldings-lydfiler må først komprimeres til riktig format ved bruk av programmet “QuickSynthesis” , deretter kan de lastes opp til enheten med EasyVR commander", se figur A.10 og A.11. Dataprogrammene kan lastes ned fra [http : //www.veear.eu/downloads/](http://www.veear.eu/downloads/). Når nye lydklipp skal lastes inn må stemmestyringsenheten åpnes forsiktig og “mode jumper” settes i “UP”, se figur A.8. Når opplastingen er fullført settes “mode jumper” tilbake til “SW” før boksen lukkes varsomt igjen.



Figur A.3: Oversikt over kommandoene mellom engelskmodusens ulike tilstander.



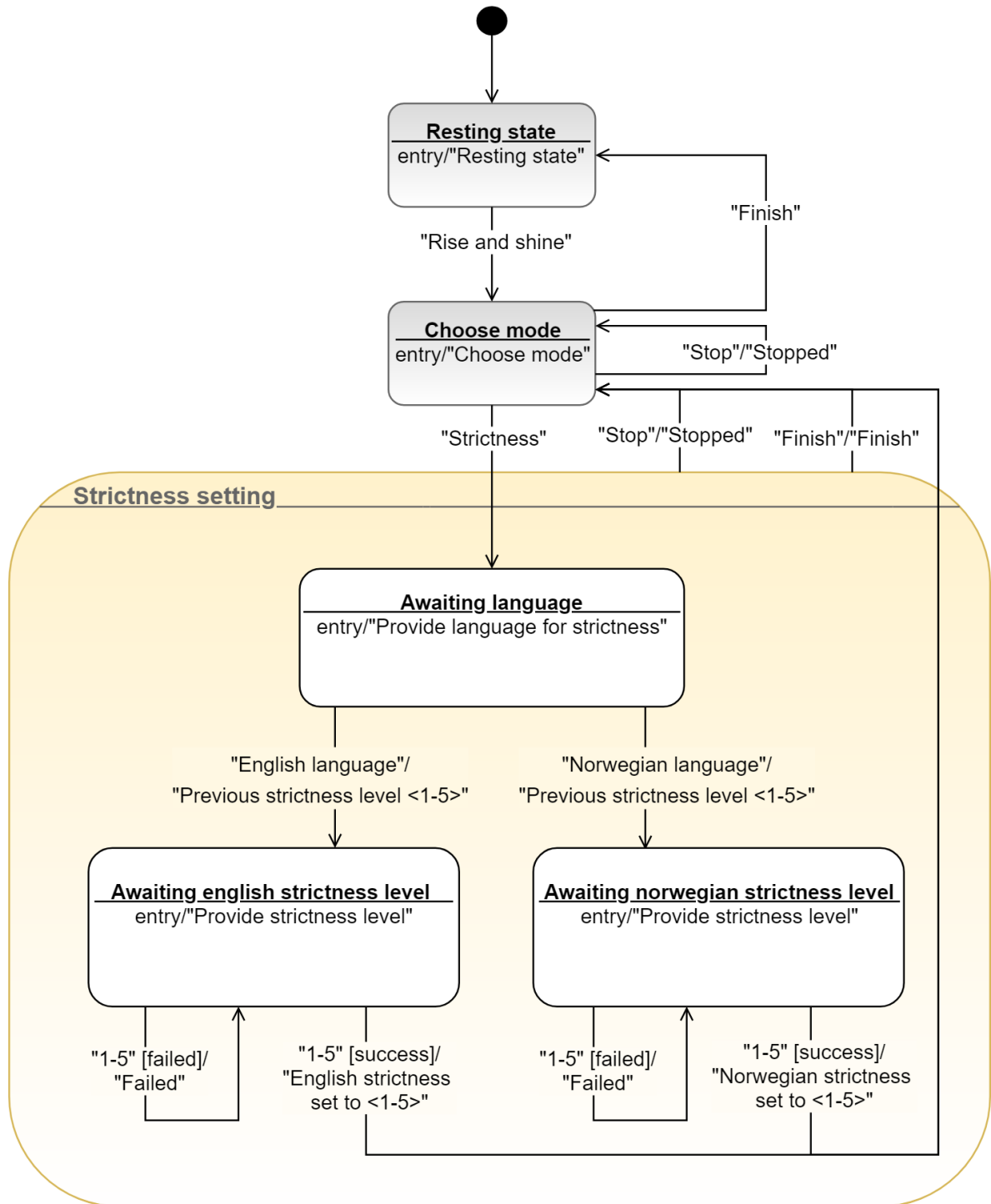
Figur A.4: Oversikt over kommandoene mellom norskmodusens ulike tilstander.



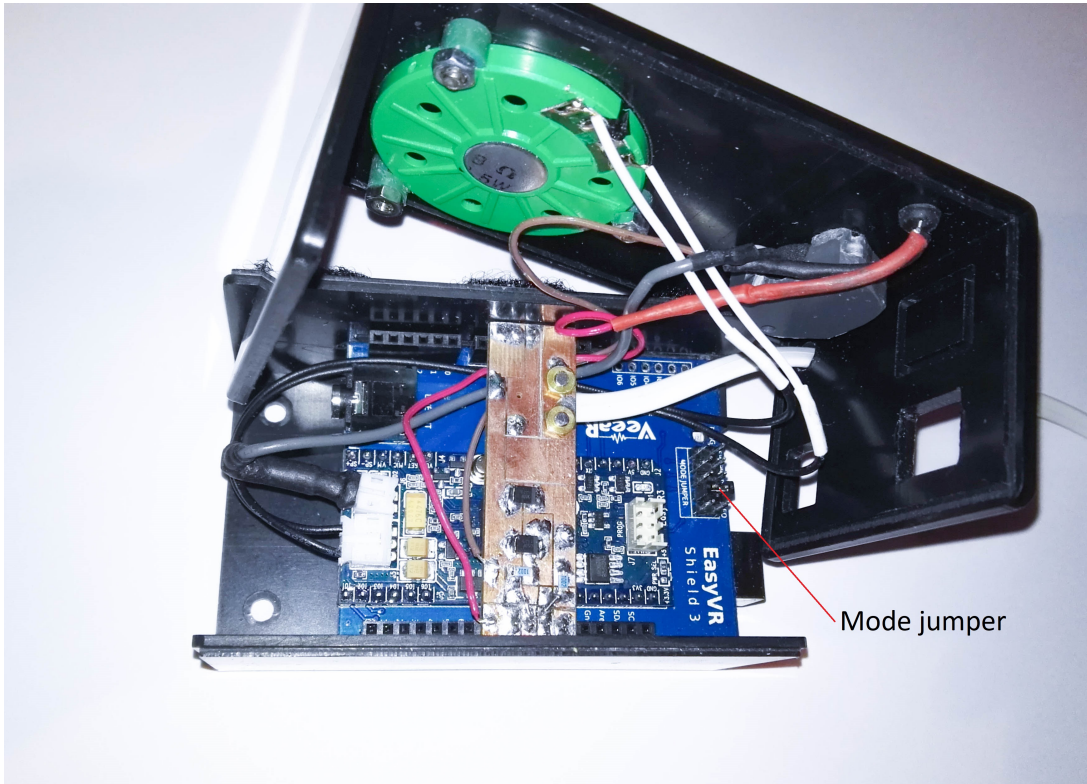
Figur A.5: Fremgangen for trening av norske kommandoer i programmeringsmodusen.

| Gruppe | Indeks | Kommando |
|--------|--------|-------------|
| 1 | 0 | Stopp |
| | 1 | Ferdig |
| | 2 | Albue |
| | 3 | Skulder |
| | 4 | Hurtigmodus |
| | 5 | Rask |
| | 6 | Sakte |
| 2 | 0 | Stopp |
| | 1 | Ferdig |
| | 2 | Løft |
| | 3 | Senk |
| 3 | 0 | Stopp |
| | 1 | Ferdig |
| | 2 | En |
| | 3 | To |
| | 4 | Tre |
| | 5 | Fire |
| | 6 | Fem |
| | 7 | Seks |
| | 8 | Syv |
| 4 | 0 | Stopp |
| | 1 | Ferdig |
| | 2 | Nøytral |
| | 3 | Bord |
| | 4 | Ansikt |

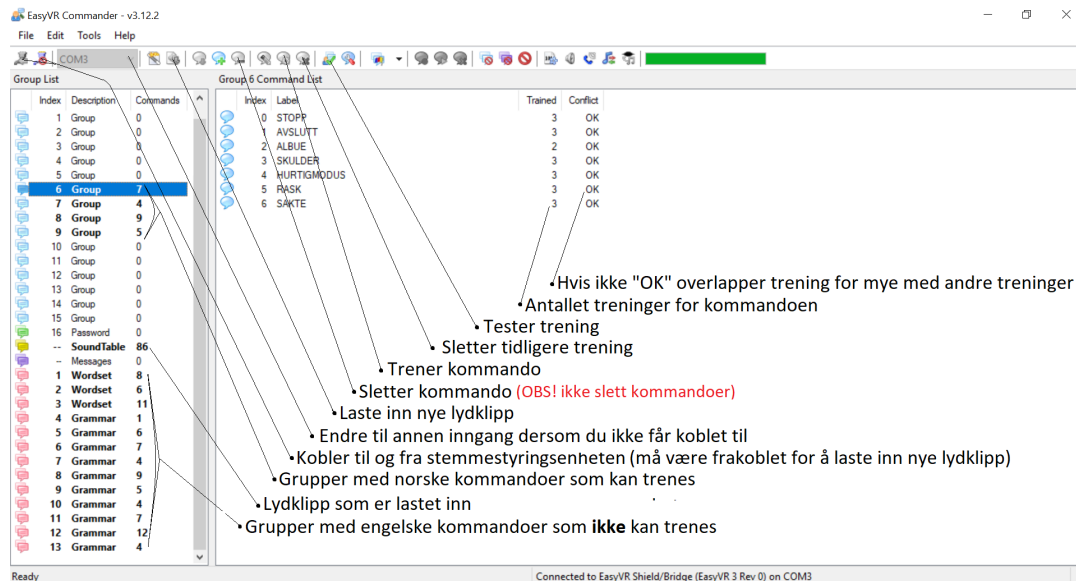
Figur A.6: Alle de norske kommandoene med tilhørende gruppe- og indeksnummer.



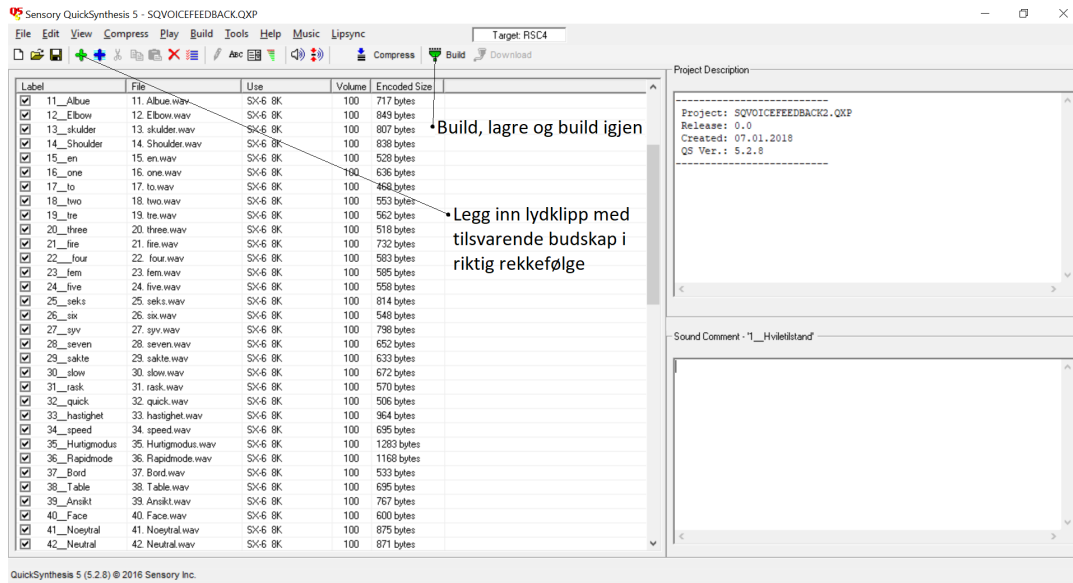
Figur A.7: Oversikt over modusen hvor talejerkjennings-sensitiviteten stilles.



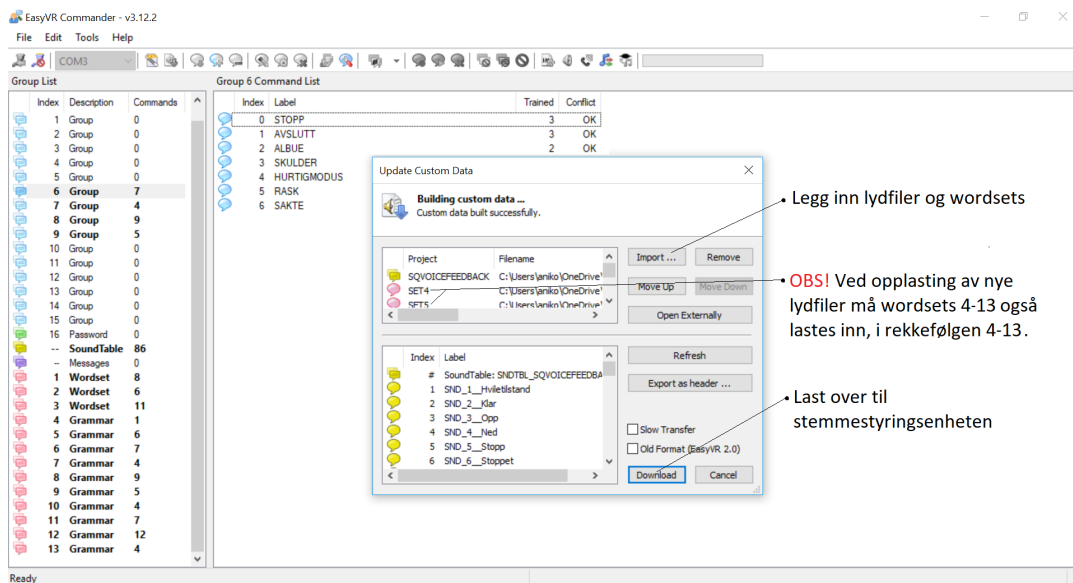
Figur A.8: Stemmestyrensenheten åpnet, "mode jumper" skifter mellom modus for stemmesty- ring (SW) og opplasting av lydfiler med EasyVR commander (UP).



Figur A.9: Skjerm bilde fra dataprogrammet EasyVR commander, programmet benyttes ved datatilkobling av stemmestyrensenheten.



Figur A.10: Skjerm bilde fra dataprogrammet QuickSynthesis, som benyttes til å komprimere og samle lydfile. Lydfilene kan deretter lastes over til stemmestyringsenheten via EasyVR commander og benyttes som tilbakemeldinger over høyttaleren.



Figur A.11: Skjerm bilde fra dataprogrammet EasyVR commander. Innlasting av lydfile og engelske ordsett.

Bibliografi

- [1] Store Medisinske Leksikon, www.sml.snl.no (ALS, Ortose).
- [2] R. A.R.C. Gopura, D. S.V. Bandara, Kazuo Kiguchi, and G. K.I. Mann. Developments in hardware systems of active upper-limb exoskeleton robots: A review. *Robotics and Autonomous Systems*, 75:203–220, 2016.
- [3] Panagiotis Polygerinos, Zheng Wang, Kevin C. Galloway, Robert J. Wood, and Conor J. Walsh. Soft robotic glove for combined assistance and at-home rehabilitation. *Robotics and Autonomous Systems*, 73:135–143, 2015.
- [4] Max Manus, samtale 15.03.18.
- [5] Abilia, samtale 11.05.18.
- [6] Anniken Olsen. Styringsprinsipper for ortose til ALS-pasient. 2017.
- [7] Utvikler av den motoriserte ortosen, Professor Terje Kristoffer Lien.
- [8] Konstruksjonsskisse fra Terje K. Lien, redigert til å samsvare med den ferdiglagde ortosen.
- [9] Martin Fowler. *UML Distilled, Third Edition*. 2004.
- [10] Figur fra Terje K. Lien, oversatt til norsk.
- [11] Øyvind Stavdahl. Systemutvikling i henhold til "V-modellen". Lysark for TTK4235 Tilpassede datasystemer. 2016.
- [12] www.tutorialspoint.com/software_testing_dictionary/v_model.htm. *Sist besøkt: 12.05.18*, 2018.
- [13] [Www.electrical-engineering-portal.com/download-center/books-and-guides/power-substations/fat-sat-power-plant](http://www.electrical-engineering-portal.com/download-center/books-and-guides/power-substations/fat-sat-power-plant). *Sist besøkt: 10.05.18*.

- [14] www.geeetech.com/voice-recognition-module-p-803.html. *Sist besøkt: 12.05.18.*
- [15] www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf. *Sist besøkt: 12.05.18.*
- [16] EasyVR Shield 3.0 - Voice Recognition Shield, www.sparkfun.com/products/13316.
- [17] VeeaR - RoboTech srl. EasyVR 3. *User Manual Release 1.0.15*, pages 1–70.
- [18] Robotech srl support, korrespondanse 19.02.18.
- [19] Sensory Inc. Rsc-4128 Speech Recognition Processor, datasheet. 2008.
- [20] Dong Yu. *Automatic Speech Recognition: A Deep Learning Approach*. London, 2015.
- [21] R.C. Simpson and S.P. Levine. Voice control of a powered wheelchair. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(2):122–125, 6 2002.
- [22] Boyle L. N. Lee J. D. Chang C-C. Venkatraman V. Gibson M. Jenness, J. W. and D. Kellman. In-vehicle voice control interface performance evaluation. 2016.
- [23] J. Norberto Pires. Robot-by-voice: experiments on commanding an industrial robot using the human voice. *Industrial Robot: An International Journal*, 32(6):505–511, 2005.
- [24] Arduino. Arduino Uno R3 datasheet.
- [25] Atmel Corporation. ATmega168 datasheet. 2016.
- [26] Ulli Sommer. *Franzis Arduino tutorial kit manual*. 2014.
- [27] Store norske leksikon, www.snl.no (Signalteori, Aliasing, Sampling).
- [28] www.allaboutcircuits.com/textbook/direct-current/chpt-16/voltage-current-calculations/. *Sist besøkt: 10.05.18.*
- [29] www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/. *Sist besøkt: 10.05.18.*
- [30] www.nato.int/nato_static_fl2014/assets/pdf/pdf_2018_01/20180111_nato-alphabet-sign-signal.pdf. *Sist besøkt: 10.05.18.*