Styresystem for kybernetisk håndledd

Geir Turtum

21. desember 2013

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk
Institutt for teknisk kybernetikk

Prosjektoppgave

Studentens navn: Geir Turtum

Fag: Teknisk kybernetikk

Tittel (norsk): Styresystem for kybernetisk håndledd

Tittel (English): Control System for a Cybernetic Wrist

Beskrivelse:

Instituttet har sammen med University of New Brunswick, Kanada, stått sentralt i utviklingen av et motorisert protesehåndledd med unike kinematiske egenskaper. Det foreligger et maskinvaredesign og en prototyp av styresystemet, og dette systemet skal nå ferdigstilles og programmeres slik at protesen kan brukes i forskningsssammenheng. Sentrale systemegenskaper inkluderer kommunikasjon over en CAN-buss ved hjelp av protokollen PDCP, noe signalbehandling og styring av en børsteløs DC-motor.

1. Gi en kort oversikt over prosjektets bakgrunn og nåværende status, der du legger vekt på systemets tilstand relativt dets spesifikasjoner og tiltenkte anvendelse.
2. Foreliggende HW-design er basert på en enkelt mikrokontroller som skal stå for både motorkommutering, regulering, signalbehandling og kommunikasjonsprotokoll(er). Foreta en vurdering av om dette designet er egnet, med spesiell vekt på avbruddshåndtering og sanntidsaspekter. Gjør om nødvendig praktiske målinger for å underbygge din konklusjon.
3. Foreslå på bakgrunn av punkt 2 en maskin- og programvarearkitektur for systemet. Eksisterende løsninger bør gjenbrukes i den grad det er mulig og hensiktsmessig.
4. Implementer og test systemet så langt tiden tillater det.

Veileder(e): Øyvind Stavdahl, Institutt for teknisk kyberentikk

Trondheim, August 2013

Øyvind Stavdahl

Sammendrag

Denne prosjektoppgaven tar for seg design og utvikling av styresystemet til håndleddsprotesen NRWD(NTNU Rotary Wrist Device). NRWD ble i sin tid skapt på bakgrunn av doktorgradsavhandingen[2] til Øyvind Stavdahl.

Det har blitt skrevet masteroppgaver[6, 3] om NRWDen hvor alle har hatt som mål å realisere en prototype av protesen, men på grunn av ulike årsaker ikke har fullført. Denne oppgaven baserer seg i hovedsak på arbeidet til Andreas Kråkenes [3]. Han har produsert maskinvare lagt ut på flerlagskort, samt skrevet programvare for testing av maskinvaremodulene. Dette la et meget godt grunnlag for videre arbeide.

I denne prosjektoppgaven har styresystemet blitt ferdigstilt og det har blitt gjort anbefalinger til endringer i maskinvare der man har sett muligheter for forbedringer.

Kommunikasjonen med protesen går over standard analoge og digitale elektroder og med den åpne protokollen PDCP(Prosthetic Device Communication Protocol). Implementeringen av protokollen er gjort av tidligere NTNU studenter[4, 5] mens undertegnede har stått for integreringen av protokollen med protesens styresystem.

Anerkjennelse

Jeg vil takke min veileder, Øyvind Stavdahl. Som selv under vanskelige omstendigheter, tokk seg rikelig med tid til sine studenter. Jeg vil også takke mine foreldre som kom med verdifulle tilbakespill under skrivingen.

Nomenklatur

- NRWD - NTNU Rotary Wrist Device
- PDCP - Prosthetic Device Communication Protocol
- CAN - Controller Area Network
- PWM - Pulse Width Modulation
- UNB - University of New Brunswick
- ISR - Interrupt Service Routine
- LDO - Low Dropout Regulator
- BLDC motor - Brushless DC electric motor

Innhold

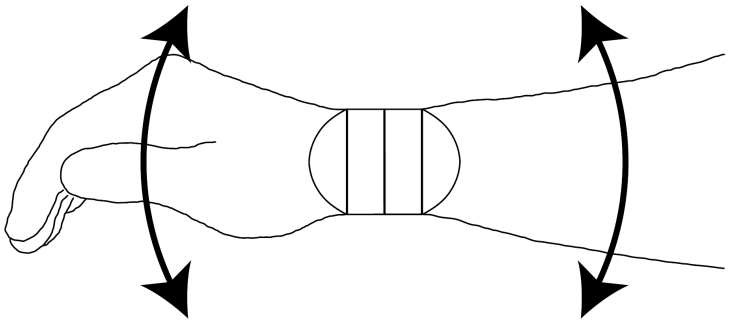
Kapittel 1

Introduksjon

1.1 Bakgrunn for oppgaven

Ideen bak protesen kommer fra Øyvind Stavdahls doktorgradsavhandling [2] fra 2002. I den viste eksperimenter med friske håndledd at en vinkel mellom lengdeaksen til underarm og rotasjonsaksen til håndleddet gjorde flere bevegelser enklere for brukeren å utføre.

For å teste disse resultatene i praksis trenger man en 1-akse håndleddsprotese med design som muliggjør valgfri rotasjonsaske, se figur 1.1. Dette ble bygget som en del av mastergraden til Arthur Zink[1] ved University of New Brunswick, Canada. Etter at mekanikken var ferdigstilt trengte man et styresystem for å styre protesen.



Figur 1.1: Vinkel mellom håndprotese og underarm kan varieres. Kilde: Styresystem for kybernetisk håndleddsprotese[3]

Det ble også funnet ut at dette var en god anledning til å teste ut den åpne protesestyringsprotokollen PDCP, som da var under utvikling hos UNB. Som krav i funksjonsspesifikasjonen til NRWDen ble det satt at den skulle

kommunisere ved hjelp av denne protokollen. I tillegg skulle den kunne styres med de mer konvensjonelle analoge og digitale elektrodene.

1.2 Tidligere arbeid

Realisering av styresystemet til protesen har hittil vært påbegynt i to mastergradoppgaver. Den siste av disse var skrevet av Andreas Kråkenes i 2011[3] som kom langt på vei til å fullføre designet. Hovedgrunnen til at protesen enda ikke har blitt fullført skyldes at det første designet innholdt en alvorlig arkitekturfeil. Feilen hindret kommunikasjonen mellom de to mikrokontrollerene som skulle styre protesen. Det ble brukt mye tid på å ordne opp i problemet. Det endte med at man gikk man over til en løsning hvor en mikrokontroller skulle styre hele systemet.

Da prosjektet ble overtatt av undertegnede, var alt av hardware designet og lagt ut på prototypekort. Programvare for å teste deler av systemet var ferdigskrevet, men koden var av varierene kvalitet og mye funksjonalitet var ikke implementert. Flest mangler var knyttet til de ulike regulatorene og integreringen med PDCP-protokollen.

To mastergradsstudenter ved NTNU[4, 5] laget i 2012 en implementasjon av PDCP-protokollen. UNB hadde tildigere laget en implementasjon basert på PIC mikrokontrolleren, mens denne var designet for ATMELs AVR mikrokontrollere. Implementasjonen var designet for at det skulle være enkelt å overføre koden til ny maskinvare, noe som gjorde at den egnet seg godt for dette prosjektet.

1.3 Mine bidrag

Prosjektet ble startet med gå gjennom den overleverte maskinvaren og medfølgende driverkode modul for modul. Detaljer rundt hvordan de ulike modulene ble testet er beskrevet i kapittel 5.

Den overleverte koden var av varierende kvalitet. Koden var strukturert på et fornuftig vis, med en c-fil for hver maskinvaremodul. Denne strukturen ble beholdt ut prosjektet. Selve driverkoden trengte en del mer arbeid. Det var skrevet nok til å teste basisfunksjonaliteten til hver modul, men måtte utvides med tilleggsfunksjonalitet før den kunne brukes i prosjektet.

Under testing av maskinvaren ble det funnet muligheter for forbedringer i spenningsregulator og posisjonssensormodulene. Kapittel 5 oppsummerer anbefalte endringer i maskinvaren.

Det første styressystemet som ble designet til NRWDen benyttet seg av

to mikrokontrollere for å håndtere alle oppgavene. Denne arkitekturen gikk man bort fra da man fikk problemer med kommunikasjonen mellom de to kontrollerne. I det nye designet gikk man over til å bruke en mikrokontroller. Som en del av oppgaven ble det gjort grundige undersøkelser om den valgte mikrokontrolleren hadde nok ressurser til å styre systemet på egen hånd. Diskusjonen rundt dette er gjort i kapittel 4.

Ifølge funksjonsspesifikasjonene skal protesen ha minst tre forskjellige styringsmoduser, maksimum fart i valgt retning, posisjons- og hastighetsstyring. Koden til de forskjellige regulatorene ble alle skrevet fra bunnen av. For detaljer om hvordan de ulike regulatorene ble implementert, se kapittel 6.

På slutten av prosjektet ble styringssystemet integrert med kommunikasjonsprotokollen PDCP. Implementasjonen av protokollen var allerede gjort, jobben besto av å skrive om maskinvare abstraksjonslaget. Hvorfor og hvordan dette ble gjort, er beskrevet i kapittel 6.

1.4 Utviklingsmetodikk og fremgangsmåte

Dette prosjektet består av et maskin- og programvaresystem som allerede da det ble overtatt var delt opp i klare moduler. Dette ble utnyttet i planleggingsfasen. Som nevnt tidligere var det meste av maskinvare designet ferdig, mens den gjensto å ferdigstille programvaren. Det var også mye programvare som var skrevet ferdig av tidligere studenter som skulle flettes inn i prosjektet for ikke å reprodusere arbeide som var gjort tidligere. Det tenkes spesielt på lavnivå implementasjonen av PDCP protokollen som ble utviklet av Andrzej Zamojski [5] og Andreas Nordal[4] samt programvare driverne til NRWDen skrevet av Andreas Kråkenes.

Metoden min sin hovedlinje var i størst grad arbeide på prosjektet modul for modul, forsikre meg om at alt fungerte atskilt, for til slutt å sette modulene sammen til et komplett styresystem. Hvordan hver modul ble testet blir gått gjennom i kapittel 5. Kort oppsummert var planen:

- Sette seg inn i bakgrunnsstoff for oppgaven.
- Test hver maskinvare modul for seg, samt tilhørende programvare driver.
- Sjekk om maskinvare og programvare oppfyller kravene satt av funksjonsspesifikasjonen.
- Fullfør driverkoden til modulen som blir testet.

- Sett seg inn i implementasjonen av PDCP gjort av tidligere NTNU studenter.
- Integrere PDCP med styringssystemet.

Verktøyet som ble brukt for å budsjettere tiden og sørge for at man kom i mål med prosjektet til rigktig tid var, på anbefaling fra veileder, Gantt diagrammet. Figur 1.2 viser framgangsplanen for de første ukene av prosjektet og hvorvidt planen ble overholdt. Gantt diagrammet var et nyttig verktøy å bruke, ikke bare ga det en klar oversikt over hvilke oppgaver man hadde igjen, men også hvorvidt man greide å overholde de tidsfrister man hadde satt.

Forklaring:	Planlagt framdrift						
	Status						
	Nåværende uke						
Uke		35	36	37	38	39	40
Oppgave	Deloppgave						
Forberedelser							
	Lage ukast til fremdriftsplan						
	Disposisjon rapport						
Bakgrunnsutredning							
	Bakgrunnsstoff						
	Tidligere prosjekt/master oppgaver						
Embedded design							
	Gå over spesifikasjonene						
	Testing av eksisterende hardware moduler:						
	-Spenningsregulator og mikrokontroller						
	-Motordriver						
	-CAN-bus						
	-Strømovervåkning						
	-Posisjonsmåling						

Figur 1.2: Gantt diagram

Utviklingen av programvare ble gjort på en windows maskin med Atmel Studio 6. For versjonskontroll av kode og dokumenter falt valget på Git.

Kapittel 2

NRWD Spesifikasjoner

Dette kapittelet går gjennom funksjonsspesifikasjonene til NRWDen punkt for punkt. Spesifikasjonene beskriver kravene til systemet i sin hellhet fra de fysiske dimensjonene, kommunikasjonsgrensesnittet opp mot protesen til oppløsningen på de ulike sensorene og maksimum vinkelhastighet til protesen.

Delkapittlene som følger tar for seg hver sin del av funksjonsspesifikasjonene. De beskriver hvike av kravene som er oppfylt, de som enda ikke er implementert og de som krever mer arbeid. Kapittelet beskriver systemet i sin nåværende tilstand og er ment for å hjelpe neste person som skal arbeide på prosjektet til å få en rask oversikt over hva som er gjort og hva som må gjøres. På noen av funksjonskravene vil statusen være kommentert som "Ikke relevant", dette gjelder de spesifikasjonene som omhandler de rent mekaniske sidene av systemet.

Ikke inkludert i kapittelet er de overordente funksjonsspesifikasjonene til NRWDen. Disse kan leses i det originale spesifikasjonsdokumentet vedlagt denne oppgaven, se vedlegg B.

2.1 Wrist Joint Function, WJF

Beskriver det roterende leddet i protesen og hvordan festeanordningen til underarm og eventuell håndleddsprotese skal være manuelt justerbar.

Nr	Beskrivelse	Status
WJF-01	The NRWD joint shall be a single, simple revolute joint which axis of rotation can be placed at an attitude with respect to the forearm and terminal device as spedcified in [1], Equations (7.4) and (7.5).	Ikke relevant
WJF-02	The joint axis should be manually adjustable to other attitudes than that specified in WJF-01.	Ikke relevant
WJF-03	The joint shall enable an angular excursion of at least 180 degrees. The excursion should be unlimited.	Ikke relevant

2.2 Wrist Motor Function, WMF

WMF delen av funksjonsspesifikasjonen beskriver kravene til protesens vinkelhastighet og moment. Viktig å merke seg at det her settes krav til at motoren skal være beskyttet mot overopphetning i maskinvare.

Nr	Beskrivelse	Status
WMF-01	The wrist joint (output of the gear train) shall have a maximum angular velocity of at least 1.4 rad/s (81 deg/s). The maximum velocity should be as high as possible.	Høyeste målte vinkelhastighet er målt til 2.72 rad/s (156 grader/sekund)
WMF-02	The wrist joint should have a maximum torque of at least 34,3 mNm.	Ikke målt, men ifølge målinger gjort av Zink [1] er stall torque på 59.6 mNm.
WMF-03	The motor shall have a maximum mechanical output power of at least TBC W.	NA
WMF-04	The motor shall be protected from overheating. The protection should be implemented by hardware.	Dette har blitt implementert i strømovervåkings modulen.

2.3 Wrist Servo Function, WSF

WDF delen av funksjonsspesifikasjonen beskriver kravet til posisjonssensoren og hvilke regulatorer som systemet skal implementere.

Nr	Beskrivelse	Status
WSF-01	The movements of the wrist joint shall be controllable according to the follwing modes: • On/Off-mode • Position mode • Velocity mode	Alle kontrollerne er Implementert i controller.c
WSF-01-01	In On/Off-mode the motor shall be at rest or run at maximum speed (open-loop) in one direction according to a given setpoint.	Implementert ved bruk av PI-kontroller som kjører motoren nærmest mulig maksimum tillatte strømforbruk.
WSF-01-02	In position mode the joint angle shall be proportional to a given angular setpoint.	Implementert med PI-kontroller.
WSF-01-02-01	The WSF shall include an absolute position sensor. This sensor shall provide no less than 10 bits resolution per revolution.	En 12-bits magnetisk enkoder er brukt.
WSF-01-03	In velocity mode the joint angular velocity shall be crudely proportional to a given velocity setpoint.	Implementert med PI-kontroller.
WSF-01-03-01	The WSF shall include a velocity sensor or estimator.	Hastighet er estimert ved hjelp av Hall-sensorene.
WSF-02	The movements of the wrist joint should be controllable according to the follwing modes: • Torque mode • Impedance mode	Ikke implementert.
WSF-02-04	In torque mode the motor torque shall be proportional to a given torque setpoint.	Ikke implementert.
WSF-02-04-01	The WSF should include means for monitoring motor current.	Implementert, se strømovervåknings modulen.
WSF-02-05	In impedance mode the mechanical impedance of the joint shall be determined by a given impedance setpoint	Ikke implementert.
WSF-03	WSF shall provide an interface to WCF through which the modes (described in WFS-01 to WSF-02) can be selected and relevant parameters can be set, and through which WSF can report relevant state variables TBD.	Koden er lagt opp for at dette enkelt kan bli implementert når kommunikasjons spesifikasjonen er klargjort.

2.4 Wrist Communication Function, WCF

WCF delen av funksjonsspesifikasjonen beskriver hva slags kommunikasjons-grensesnitt systemt skal implementere.

Nr	Beskrivelse	Status
WCF-01	The NRWD shall have a two-wire Proximal Communication Interface (PCI) and a two-wire Distal Communication Interface (DCI).	Implementert ved CAN-bus.
WCF-02	The PCI shall be configurable so that it implements two (0V, 7,2V) analog input lines or a bidirectional twowire CAN interface with the PDCP protocol(Losier,2009).	Implementert ved CAN-bus og PDCP-protokollen.
WCF-02-01	The analog input lines shall be able to sample both lines at a rate of 1 kHz. The sampling rate should be as high as 2 kHz.	Implementert i analog.c
WCF-03	The DCI shall be configurable so that it implements two (0 V, 7.2 V) analog output lines or a bidirec[1]onal two-wire CAN interface with the PDCP-protocol.	Implementert ved CAN-bus og PDCP-protokollen. Implementasjonen av PDCP-protokollen er ikke ferdig.
WCF-04	The WCF shall be configurable to an all-Analog mode, with the PCI as an analog input interface and the DCI as an analog output interface.	Analog output interface ikke implementert i verken HW eller SW.
WCF-05	The WCF shall be configurable to an all-digital mode, with the PCI and the DCI acting as bidirectional CAN bus interfaces.	Implementert. CAN buss trenger bare et tilkoplingspunkt.
WCF-07	The WCF should be configurable to a hybrid mode in which the PCI acts as two analog input lines while DCI acts as a bidirectional CAN interface (cf. WCF-01)	Ikke implementert i HW.
WCF-10	The WCF shall include a serial interface for downloading software and for debugging/diagnostic purposes.	Implementert ved bruk av JTAG.
WCF-11	WCF shall implement an interface to WSF according to WSF-03.	Se WSF-03.

2.5 Wrist Power Function, WPF

WPF delen av funksjonsspesifikasjonen beskriver kraftforsyningen til systemet.

Nr	Beskrivelse	Status
WPF-01	The WPF shall accept external power in the form of an unregulated two-wire DC supply.	Implementert ved brukt av intern spenningsregulator.
WPF-02	The NRWD shall tolerate and run normally when powered with a voltage in the range (6 V, 12 V). The range of usable voltages should be as wide as (5 V, 18 V).	Valgte spenningsregulator opererer normalt innenfor området (7V, 18V). Det er anbefalt å gå over til en Low Droupout Regulator for å tilfredstille kravet.
WPF-03	The NRWD shall tolerate supply voltage in the range (0 V, 12 V) without exhibiting unpredictable behaviour and without getting damaged.	Ikke implementert.
WPF-04	The NRWD should automatically limit its motor current to a level that does not reduce the supply voltage below the interval given in WPF-01.	Ikke implementert.

2.6 Proximal Attachment Function, PAF

PAF delen av funksjonsspesifikasjonen beskriver festeanordningen og grensesnittet mellom NRWDen og underarmen.

Nr	Beskrivelse	Status
PAF-01	The PAF shall comprise two parts, the proximal of which is adapted to be permanently attached to the forearm socket and the distal permanently attached to the wrist unit. The parts must ?mate? to form a mechanically stable connection while also being detachable.	Ikke relevant
PAF-01-01	The proximal part of the PAF shall be hollow to allow access to the space within the socket proximally to the wrist.	Ikke relevant
PAF-02	PAF disconnection should be possible with hand or a simple tool, e.g. a screwdriver.	Ikke relevant
PAF-03	The PAF shall include a four-wire electric coupling, preferably mechanically integrated with the PAF itself.	Ikke implementert i prototype hardwaren.
PAF-03-01	The PAF electrical coupling shall include at least two power supply wires/contacts capable of transferring a constant current of 4 A per wire.	Ikke implementert i prototype hardwaren.

2.7 Distal Attachment Function, DAF

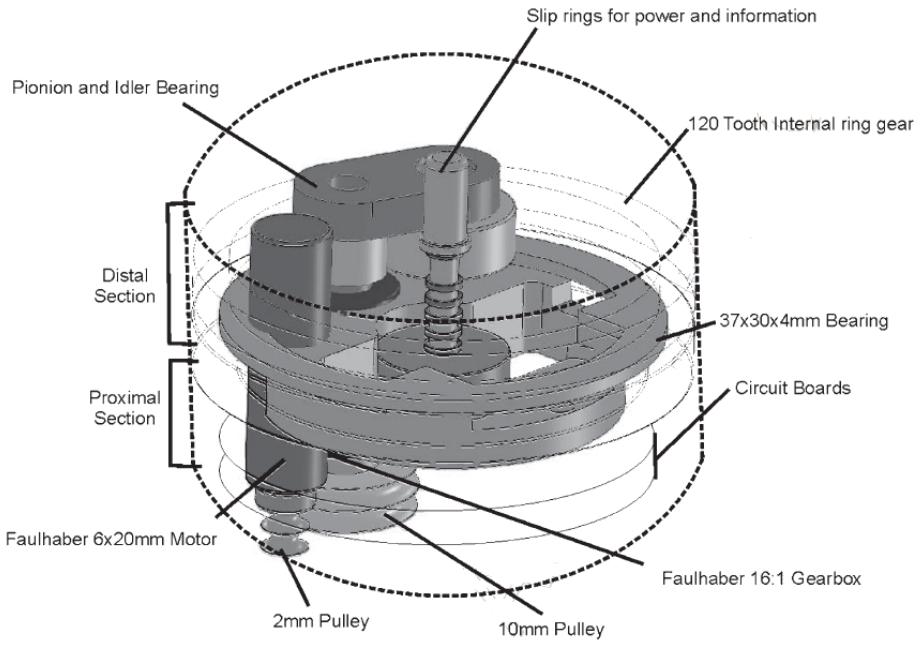
DAF delen av funksjonsspesifikasjonen beskriver festeanordningen og grensesnittet mellom NRWDen og en eventuell håndprotese.

Nr	Beskrivelse	Status
DAF-01	The DAF should comprise two parts, the proximal of which is permanently attached to the wrist and the distal permanently attached to the terminal device. The parts must mate to form a mechanically stable connection while also being detachable.	Ikke relevant
DAF-02	The DAF shall include a four-wire electric coupling, preferably mechanically integrated with the DAF itself.	Ikke implementert i prototype maskinvaren.
DAF-02-01	The DAF electrical coupling shall include at least two wires/contacts capable of transferring a constant current of 2 A per wire, and these wires shall be connected to the power supply wires from PAF.	Ikke implementert i prototype maskinvaren.
DAF-02-02	The DAF electrical coupling should be rotatable without twisting the wires.	Ikke implementert i prototype maskinvaren.

Kapittel 3

Mekanikken

Mekanikken i protesen er et resultat av en masteroppgave skrevet ved universitetet i New Brunswick, Canada. Mekanikken består av to seksjoner, den proksimale og distale (henholdsvis seksjonen nærmest underarmen og den lengst unna). De to seksjonene kan rotere fritt mot hverandre ved hjelp av et kulelager. Selve bevegelsen er drevet av en børsteløs DC-motor som er koplet til et gir tog med utveksling 333:1, se seksjon 5.4.



Figur 3.1: Oversiktbilde over den indre mekanikken til protesen. Figur hentet fra [1]

Under testing av motordriveren ble det funnet at det, ved en forsyningsspenning på 12V, måtte til en dutycycle på 60% for å bevege mekanikken. Ved rundt 80% dutycycle ble strømmen til motoren kuttet av strømovervåkingsmodulen, da strømmen gjennom motoren oversteg maksimum grensen. Denne grensen er satt i maskinvare for å unngå at motoren blir ødelagt på grunn av overopphetning. Dette ga regulatoren et veldig lite rom å jobbe innenfor, noe som gjorde implementasjonen av posisjon og hastighetskontrolleren meget vanskelig.

Mekanikken ble tatt med til mekaniske verksted ved NTNU for å se om de kunne løse problemet. Det viste seg at aksen til det minste tannhjulet var bøyd, noe som førte til ekstra friksjon i systemet. Tannhjulet det er snakk om er det som er koblet til Faulhaber girboksen i den distale seksjonen, se figur 3.1. Verkstedet fikk bøyd det til slik at det gikk fra å ha en kast på 0.22 mm til 0.03 mm. Dette reduserte friksjonen betraktelig og mekanikken kunne nå beveges med en dutycycle ned mot 40%.

Da mekanikken ble undersøkt ble det også funnet at kulelageret i protesen begynte å bli slitt, noe som førte til yterligere friksjon i systemet.

Om rettingen av giraksen løser problemet på langt sikt er usikkert. At problemet har oppstått tyder på at mekanikken ikke er riktig dimensjonert for de krefter den kan bli utsatt for. En teori på hvordan problemet kan ha oppstått, er at man har vridd på den distale seksjonen, noe som påfører store krefter på den indre mekanikken. Dette tyder på at mekanikken kan ha behov for en ny revisjon før den er robust nok til å kunne brukes i eksprimenter.

For fremtidig bruk er det lagt ved en oversikt over rotasjonshastigheten til protesen ved forskjellige dutycycler, se vedlegg C.1. Målingene er gjort etter at mekanikken var rettet av verkstedet. Denne kan brukes for å se om problemet har gjenoppstått.

Kapittel 4

Interrupt diskusjon

4.1 Bakgrunn

Tidligere forsøk på å realisere styresystemet til NRWDen har brukt to mikrokontrollere. Den ene skulle brukes til motor kommuteringen, mens den andre skulle håndtere regulering, signalbehandling og kommunikasjonsprotokollene. Da systemet ble testet oppsto det problemer med kommunikasjonen mellom mikrokontrollerene. Selv etter mye testing fant man aldri løsningen til problemet. Dette førte til at man gikk over til en arkitektur som brukte en enkelt mikrokontroller.

Spørsmålet var da om den valgte mikrokontrolleren, AT90CAN128 kjørende på 16 MHz, var nok til å håndtere alle oppgaver som før hadde blitt fordelt på to mikrokontrollere. Dette kapitlet tar for seg denne problemstillingen.

4.2 Et interrupt drevet system

Så hva slags type system er det som styrer protesen? Styringssystemet er nesten utelukkende interrupt drevet. Bortsett fra regulatoren som blir kjørt inni main while løkken er resten styrt med interrupts. Motorkommuteringen, kommunikasjonsprotokollen, fartsestimeringen, posisjonsavlesningen og strømovervåkingen blir alle styrt gjennom interrupts. Hver av disse oppgavene krever at tiden mellom et interrupt blir generert og mikrokontrolleren går inn i tilhørende ISR(Interrupt Service Routine) ikke overstiger en viss lengde.

Om et interrupt ikke blir håndtert innenfor tidsfristen en eller flere ganger, vil systemet i beste fall bli mindre nøyaktig, i verste fall vil det slutte å fungere. Som et eksempel kan man ta for seg posisjonssensoren. Posisjonen er proporsjonal med dutycycelen til sensorsignalet. Denne leses av ved å måle

tiden mellom positiv og negativ flanke på signalet. Hver endring i nivå hos signalet genererer et interrupt, se figur 5.3. Hvis tiden det tar for mikrokontrolleren å behandle interruptet tar lengere tid enn forventet vil posisjonen bli feil avlest. Man må kunne gi en garanti for at mikrokontrolleren har nok ressurser til å håndtere alle interruptene som blir generert. Ellers vil man risikere at man får et system som virker tregt og uresponsivt eller slutter å virke. Systemet har derfor blitt analysert for hvor mye tid man bruker inni ISRene.

Typen interrupts som systemet skal håndtere kan deles inn i to typer, periodiske og aperiodiske.

De periodiske er som følger:

- HALL_X_vect - Genereres hver gang en av de tre Hall-sensorene i motoren skifter nivå
- TIMER3_CAPT_vect - Genereres på hver flanke til PWM signalet fra posisjons sensoren
- TIMER0_OVF_vect - Brukes til å beregne hastigheten til motoren

De aperiodiske er:

- CANIT_vect - Genereres hver gang det kommer en CAN melding
- CURRENT_vect - Genereres når maskinvaren slår av strømmen til motoren
- RESET

For å analysere systemet ble varigheten til hver interrupt målt. Et interrupt i mikrokontrolleren består av tre deler. Det er kontektsskiftet hvor mikrokontrolleren lagrer unna stacken før man går inn i ISRen, arbeidet som blir utført inni rutinen og gjennopprettingen av stacken i det man forlater den. Tiden det tar for mikrokontrolleren å utføre de to kontekssskiftene er et fast antall instruksjoner. Det faktiske tallet finnes i mikrokontrollerens datablad [8]. Varigheten til hvert interrupt ble målt ved å sette en GPIO-pinne høy ved inngangen av interruptet og lav i det den gikk ut. Hvor lenge utgangen til GPIO-pinnen var høy ble målt med et oscilloskop. Se tabell 4.1 for oppsumering av tidsbruk til de periodisk interruptene.

Vi ser av tabell 4.1 at mikrokontrolleren bruker mindre enn 2% av tiden inne i en interrupt rutine. Det gir rikelig med tid til å håndtere kontrolloopen som kjører utenfor interruptene. Dette garanterer derimot ikke at alle interrupts vil bli håndtert innenfor rimelig tid. I et tilfelle hvor en eller flere av

Navn	Varighet (μs)	Maks frekvens	Tidsbruk per sekund(μs)
HALL_X_vect	4.5	3600	16200
TIMER3_CAPT_vect	4.8	490	2352
TIMER0_OVF_vect	3.2	61	196
Totalt			18742

Tabell 4.1: Viser varigheten til hvert interrupt, hvor mange ganger de blir kjørt i sekundet og totalt tidsbruk per sekund.

modulene genererer et interrupt samtidig, kan dette føre til midlertidige feil i systemet. Siden dette ikke er et hardt sanntidssystem, hvor overskridelsen av en tidsfrist hadde gjort systemet ubrukelig, er disse potensielle midlertidige feilene akseptable.

Et annet punkt er virkemåten til ISRene i dette systemet. Resultatet til en ISR er ikke avhengig av resultatet fra forrige gang den kjørte. Hvilket betyr at om en ISR gir ut feil resultat, vil dette være ordnet neste gang den kjører. Underforstått at det bare var en sporadisk forsinkelse som førte til feilen og ikke en permanent feil ved systemet. Da ISRene opptar under 2% av prosessortiden og de alle operere ved ulike frekvenser, vil kollisjon mellom to eller flere ISR være en sjelden hendelse.

Det ble også undersøkt om mikrokontrolleren er rask nok til å håndtere kommuteringen av motoren. En den for treig vil den ikke kunne kommutere raskt nok til at motoren oppnår den ønskede hastighet. ATMELs applikasjonsnote AVR452 tar for seg motor styring av børste løse DC motorer ved bruk av AT90CAN mikrokontrolleren. Den setter en teoretisk grense ved 3'478'000 rpm ved 8 MHz, da maksfarten til NRWD motoren er satt til 3'600 rpm og mikrokontrolleren kjører på 16 MHz befinner vi oss godt innenfor kravene.

Kapittel 5

Maskinvaretesting og diskusjon

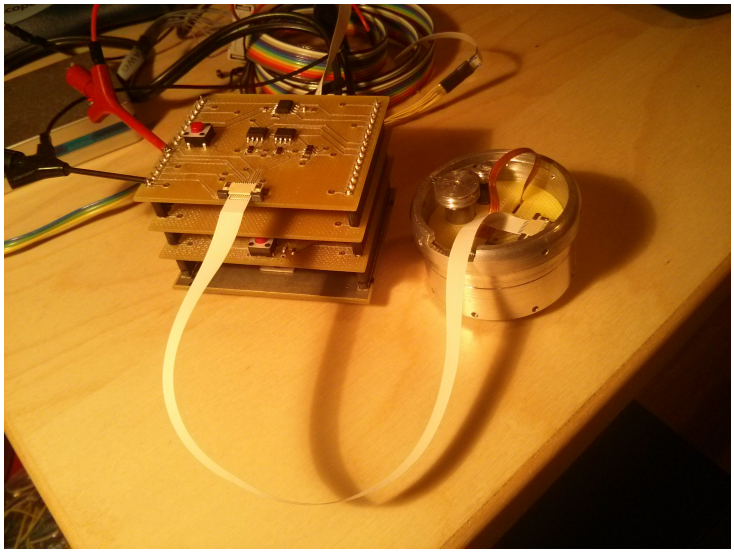
Dette kapittelet går gjennom maskinvaren som dette prosjektet har basert seg på. Det blir forklart hvordan hver modul har blitt testet og eventuelle forslag til forbedringer blir gitt.

Dette prosjektet startet ikke fra grunnen av, men bygger videre på arbeidet til flere studenter. Således var det mye dokumentasjon, maskinvare og programvare som måtte gjennomgåes før det kunne bygges videre på. Det tenkes her spesielt på arbeidet til Andreas Kråkenes som var siste student som arbeided på NRWDen. Han sto for utlegg og produksjon av maskinvareprototypen, se figur 5.1, som har blitt brukt under dette prosjektet.

Alt av overlevert maskin- og programvare ble gjennomgått og testet modul for modul for å se om det fungerte i henhold til det som sto i Kråkenes' oppgave[3]. Før arbeidet startet ble en kort testplan for hver modul utarbeidet. Den ble laget slik at da alt i planen var testet, skulle man være sikker på at maskin- og programvaren for valgt modul oppførte seg som forventet. Under følger resultatene av testingen, samt en diskusjon rundt hver modul. For flere detaljer rundt maskinvaren og skjemategninger, se Kråkenes' oppgave[3].

5.1 Mikrokontroller

Styringen av protesen er gjort med en ATMEL AT90CAN128 AVR mikrokontroller. Opplasting av programvare og debugging ble gjort over JTAG via ATMELs programmerinsverktøy, JTAGICE 3. Det ble opprettet kontakt med AVRen og alt fungerte som forventet.



Figur 5.1: Prototypen av NRWD maskinvaren produsert av Andreas Kråkenes, her vist tilkoplet protesemekanikken.

5.2 Spenningsregulator

Ifølge funksjonskravet WPF-02 skal NRWDen operere normalt når den blir drevet av en spenningsforsynning mellom 6 og 12V. Spesifikasjonene sier ikke noe om hvilke spenninger som skal brukes internt, men den valgte mikrokontrolleren trenger 4.5V for å kjøre på 16MHz. Det har derfor blitt valgt at den eksterne spenningen skal reguleres ned til 5V.

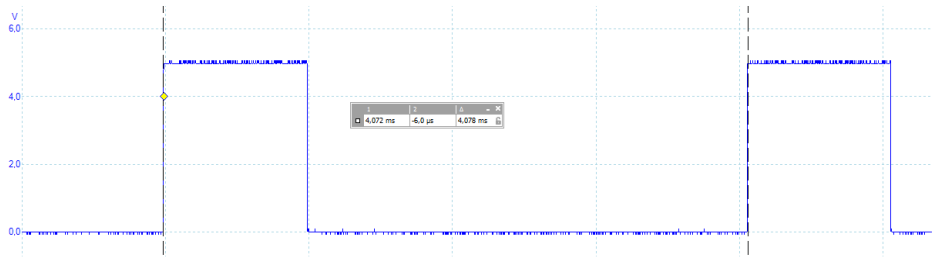
Spenningsregulatoren brukt i NRWDen er en linær regulator som i henhold til databladet gir ut 5±0.2V ved inngangsspenninger 7-20V. Det er viktig at NRWDen kan operere på spenninger ned til 6V da de vanligste batteriene som er brukt i slike proteser gir ut en spenning som kan gå ned til 6V. Da differansen mellom inngangs- og utgangsspenning kan være så lav som 1V, anbefales det å gå over til å bruke en LDO (Low Dropout Regulator). En LDO er spesielt designet for å operere under slike forhold. Et eksempel på en LDO som kan brukes er LF50CDT fra STMicroelectronics.

5.3 Posisjonsmåling

Ifølge funksjonskravet WSF-01-02-01 skal NRWDen være utstyrt med en absolutt posisjonssensor med en oppløsning på minst 10 bit.

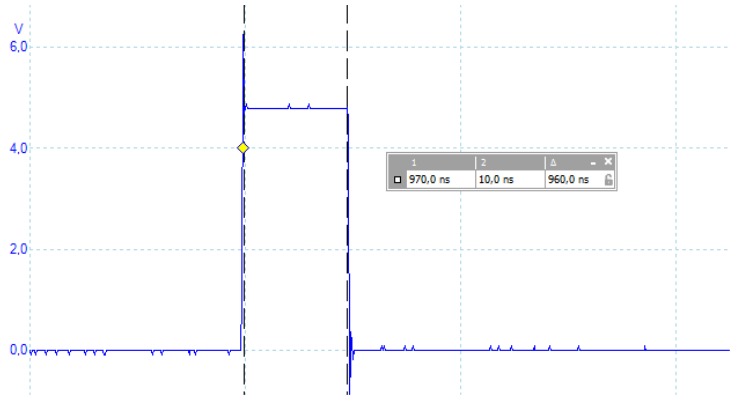
Den nåværende løsningen er en magnetisk enkoder. Den leser av magnetfeltet til en magnet festet på den distale delen av NRWDen. Sensoren gir ut et

PWM signal hvis dutycycle er proporsjonalt med vinkelen til magnetfeltet på sensoren, se figur 5.2. Hver flanke på PWM-signalet generer et interrupt og tiden mellom en stigende og synkende flanke blir målt av mikrokontrolleren.



Figur 5.2: Utgangen til posisjonssensoren målt med oscilloskop

Problemet med dette oppsettet kommer ved vinkler nær 0 og 360 grader. Ved disse vinklene vil tiden mellom to flankeendringer være under 1 μs, se figur 5.3. Som beskrevet i kapittel 4 bruker mikrokontrolleren 4.8 μs å behandle interruptet fra posisjonssensoren. Dette vil si at det andre interruptet som kommer enten ikke vil bli behandlet, eller bli behandlet for sent, noe som vil gi feil avlest vinkel.



Figur 5.3: Utgangen til posisjonssensoren ved lavest dutycycle

Foreløpig er problemet løst i programvaren, men en bedre løsning ville vært å gjøre det i maskinvaren. En god løsning ville vært å lavpassfiltrere PWM-signalet for så å lese av vinkelen med en ADC. Dette løser ikke bare problemet med feil avlesning av vinkel nær 0 og 360 grader, men det vil også redusere antall interrupts som mikrokontrolleren må behandle. PWM-signalet har en frekvens på 245 Hz, noe som gir 490 interrupts i sekundet. Ved bruk av et lavpassfilter og en ADC vil man bare trenge å kalkulere posisjonen

hver gang posisjonsregulatoren skal kjøre. GPIO pinnene på mikrokontrolleren har allerede mulighet for å aktivere en intern pullup motstand. Dette betyr at den eneste eksterne komponenten som kreves for å lage et lavpassfilter er en kondensator med passende verdi.

5.4 Motordriver

Protesens bevegelse er drevet av en børsteløs DC-motor med innebydge Hallsensorer. Ved å lese av Hall-sensorene vet man posisjonen til motoraksen og kan utifra det vite hvilket pådragsmønster man skal gi motoren får å få den til å rotere. Utgangen til Hall-sensorene er digitale og er koplet opp mot hver sin interrupt linje. Det blir da generert et interrupt hver gang utgangen skifter nivå.

Ifølge funksjonskravet WMF-01 skal protesen minst ha en maksimums omdreiningshastighet på 81 grader/sek. Ifølge Zink[1] er motoren giret ned med et forhold 800:1, se tabell 5.4. Dette stemmmer ikke med protesen brukt i dette prosjektet, da diameterene til drivhjulene fra motor til Faulhaber planetary gear ikke er som spesifisert hos Zink. Faktiske målinger viser at girforholdet til protesen er på 333:1.

Details of the drive train

Stage	Stage gear ratio	Cumulative ratio	Velocity after each stage (rpm)	Torque after each stage (mNm)	Notes
Motor	1:1	1:1	47000	0.73	Faulhaber 0620 K 006 B
First	5:1	5:1	9400	3.65	Belt
Second	16:1	80:1	587.5	58.4	Faulhaber 08/1K - Planetary Gear
Third	10:1	800:1	58.8	584.0	Ring and pinion 303 stainless steel Pitch: 96 Face width: 3.175 mm Pressure angle: 200

Figur 5.4: Tabellen viser girtoget til motoren slik det originalt var ment. Dette stemmer IKKE med den mekanikken brukt i dette prosjektet. Tabell hentet fra Zink, Design of a compact, reconfigurable, prosthetic wrist[1].

For å oppnå ønsket vinkelhastighet tilsvarer dette en motorhastighet på 4'496 rpm, se formel 5.1. Ifølge motordatabladet [9] er makshastigheten til motoren 35'600 rpm noe som faller godt innenfor kravet.

$$\frac{81 \text{ grader/sekund}}{360 \text{ grader}} \times 333 \times 60 \text{ sekund} = 4495.5 \text{ rpm} \tag{5.1}$$

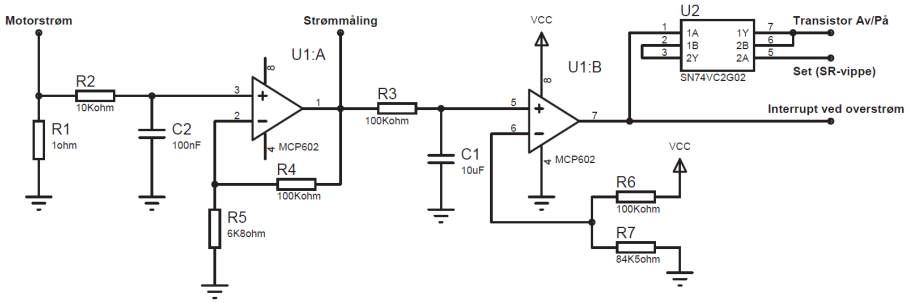
Da motorens Hall-sensorer genererer 6 interrupt per omdreining vil mikrokontrolleren måtte kunne håndtere minst opptil 3560 interrupts i sekundet, se formell 5.2.

$$\frac{35600 \text{ rotasjon/min}}{60 \text{ sek/min}} \times 6 \text{ interrupts/sek} = 3560 \text{ interrupts/sek} \qquad (5.2)$$

En testbenk ble satt opp for å teste om mikrokontrolleren greide å håndtere alle Hall-sensor interruptene. Et Arduino-brett ble brukt for å simulere utgangene til motorens Hall-sensorer med en hastighet som tilsvarte 400'000 rpm. Arduino-brettet var programmert til å påføre et bestemt antall interrupts som mikrokontrolleren telte. Under testing var antallet registrerte interrupts hos mikrokontrolleren likt antallet sendt ut fra Arduino-brettet. Da mikrokontrolleren fungerte feilfritt ved disse hastighetene kunne man være sikker på at mikrokontrolleren ville greie å håndtere kommuteringen av motoren ved de faktiske hastighetene.

5.5 Strømovervåkning

Ifølge funksjonskravene WMF-04 og WSF-02-04-01 skal motoren være beskyttet mot overopphetning i maskinvare og man skal kunne måle hvor mye strøm motoren trekker. Overopphetning skjer når motoren trekker for mye strøm. Utregningen av den maksimalt tillatte strømmen er gjort tidligere av Kråkens[3] utifra motorens datablad[9] og er satt til 145 mA.



Figur 5.5: Skjemategning til strømovervåknings modulen. Figur hentet fra Kråkenes[3]

Motoren får strøm gjennom et 20 kHz PWM-signal som går gjennom den 1 ohms store målemotstanden før det går til jord. Spenningen over målemotstanden blir lavpasfiltrert og forsterket av en operasjonsforsterker.

Deretter går signalet inn på en komparator krets som setter en SR-vippe. Vippen sørger for å slå av strømforsyningen til motoren når den trekker mer enn den maksimalt tillatte strømmen. SR-vippen kan bli resatt av mikrokontrolleren. Utgangen til operasjonsforsterkeren er også koplet til en ADC-pinne på mikrokontrolleren. Strømmen kan da måles ved å lese av ADCen ved å bruke formel 5.5.

```
/* Multiply K with ADC value to get current in 1/10 uA
* ADCref = 5 V
* Aopamp = 15.7 (opamp gain)
* K = ADCref * (100'000/1024)/(Aopamp) = 31
*/
uint16_t K = 31;
current_mA = (K*ADC)/100; //divide by 100 to get mA
```

For å teste kretsen ble det brukt en variabel strømforsyning. Modulen skrudde av strømmen ved riktig verdi og mikrokontrolleren leste av strømmen med stor nøyaktighet.

5.6 CAN

Ifølge funksjonskravet WCF-02 skal protesen være styrt over PDCP-protokollen, som baserer seg på CAN for å sende og ta imot meldinger. Prototypekortet er utstyrt med en CAN-tranceiver, som deretter er koplet opp mot mikrokontrolleren. CAN-modulen ble testet med et USB til CAN-modul. Dette gjorde det enkelt å sende meldinger fra PCen. Mikrokontrolleren ble testet ved overføringshastigheter opp til 125 kbit. Da fungerte alt som forventet.

Kapittel 6

Programvare implementasjon

Dette kapittelet beskriver hvordan programvaren er bygget opp og satt sammen. I tillegg diskuteres modulariseringen av koden og fallgruver ved programmering av et interrupt drevet system. Underkapitlene beskriver hva hver modul gjør og hvordan modulen forholder seg til resten av koden.

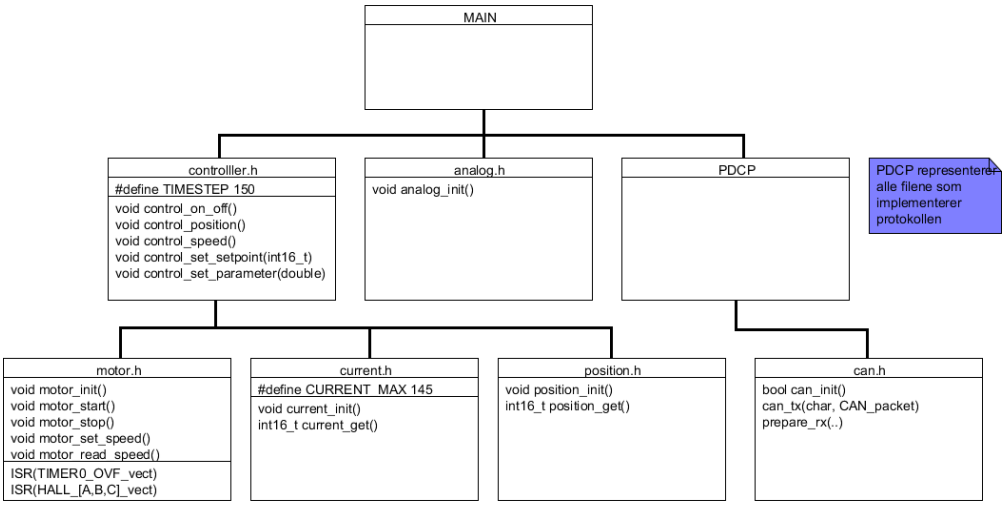
Det som i hovedsak manglet ved overtagelsen av prosjektet var å skrive ferdig programvare til NRWDen. Det var skrevet drivere til de fleste modulene hvor koden var av varierende kvalitet, skrevet mer for å teste maskinvaren enn for å kunne brukes i det ferdige produktet.

Den ferdige koden kan deles opp i to hoveddeler. Driverkoden til NRWDen som er maskinvareavhengig og høy nivå laget(HLL) til PDCP-protokollen. Driverkoden er igjen delt opp etter de ulike maskinvaremodulene. En oversikt over programvaren kan ses i figur 6.1. Her vises alle de ulike modulene og hvem som inkluderer hvem. De forskjellige maskinvaredriverne er current.h for strømovervåkning, position.h for posisjonssensoren, motor.h for motorstyring og can.h for det nederste laget av PDCP protokollen.

Selve implementasjonen av PDCP protokollen er en porting av programvare skrevet av Andreas Nordal og Andrzej Zamojski i sine masteroppgaver [4, 5].

Systemet er for det meste drevet av interrupts, både interne og eksterne. Motorkommuteringen, posisjonsmålingen, motagelser av CAN-meldinger, fartsmåling og så videre skjer alle inni interruptrutiner. Når koden som kjører til en hver tid kan bli avbrutt av en interrupt rutine er det meget viktig å kode slik at det ikke oppstår raceconditions. Den enkleste måten å forhindre dette på er å sørge for best mulig seperasjon mellom modulene, så ikke en modul kan endre på variabler brukt av en annen modul.

Under følger detaljer rundt hver modul.



Figur 6.1: Klassediagram som viser innholdet i og avhengighetene mellom de ulike programvare modulene.

6.1 Motor

Motor-modulen sørger for kommuteringen av motoren og vinkelhastighets-avlesning.

Pådraget til motoren settes ved motor_set_speed(). Denne funksjonen setter dutycyclen til PWM-signalet som driver motoren. Siden strømmen levert av dette signalet varierer med forsyningsspenningen setter man ikke vinkelhastigheten direkte. Styringen av farten til protesen skjer inni controller-modulen.

Etter at dutycyclen er satt vil Hall-sensor interrupt rutinene automatisk ta seg av kommuteringen av motoren. En timer overflow interrup rutine vil kontinuerlig oppdatere vinkelhastigheten til protesen slik at den til enhver tid kan leses av med motor_read_speed().

6.2 Current

Current-modulen bruker mikrokontrollerens ADC til å lese av strømmen som går gjennom motoren. Interruptrutinen, CURRENT_vect, som blir kjørt når maskinvaren skrur av strømmen til motoren ligger ikke her, men i main c-filen. Dette er fordi CURRENT_vect benytter seg av flere av de andre maskinvaremodulene til å rydde opp i systemet etter at motorstrømmen har blitt skrudd av.

6.3 Position

Position-modulen setter opp en timer til "input capture mode". Den er satt opp til å generere et interrupt ved endring i flanken på signalet fra posisjonssensoren. Ved en positiv flanke starter timeren og ved negativ flanke blir telleren i timeren lagret. Utifra verdien til telleren blir vinkelposisjonen til protesen regnet ut og lagret i en global variabel. Denne kan til en hver tid bli lest ut av position_read funksjonen.

6.4 Controller

Ifølge funksjonskravet WSF-01 skal bevegelsen til protesen være kontrollerbar gjennom de følgende måtene:

- On/Off-mode
- Position mode
- Velocity mode

Disse finnes implementert i controller.c som henholdsvis control_on_off(), control_position() og control_speed(). Utfordringen ved designet av regulatorene var at strømmen til motoren automatisk ble slått av hvis den oversteg en forhåndsbestemt maksgrense. En regulator som ikke hadde tatt hensyn til dette vil ikke ha gitt en god brukeropplevelse da protesemotoren ville skrudd seg av med gjevne mellomrom.

Funksjonskravet WSF-01-01 sier at On/Off-mode skal være kjørt i en open-loop kontroller. På grunn av overnevnte grunner har den istedenfor blitt kjørt i en PI-kontroller. Denne regulerer strømmen gjennom motoren tett opptil den maksimale grensen for å oppnå høyest mulig vinkelhastighet.

Fartsregulatoren løser begrensingen i motorstrøm ved å brukke en dobbel PI-regulator. Den ytre sløyfen tar inn vinkelhastighet og setter strøm referansen til den indre sløyfen. Den indre setter pådraget til motoren basert på strømreferansen og sørger samtidig for at strømmen ikke går over den maksimalt tillate verdien.

Posisjonsregulatoren fungerer på samme måte som fartsregulatoren, bortsett fra at den ytre sløfen tar inn vinkelposisjonen til protesen.

I alle regulatorene er utregningen av integralleddet uavhengig av frekvensen som regulatoren kjører på. Dette betyr at om frekvensen til regulatorløkken blir endret så må regulatorene justeres. Frekvensen til regulatorløkken er styrt med en busy-wait forsinkelse funksjon, se figur 6.2. Denne

typen forsinkelsesfunksjoner registrerer ikke når man er inni en interruptrutine. Dette betyr at regulatorløkken blir forsinket med så mye tid man bruker inni hver av interruptrutinene. Da mikrokontrolleren bruker mindre enn 2% av tiden sin inne interruptrutiner vil dette ikke medføre noen problemer for regulatoren.

```
while(1)
{
    control_controller();
    _delay_ms(TIMESTEP);
}
```

Figur 6.2: Main while loop

Verdien til P- og I-leddet til de tre regulatorene ble funnet gjennom praktiske forsøk. De gir god regulering, men det kan med fordel brukes mer tid på å justere for å få enda bedre respons.

Funksjonskravet WSF-03 setter krav til at man skal kunne skifte både regulatortype og vedien til regulatorparametrene gjennom PDCP-protokollen. Bytte av regulatortype gjøres gjennom control_set_mode() funksjonen. For å bytte parametere må man først velge regulator gjennom nevnte funksjon, deretter kan man sette P- og I leddet ved control_set_parameter() funksjonen.

6.5 Analog

Funksjonskravet WCF-04 sier at NRWDen skal kunne styres ved å bruke analoge elektroder. Dette er den normale måten å styre myoelektriskeproteser. Videre sier WCF-02-01 at de analoge linjene skal kunne samples ved minst 1 kHz. Myoelektriske signaler har en båndbredde på 500 Hz.

Analog-modulen ble ikke ferdig implementert i dette prosjektet. Hadde den blitt ferdig hadde den hatt de samme oppgavene som PDCP-modulen, ta imot eksterne signaler og styre protesen.

Da kommunikasjonen ville vært rent analogt ville den ikke hatt like rik funksjonalitet som PDCP-protokollen. Man kan tenke seg at type regulator da vil være forhåndsbestemt. Eventuelt at man kan skifte regulator ved bruk av annen input metode enn elektrodene.

6.6 PDCP and can

Når protesen er ferdig skal den kunne styres gjennom PDCP-protokollen. PDCP-protokollen er en åpen kilde protese kommunikasjonsprotokoll. Den har blitt til ved UNB i Canada og har som ambisjon og bli den nye standarden for protesekommunikasjon.

I dette prosjektet har det blitt brukt en NTNU produsert PDCP-implementasjon. Den var oppdelt i to lag, et høynivå[4] lag som implementerte selve funksjonaliteten til protokollen og et maskinvare abstraksjonslag[5] som var kodet for å være lett å tilpasse til ny maskinvare.

I masteroppgaven[5] knyttet til koden hevdes det at koden skal være lett å tilpasse til en mikrokontroller med intern CAN-kontroller. Dette var ikke min erfaring. Det ble brukt mye tid på å forsøke og kode om maskinvarelaget til å passe med AT90CAN mikrokontrolleren. Etter å ha jobbet mye med koden er min konklusjon at den er godt egnet om man bare skal skifte AVR mikrokontroller. Men fortsatt holde seg til den eksterne CAN-kontrolleren MCP2515, som koden har blitt designet rundt.

Da man ikke greide å tilpasse det eksisterende maskinvare abstraksjonslaget ble det skrevet et nytt et som implementerte basis funksjonaliteten som krevdes for å teste ut PDCP-protokollen. Denne koden finnes i hal-c filen. Bruken av den interne CAN-kontrolleren er gjort med et åpent CAN bibliteket skrevet for AT90CAN mikrokontrolleren. Biblioteket består av filene can.c og can.h.

6.7 Main

Main starter med å initialisere de ulike modulene og velge kontrollertype. Det er i main-c filen at callback funksjonen fra PDCP-modulen er plassert, se kode 6.3. Callback funksjonen blir kalt fra PDCP-modulen hver gang den motar en melding.

Da dette prosjektet ble skrevet fantes det en PDCP-spesifikasjon[7], men ingen eksempel kode som viste hvordan de ulike PDCP-funksjonskodene skulle brukes. Det har blitt laget skjelettkode for å håndtere de ulike PDCP-funksjonskodene inni callbackfunksjonen. Når eksempel kode blir gjort tilgjengelig kan funksjonaliteten til de ulike kommandoene skrives ferdig.

```
/* callback_incoming_can */
/*
* @fn           void callback_incoming_can(struct socket *so)
* @brief        Callback function from HLL
*/
void callback_incoming_can(struct socket *so){

        struct can_msg *msg = sock_pull(so);
        switch(msg->function_code){

        case REQUEST_BIND:

        break;
        case RESPONS_BIND:

        break;
        case REQUEST_GET_INFO:

        break;

        ...
```

Figur 6.3: Callback funksjonen som blir kalt fra PDCP-modulen hver gang den motar en melding.

Kapittel 7

Diskusjon

7.1 Arbeidsmetodikk

Som vår veileder flere ganger kommenterte. Et prosjekt uten noen arbeidsplan er som å regulere et system i åpen sløyfe; ting kan fort gå ut av kontroll. Planleggingen av dette prosjektet ble hjulpet ved bruk av Gantt-diagrammet. Prosjektet ble startet med å dele opp arbeidet i deloppgaver. Hver oppgave ble gitt en anslått start- og sluttdato.

Dette var svært nyttig da det tvang meg til å tenke på hvordan prosjektarbeidet skulle deles opp. Hvilke deloppgaver som bygget på hverandre og hvordan tiden skulle budsjetteres. Planen ble fulgt godt de første åtte ukene av prosjektet. Etter det gikk man mer eller mindre bort fra å bruke planen. Dette hadde flere grunner. Da planen først ble skrevet var det lettere å planlegge starten av prosjektet. Sluttfasen var mer uklar og ble dermed dårligere beskrevet i Gantt-diagrammet. Dette gjorde det vanskligere å følge planen. For det andre hadde man, etter et halvt semester, mye bedre oversikt over prosjektet og følte ikke like stort behov for detaljplanlegging.

7.2 Mekanikken

Selve mekanikken i protesen har ikke vært fokuset i dette prosjektet. Men har blitt brukt for å teste om maskin- og programvaren fungerer. Det ble tidlig oppdaget problemer med mekanikken. Dette førte til at utviklingen av de ulike regulatorene ble forsinket. Da den ble tatt med til NTNU mekaniske verksted ble det oppdaget en bøyd girakse og slitt kulelager. Disse feilene kan tyde på at mekanikken ikke er robust nok til å brukes i praktiske eksperimenter. Det stilles også spørsmål til om motoren er riktig dimensjonert for de oppgaver protesen skal kunne utføre. Under testing av regulatorene skulle

det ikke mye mekanisk motstand til før strømmen gjennom motoren oversteg maksimum grensen. Uten å ha målt det faktiske dreiemomentet til protesen anslåes det til for lite for praktiske oppgaver. Det anbefales, ved eventuell ny revisjon av mekanikken, at motoren byttes ut i en kraftigere utgave.

7.3 Implementasjon

Et spørsmål til den nye maskinvarearkitekturen var den valgte mikrokontrolleren var hadde nok ressurser til å styre hele systemet. For detaljer rundt diskusjonen henvises det til kapittel 4. Hovedsaken her var om mikrokontrolleren ville ha god nok tid til å behandle systemets mange interruptkilder. Da systemet i hovedsak er interruptdrevet er kort responstid på interrupts kritisk for at systemet skal virke. Målinger viste at mikrokontrolleren brukte mindre enn 2% av tiden inni ISRer. Dette er godt innenfor hva som er akseptabelt.

Diskusjonene rundt maskinvaretesting og progamvare er gjort i kapittel 5 og 6.

Kapittel 8

Konklusjon

Det har i denne oppgaven blitt utviklet programvare for å styre NTNUs Rotary Wrist Device, en 1-akse håndleddsprotese. Koden har blitt skrevet for å kjøre på maskinvare laget av tidligere NTNU studenter.

Systemet har blitt klargjort for å kunne bruke den åpne protese kommunikasjonsprotokollen PDCP. Implementasjonen av PDCP-protokollen for AVR hadde blitt gjort tidligere på NTNU. I denne oppgaven ble koden tilpasset for å kjøre på ATMELs AT90CAN128 mikrokontroller.

Det har blitt skrevet programdrivere for alle de forskjellige maskinvaremodulene. For å styre bevegelsen til håndleddet har det blitt skrevet og testet tre ulike regulatorer. En fartsregulator, en posisjonsregulator og en av/på-regulator som kjører protesen ved maksimum vinkelhastihet i valgt retning.

Da prosjektet ble overtatt var friksjon i mekanikken til protesen for stor til at den kunne brukes. Den ble tatt med til NTNUs mekaniske verksted hvor de fikk rettet en girakse som ga betydelig reduksjon av friksjon.

Alt av overlevert maskinvare har blitt nøye testet. Forslag til forbedring av maskinvaren har blitt dokumentert.

Videre arbeide vil bestå i å fullføre integrasjonen med PDCP-protokollen. Dette avhenger av at det blir publisert flere detaljer rundt bruken av protokollen. Før protesen skal brukes i eksperimenter vil man ha stor nytte av å miniatyrisere kretsen slik at den får plass inni protesen. Under miniatyriseringen av kretsen anbefales det å utbedre maskinvaren med de anbefalinger gjort i denne oppgaven.

Arbeidet på en analog kommunikasjonsmodul, som et alternativ til PDCP-modulen, har blitt påbegynt.

Bibliografi

[1] Arthur Zinck, *Design of a compact, reconfigurable, prosthetic wrist.* 2011.

[2] Øyvind Stavdahl, *Optimal wrist prosthesis kinematics: Three-dimensional rotation statistics and parameter estimation.* Department of Engineering Cybernetics, Norwegian University of Science and Technology. 2002. PhD Thesis.

[3] Andreas Kråkenes, *Styresystem for kybernetisk håndleddsprotese.* Master i teknisk kybernetikk 2011.

[4] Andreas Nordal, *Implementasjon og testing av en åpen bussprotokoll for armproteser.* Master i teknisk kybernetikk 2012.

[5] Andrzej Zamojski, *Design, Implementation and Testing of Low-level Layers of the PDCP for the AVR Platform.* Master i teknisk kybernetikk 2012.

[6] Inge Brattbakken *Embedded control system for cybernetic wrist prosthesis* Master of Science in Engineering Cybernetics 2010.

[7] Yves Losier *Prosthetic Device Communication Protocol for the AIF UNB Hand Project* UNB Hand Project Systems Integrator 2012.

[8] *8-bit AVR Microcontroller with CAN controller AT90CAN128. Datablad.* Atmel

[9] *Brushless DC-Servomotor, series 0620 B. Datablad* Faulhaber

Tillegg A

Innhold CD

A.1 Rapport

- Rapport pdf
- Figurer brukt i rapport

A.2 Programvare

- main.c
- motor.c, motor.h
- current.c, current.h
- position.c, position.h
- controller.c, controller.h
- analog.c, analog.h
- PDCP-protokoll implementasjon
- hal.c, hal.h
- can.c, can.h

A.3 Diverse

- NRWD video
- NRWD-RequirementsSpec_v0.4
- CAN-bibliotek

Tillegg B

Funksjonsspesifikasjon NRWD

NTNU

Norwegian University of Science and Technology

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

REPORT

Report number

Classification

ISBN

Address: **NTNU**
Department of Engineering Cybernetics
O.S. Bragstads plass 2D
NO-7491 TRONDHEIM, NORWAY

Switchboard: +47 73 59 40 00
Telephone: +47 73 59 43 76
Fax: +47 73 59 43 99

Report title

Functional Requirements Specification
for the
NTNU Revolute Wrist Device (NRWD)
v0.4

Date

Number of pages
11

Project supervisor
Øyvind Stavdahl

Author (s)
Øyvind Stavdahl

Project no.
TBC; Spør Tove

Sponsoring Organization
NTNU

Client's reference

Abstract

This document specifies the functional requirements for the initial version of the NTNU Revolute Wrist Device (NRWD).
The specification is intended to provide the necessary basis for developing a technical requirements specifications with respect to mechanical, electrical and algorithmic properties.

TBC

Keywords

TBC		

Contents

List of figures

1. Background

The functionally optimal 1-DOF wrist prosthesis kinematics was theoretically derived in [1]. Now one aims at implementing a physical device based on these principles.
The current document constitutes a functional specification for this device, which covers not only the kinematics but even mechanical, electrical, electronic and algorithmic aspects. Several of the numerical data are found in/derived from

2. Revision History

When	Who	What
2005.02.08	Ø. Stavdahl	Brief revisions for FPGA-based version. Several requirements and comments clarified and typos fixed. This relates to GEN-08, WJF-01, WJF-02, WJF-03, WSF-01-03, WSF-03, WCF-02, WCF-03, WCF-04, WCF-04-01, WCF-05, WCF-09, WCF-10, WPF-02, WPF-03, Circuit board geometry included.
Spring 2010	Inge Bratbakken	Revised specifica ons regarding communica on. This relates to WCF-02 and WCF-03. Also removed requirements no longer relevant. Those beeing: WCF-04, WCF-04-01, WCF-04-02, WCF-06, WCF-08 and WCF-09.
Autumn 2013	Geir Turtum	

3. Conventions

3.1 Abbreviations etc

The following conventions apply to this document:

NA	Not Applicable; irrelevant
HW	Hardware
SW	Software
TBC	To Be Completed; an aspect that has to be filled in.
TBD	To Be Defined; an aspect that is still not completely defined.

Use of the word *shall* denotes requirements that must be met by the system, while the word *should* denotes requirements that are desirable and must be met unless justification is provided for an alternative.

3.2 Document Structure

TBC.

4. System Description

4.1 Context and Purpose

The "system" referred to here is the entire wrist prosthesis, including both mechanical, electrical/electronic and software components. The primary purpose of the system is to rotate/orient the terminal device with respect to the forearm according to user input. Figure 1 gives an overview of the system and its context.

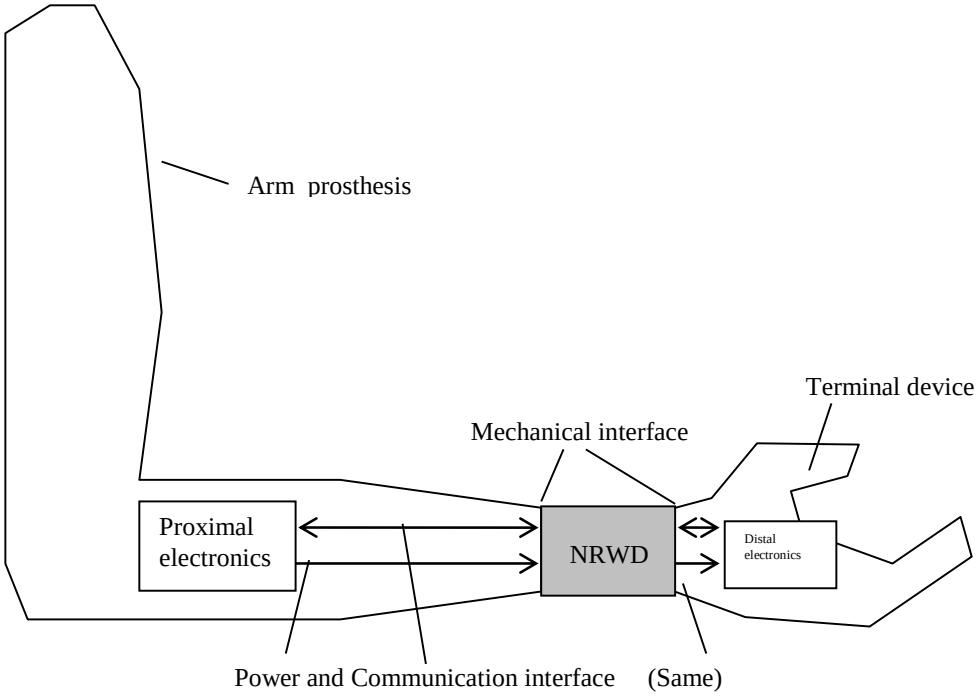


Figure 1 System context

The proximal electronics includes a battery pack or another electric power source, as well as digital and/or analog circuits with sensors for reading the user's motor intent (typically EMG electrodes, switches, "pressure pads" or the like) and communication of this intent to the joint controllers in the prosthesis.
While the figure suggests that the proximal electronics is situated in the forearm, this needs not be the case; in the case of a total arm replacement the prosthesis may comprise motorized shoulder, elbow, wrist and/or finger functions, and the arm electronics may correspondingly be distributed in the different parts of the arm. The "Proximal electronics" box in Figure 1 thus represents all the electronics proximal to the wrist, while the "Distal electronics" represents any or all electronic components distal to the wrist, such as a motor controller responsible for opening and closing of the hand.

4.2 Configurations

The system is applicable to several different equipment configurations, as indicated in Figure 2. These configurations include the following, listed in the order of relevance and only the first two being absolutely necessary to implement:

1. A completely analog mode where all intercomponent communication is based on dedicated analog lines. This mode complies with standard Otto Bock and comparable components.
2. A completely digital mode where all intercomponent communication is based on a data bus. This mode is for use with other novel systems that support the same communication protocol.
3. A hybrid mode where the proximal communication (i.e. between the wrist and proximal electronics) is digital while the distal communication is analog, intended for use with a Bock-like hand and novel proximal controllers.

4. A second hybrid mode where the proximal communication (i.e. between the wrist and proximal electronics) is analog while the distal communication is digital. This is for using a novel hand in systems based on analog electrodes and the like.

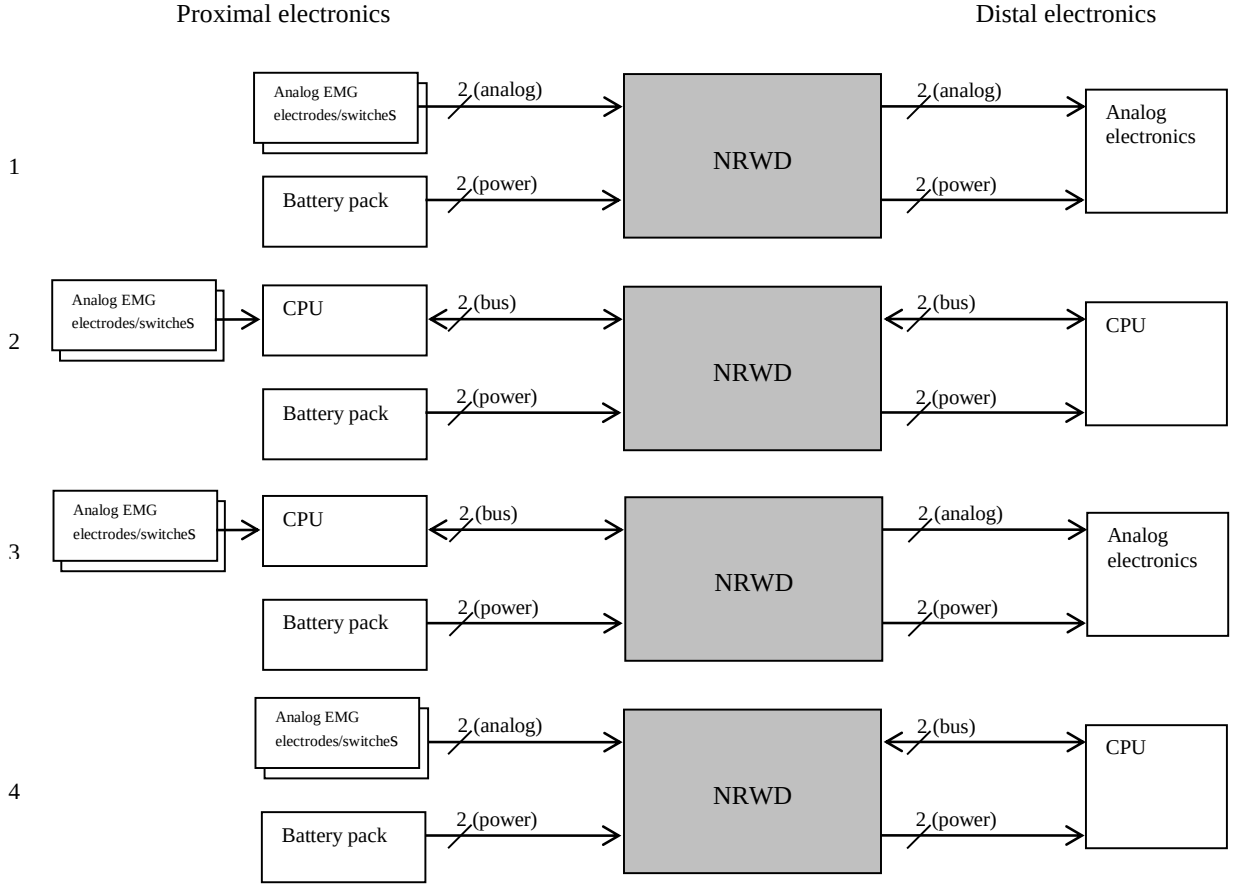


Figure 2 The four different equipment configurations

5. Functional Requirements

5.1 General requirements

The NRWD system shall implement the following functions and satisfy the following requirements.

Req. no.	Description	Comments
GEN-01	A joint which implements the optimal kinematics described in [1] – Wrist Joint Function, WJF.	
GEN-02	An electric motor which drives the movement of the joint via a gear train – Wrist Motor Function, WMF	
GEN-03	A motor control interface and regulation process to control the movements of the WJF – Wrist Servo Function, WSF.	
GEN-04	A communication function that interfaces with WSF and other connected systems – Wrist Communication Function, WCF.	"Other connected systems" typically include proximal and distal joint controllers, as well as units for system diagnosis and configuration.

GEN-05	A power adaptation function that enables the system to run from a variety of voltage levels – Wrist Power Function, WPF.	Compliance with commercial and research systems.
GEN-06	A mechanism for attachment of the wrist to the forearm socket – Proximal Attachment Function, PAF.	
GEN-06-01	The PAF **shall** be disconnectable and, when disconnected, allow entry to the space proximal to the wrist.	For maintenance and replacement of forearm electronics.
GEN-07	A mechanism for attachment of the wrist to the terminal device – Distal Attachment Function, DAF.	
GEN-08	An outer geometry that crudely resembles that of an adult wrist – Wrist Geometry Function, WGF.	No component shall extend beyond the envelope of a normal wrist, which has an elliptic cross-sectional area of approx. 5 cm x 4 cm. All parts and connectors must be kept small.
GEN-08-01	The longitudinal dimension (along the forearm) of the entire NRWD **should** be as short as possible, and **shall** not exceed 65 mm.	Allows longer residual limbs to use it, i.e. "larger market". 65 mm is Bock spec.
GEN-09	The system **should** consume a minimum of energy, and the current conspumtion **shall** be kept below 2 A at all times.	When motor is active, its output dictates a lower bound for the power consumption. This requirement just implies that unnecessary circuits should be turned off, and active use should be made of the controller's various sleep modes.
GEN-10	The system shall be modular with respect to both HW and SW.	
GEN-11	The weight of the entire NRWD **should** not exceed 100 g.	Industrial components: Otto Bock: 96 g, VASI: 100 g.

Figure 3 depicts the functions and their dependencies.

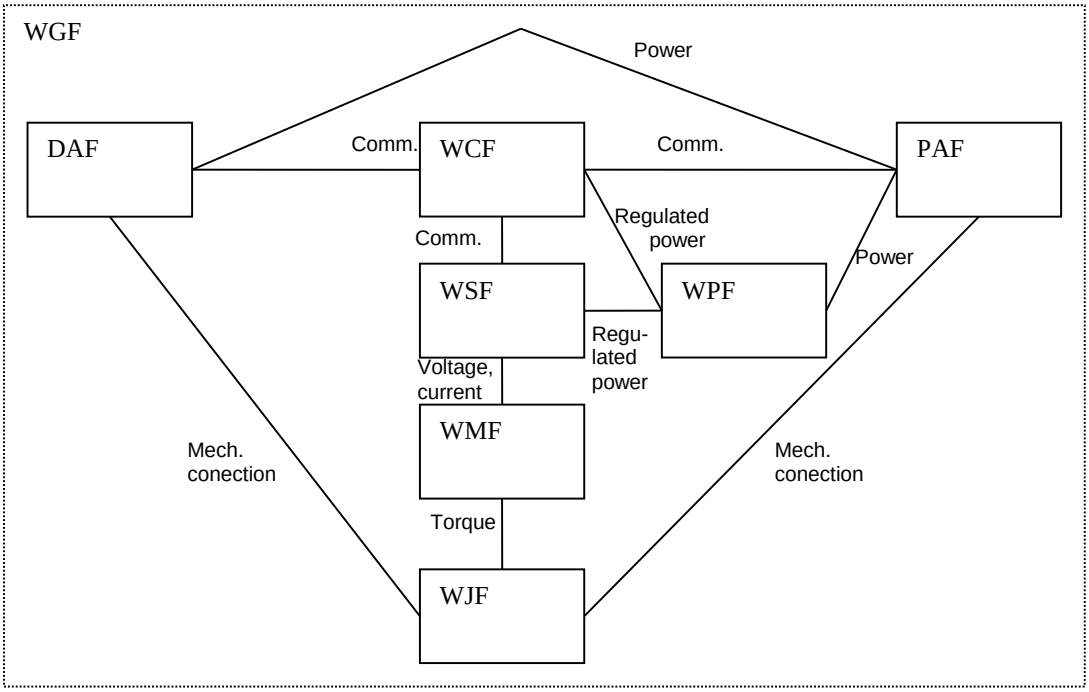


Figure 3 Functional block diagram

5.2 Wrist Joint Function, WJF

The following requirements are derived from GEN-01.

Req. no.	Description	Comments
WJF-01	The NRWD joint **shall** be a single, simple revolute joint which axis of rotation can be placed at an attitude with respect to the forearm and terminal device as spedcified in [1], Equations (7.4) and (7.5).	Derived from (specific version of) GEN-01. Note a typographical error in Eqn. (7.5); last element of vector should read "-0.23" instead of "0.23".
WJF-02	The joint axis **should** be manually adjustable to other attitudes than that specified in WJF-01.	In order to test other axis alignments. Note that adjustability wrt. Forearm AND hand requires TWO adjustable functions!
WJF-03	The joint **shall** enable an angular excursion of at least 180 degrees. The excursion **should** be unlimited.	180 deg is crudely that of a healthy limb.

5.3 Wrist Motor Function, WMF

The following requirements are derived from GEN-02.

Req. no.	Description	Comments
WMF-01	The wrist joint (output of the gear train) **shall** have a maximum angular velocity of at least 1.4 rad/s (81 deg/s). The maximum velocity **should** be as high as possible.	Otto Bock spec. No-load speed
WMF-02	The wrist joint **should** have a maximum	VASI spec.

	torque of at least 34,3 mNm.	Stall torque.
WMF-03	The motor **shall** have a maximum mechanical output power of at least TBC W.	
WMF-04	The motor shall be protected from overheating. The protection should be implemented by hardware.	

5.4 Wrist Servo Function, WSF

The following requirements are derived from GEN-03.

Req. no.	Description	Comments
WSF-01	The movements of the wrist joint **shall** be controllable according to the follwing modes: 1. On/Off-mode 2. Position mode. 3. Velocity mode	
WSF-01-01	In On/Off-mode the motor shall be at rest or run at maximum speed (open-loop) in one direction according to a given setpoint.	Requires only transistor bridge.
WSF-01-02	In position mode the joint angle **shall** be proportional to a given angular setpoint.	
WSF-01-02-01	The WSF **shall** include an absolute position sensor. This sensor **shall** provide no less than 10 bits resolution per revolution.	
WSF-01-03	In velocity mode the joint angular velocity **shall** be crudely proportional to a given velocity setpoint.	"Crudely proportional" implies that there is no explicit need for velocity feedback, the speed may be controlled in open-loop.
WSF-01-03-01	The WSF **shall** include a velocity sensor or estimator.	If brushless motor: use Hall elements for speed estimates.
WSF-02	The movements of the wrist joint **should** be controllable according to the follwing modes: 4. Torque mode 5. Impedance mode	
WSF-02-04	In torque mode the motor torque **shall** be proportional to a given torque setpoint.	Torque crudely proportional with motor current, so current can be used for feedback.
WSF-02-04-01	The WSF **should** include means for monitoring motor current.	Also follows from GEN-09.
WSF-02-05	In impedance mode the mechanical impedance of the joint **shall** be determined by a given impedance setpoint	Mech. Impedance = torque/velocity. May require strain gauge measurements etc. to "bypass" friction.
WSF-03	WSF **shall** provide an interface to WCF through which the modes (described in WFS-01 to WSF-02) can be selected and relevant parameters can be set, and through which WSF can report relevant state variables TBD.	Typically: Setpoints in, process values out. The precise content of this communication will be defined by a protocol sepcification (SCIP) TBD.

5.5 Wrist Communication Function, WCF

The following requirements are derived from GEN-04.

Req. no.	Description	Comments
WCF-01	The NRWD **shall** have a two-wire Proximal Communication Interface (PCI) and a two-wire Distal Communication Interface (DCI).	
WCF-02	The PCI shall be configurable so that	The 7.2V spec is approximate. For protocol, see

	it implements two (0V, 7,2V) analog input lines or a bidirec onal twowire CAN interface with the PDCP protocol(Losier,2009).	comment re. WSF-03.
WCF-02-01	The analog input lines **shall** be able to sample both lines at a rate of 1 kHz. The sampling rate **should** be as high as 2 kHz.	EMG signals have a bandwidth of approx. 500 Hz.
WCF-03	The DCI shall be configurable so that it implements two (0 V, 7.2 V) analog output lines or a bidirec onal two-wire CAN interface with the PDCP-protocol.	The 7.2V spec is approximate; it may be acceptable to reduce this to 5.0V.
WCF-04	The WCF **shall** be configurable to an all-Analog mode, with the PCI as an analog input interface and the DCI as an analog output interface.	
WCF-05	The WCF **shall** be configurable to an all-digital mode, with the PCI and the DCI acting as bidirectional CAN bus interfaces.	Only a single CAN interface is necessary, as both PCI and DCI can be internally connected to this single interface.
WCF-07	The WCF **should** be configurable to a hybrid mode in which the PCI acts as two analog input lines while DCI acts as a bidirectional CAN interface (cf. WCF-01)	Not strictly necessary, but improves interoperability with older systems.
WCF-10	The WCF **shall** include a serial interface for downloading software and for debugging/diagnostic purposes.	Same fashion as AVR Butterfly serial programming. This requires a bootloader. Might include RS-232 and/or other proper interfaces.
WCF-11	WCF **shall** implement an interface to WSF according to WSF-03.	

5.6 Wrist Power Function, WPF

The following requirements are derived from GEN-05.

Req. no.	**Description**	**Comments**
WPF-01	The WPF shall accept external power in the form of an unregulated two-wire DC supply.	Implies on-board voltage regulation.
WPF-02	The NRWD shall tolerate and run normally when powered with a voltage in the range (6 V, 12 V). The range of usable voltages should be as wide as (5 V, 18 V).	This corresponds to Otto Bock, Motion Control and other systems. Upper limit possibly to be relaxed (lowered).
WPF-03	The NRWD shall tolerate supply voltage in the range (0 V, 12 V) without exhibiting unpredictable behaviour and without getting damaged.	E.g. shutting down the controller before the voltage gets dangerously low, may otherwise damage Flash and EEPROM content etc. See comments to WPF-02.
WPF-04	The NRWD **should** automatically limit its motor current to a level that does not reduce the supply voltage below the interval given in WPF-02.	Low batteries => careful motor control to avoid power-down.

5.7 Proximal Attachment Function, PAF

The following requirements are derived from GEN-06 and more.

Req. no.	Description	Comments
PAF-01	The PAF **shall** comprise two parts, the proximal of which is adapted to be permanently attached to the forearm socket and the distal permanently attached to the wrist unit. The parts must "mate" to form a mechanically stable connection while also being detachable.	A commercially available "quick disconnect" unit may be used, but the entire mechanism must be kept as short as possible.
PAF-01-01	The proximal part of the PAF **shall** be hollow to allow access to the space within the socket proximally to the wrist.	Batteries and electrodes etc. is mounted here, so an opening must be present to allow maintenance and replacement of these units.
PAF-02	PAF disconnection **should** be possible with hand or a simple tool, e.g. a screwdriver.	
PAF-03	The PAF **shall** include a four-wire electric coupling, preferably mechanically integrated with the PAF itself.	GEN-04 and GEN-05. Preferably a "quick disconnect" type, optionally loose wires and a manually detachable coupling/plug.
PAF-03-01	The PAF electrical coupling **shall** include at least two power supply wires/contacts capable of transferring a constant current of 4 A per wire.	This is the current for the wrist AND the terminal device.
PAF-03-02	The PAF electrical coupling **should** be rotatable without twisting the wires.	Brush rings etc. This requirement and DAF-02-02 are mutually exclusive; both are not needed!

5.8 Distal Attachment Function, DAF

The following requirements are derived from GEN-07 and more.

Req. no.	Description	Comments
DAF-01	The DAF **should** comprise two parts, the proximal of which is permanently attached to the wrist and the distal permanently attached to the terminal device. The parts must "mate" to form a mechanically stable connection while also being detachable.	The distal part may be a function of the terminal device, e.g. a Bock hand. No "quick disconnect" required here!
DAF-02	The DAF **shall** include a four-wire electric coupling, preferably mechanically integrated with the DAF itself.	GEN-04 and GEN-05. Preferably a "quick disconnect" type, optionally loose wires and a manually detachable coupling/plug.
DAF-02-01	The DAF electrical coupling **shall** include at least two wires/contacts capable of transferring a constant current of 2 A per wire, and these wires **shall** be connected to the power supply wires from PAF.	This is the current that drives the terminal device. Check with Otto Bock (=1 A?); higher currents needed for more advanced/multifunction hands.
DAF-02-02	The DAF electrical coupling **should** be rotatable without twisting the wires.	Brush rings etc. This requirement and PAF-03-02 are mutually exclusive; both are not needed!

5.9 Wrist Geometry Function, WGF

See GEN-08.

Req. no.	Description	Comments

6. Bibliography

[1] Stavdahl, O., Optimal wrist prosthesis kinematics: Three-dimensional rotation statistics and parameter estimation. PhD Thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2002.

[2] Storheil, R., Optimal Wrist Prosthesis with One Degree-of-Freedom. Term Project Report, TTK4720, Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2004. (In Norwegian.)

[3] ToMPAW document (TBC)

Tillegg C

Diverse

Duty cycle	Vinkelhastighet
80	156 grader/sekund
65	94 grader/sekund
50	44 grader/sekund

Tabell C.1: Vinkelhastighet til NRWDen ved ulike duty cycles, 12V spenningsforsyning