NTNU
Det skapende universitet

Aktiv og adaptiv fremdrifsmekanisme for rullator

Sigurd Skarra

Master i teknisk kybernetikk (2 årig)
Innlevert: Juni 2012
Hovedveileder: Geir Mathisen, ITK

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

Masteroppgave: Aktiv og adaptiv fremdriftsmekanisme for rullator

De fleste teknologiske hjelpemidler som utvikles for velferdssektoren; det være seg tjenesteassistenter, løftemekanismer eller intelligente omgivelser, bidrar til å gjøre brukeren mindre aktiv og mer isolert i sitt eget hjem. I denne oppgaven skal det imidlertid utvikles teknologi og metoder som får brukeren opp av godstolen og ut av huset, nemlig en avansert rullator som krever en aktiv bruker. En rullator brukes i dag ikke bare som en hjelp til å komme fra A til Å. Det handler også om å ha noe å støtte seg til, å frakte med seg ting, å ha noe å sitte og hvile på og hjelp til å komme seg frem også der underlaget er ujevnt. I denne oppgaven vil jeg ta brukeropplevelsen ett skritt videre ved å utstyre rullatoren med et system basert på en adaptiv fremdriftsmekanisme, styring og regulering, slik at brukeren får ekstra drahjelp for eksempel i bakker eller når rullatoren er fullastet etter en tur på butikken.
Oppgaven bygger videre på en bacheloroppgave fra HIST og en prosjektoppgave på NTNU. Den eksisterende prototypen har aktive bakhjul, et sensorgrensesnitt på håndtaket som måler kreftene brukere trykker mot rullatoren og en kontrollerenhet.

Oppgaven går ut på følgende:

1. Foreta en litteraturstudie på regulering og styringsstrategier for adaptive og aktiv fremdriftsmekanismer av tilsvarende systemer.
2. Vurder forskjellige brukerscenarier og analyser hvordan styring av rullatoren på et overordnet nivå bør skje.
3. Ta utgangspunkt i funn fra punkt 1, 2 og/eller egenutviklede løsninger, og beskriv / simuler en styringsstrategi for vår rullator.
4. Basert på funn fra punkt 1, 2 og 3; lag en kravspesifikasjon som teknisk gjør rullatoren i stand til å benytte seg av den foreslåtte styringsstrategien.
5. Implementer og test foreslått styringsstrategi så langt tiden tillater det.

Forord

Jeg vil først og fremst takke veileder Terje Mugaas som har hjulpet meg både med høstprosjektet og denne masteroppgaven. Han har bidratt med konstruktive tilbakemeldinger og utfordrende spørsmål gjennom utviklingen av denne prototypen. Dette har vært til stor hjelp, da det har vært godt å være to når man diskuterer ulike løsninger på de utfordringene jeg har møtt på.

Jeg vil takke faglærer Geir Mathisen som alltid har hatt døren åpen for diskusjon og læring fra start til mål.

Jeg vil også takke mine medstudenter Erlend Frøhaug og Bjørn Skumsnes som har hjulpet meg med programmeringstips og diskusjon under masterperioden. I tillegg fortjener Per Inge Snildal og Terje Haugen på verkstedet til Instituttet for Teknisk Kybernetikk en takk for hjelpen med det mekaniske arbeidet.

Trondheim 14.juni 2012
Sigurd Skarra

Sammendrag

Dette prosjektet har omhandlet videreutvikling av en modernisert rullator for SINTEF, som har vært arbeidsgiveren. Rullatoren startet som en bachelorprosjekt ved Høgskolen i Sør-Trøndelag (HiST), avdeling for automasjon våren 2011, og høsten 2011 ble det gjennomført et prosjekt hvor det ble utviklet et nytt brukergrensesnitt. I denne masteroppgaven har man videreutviklet rullatoren da den hadde funksjoner som ikke fungerte optimalt. Rullatorløsningen SINTEF overtok fra HiST bestod av en rullator med elektromotorer, en instrumentkasse med et batteri og en mikrokontroller samt noe elektronikk til sensorer og motorer og to sensorer i håndtakene som var input for systemet. Systemet fungerte slik at man trykket på to sensorer som var lagt under gummihåndtakene for å få rullatoren til å kjøre fremover i lineær hastighet med kraftinputen. Tanken var at brukeren skulle bruke kroppsvekten for å generere fremdrift når man støttet seg på rullatoren. Dette viste seg ikke å fungere optimalt da man måtte bruke ganske stor kraft med tomlene for å trykke ned disse sensorene. Dette ble endret i høstprosjektet til håndtak som måler kraften brukeren dytter rullatoren fremover med. På den måten fikk man en mer naturlig brukeropplevelse og rullatoren ble enklere å betjene. Mer om løsningen kan leses om i prosjektrapporten *«Nye sensorer for en mer brukervennlig rullator» (Skarra, 2011)*[xvii].

Rullatoren skal være et intelligent hjelpemiddel som bidrar med aktiv fremdrift når brukeren trenger det, uten at brukeren selv trenger å gjøre noe for at denne hjelpen kobler inn. For å få til dette ble det gjennom en Use case-analyse avdekket hvilke funksjoner og krav vi måtte sette til rullatorsystemet for å få dette til å fungere. Dette resulterte i forslag til endringer både i det mekaniske og i reguleringsalgoritmen. I den gamle løsningen satt hjulene på faste aksler på motorene. Da vi ville at systemet kun skulle hjelpe til når brukeren trang det måtte flere ting løses. Hjulene ble byttet ut med hjul med frinav slik at rullatoren nå kan rulle fritt uten at motorene trenger å være koblet inn. Det måtte også endres i algoritmen for å få rullatoren til å virke slik vi ville. Vi ville at den skulle hjelpe til i bakker, når den var tungt lastet og når den møtte hindringer som krevde stor kraft for å bevege den fremover. Vi var altså ikke interessert i at den skulle bidra unødig og belaste batteriet når det var lett å dytte den fremover. Algoritmen ble derfor endret slik at man måtte over en viss innkoblingsgrense før motorene hjalp til. Kommer kraften over denne grensen kobler motorene inn og hastigheten er lineær ettersom kraften øker.

Det ble så sett på hvordan inngangssignalet ble behandlet i algoritmen da det viste seg at motorpådraget var noe ustabilt. Regulatoren var en ren P-regulator hvor inngangssignalet ble regnet ut som et gjennomsnitt av 5 målinger. Programmeringsfeil gjorde at deler av disse målingene ble lagret etter at rullatorsystemet ble slått av slik at rullatoren beveget seg litt når man skrudde den på igjen. Et nytt filter med glemmefaktor ble designet etter inspirasjon fra faget TTK4195 Systemidentifikasjon og adaptiv regulering, og programmert i mikrokontrolleren. Etter en del testing og justering førte dette frem til at regulatoren nå gir et stabilt og forutsigbart pådrag til motorene slik at den går med en jevn hastighet.

Håndtakene som ble laget i høstprosjektet, ble også modifisert, da de ikke hadde en stopperfunksjon som hindret dem i å bli demontert ved å dra holkene bakover. De er laget slik at holkene man holder i er montert på en ytre hylse som kan beveges frem og tilbake med fjærretur. Dette er gjort for å unngå en eventuell offset. Dette gjorde at man kunne ved et uhell dra holkene av rullatorrammen og ledningen til trykksensorene falt av. Holkene kunne også roteres så brukeropplevelsen virket litt lite robust når man håndterte rullatoren. En skrue ble skrudd gjennom den ytre hylsen og inn i et avlangt hull i den indre. Dette gjorde at man nå kan håndtere rullatoren som en vanlig rullator og trenger ikke være redd for at ledningene faller ut. Skruen kan enkelt skrus ut igjen med en standard stjernetrekker ved eventuelle endringer.

Alle disse endringene har gjort rullatoren nærmere et produkt som eventuelt kan settes i serieproduksjon. Det gjenstår fortsatt oppgaver med tanke på å få på plass en mekanisk brems, vektreduksjon, plassering av komponenter og å heve IP-graden til motorløsningen. Rullatoren har nå den styringsstrategien vi ønsker, og resultatene som er presentert i denne rapporten underbygger dette.

Arbeidet har vært utfordrende, og spesielt det å jobbe alene i en utviklingsperiode har vært krevende. Prosjektet har krevd at man har måttet ta avgjørelser under tidspress for å komme videre, og dette har vært både utfordrende og lærerikt. Dette er erfaringer som er nyttig å ha med seg inn i arbeidslivet.

Summary

This master thesis has dealt with development of a modernized walker by employment from SINTEF. The walker started as a bachelor project at Høgskolen I Sør-Trøndelag, department of automation, in the spring of 2011, and in the autumn of 2011 a project about improving the user interface was conducted. In this thesis some of the functions that didn't work optimal have been developed.

The walker that SINTEF received from HIST contained a walker with electric motors, an instrument box with a battery and a microcontroller along with some electronics for the sensors, and two sensors in the handles which was the input to the system. The system work in that way that you pushed on the two sensors which were put under the handlebar grip to get the walker to move. The walker then moved in a linear speed according to the force added. The idea behind this was that the user used his bodyweight to generate forward speed. This turned out not to work properly when you had to use quite a big impact with your thumbs to press down these sensors. This was changed in the fall project to handles that measures the force the user pushes the walker forward with. In this manner you got a more natural user experience and the walker was easier to operate. More about the solution can be read about in the project report *"Nye sensorer for en med brukervennlig rullator" (Skarra)*[xvii].

The walker should be an intelligent tool that helps with active progress when the user needs it, without the user having to do anything for this to engage. In order to achieve this, a Use case analysis was carried out to reveal the functions and requirements we had to set to the walker system. This resulted in proposals for changes in both the mechanical and the control algorithm. Since we wanted the system only to help when the user needed it, several things had to be solved. In the old solution the wheels was mounted on fixed axles on the motors. The wheels were replaced with wheels that could roll freely one way but locked the other way. Same as used for roller skis. Now the walker can roll freely without the need of the motors to be enabled. We also had to change the algorithm to get the walker to work as we wanted. We wanted it to help out in the hills, when it was heavily loaded and when it encountered obstacles that required great force to move it forward. We were thus not interested in that it would contribute unnecessary and drain the battery when it was easy to push it forward. The algorithm was therefore modified so that one had to exceed a threshold before the engines helped. If the force applied exceeds this threshold the motors engage and the speed is linear as the force increases.

It was then looked at how the input signal was processed in the algorithm as it turned out that the control signal was somewhat unstable. The controller was a pure P-controller where the input signal was calculated as an average of 5 measurements. Some programming error resulted in that parts of these measurements were stored after the walker system was turned off so that the walker was moving a little when it was turned on again. A new filter with weighting factor was designed with inspiration from the subject TTK4195 System identification and adaptive control, and programmed into the microcontroller. After some testing and tuning it led to the conclusion that the regulator now provides a stable and predictable control signal to the motors so that you go with a steady speed.

The handles, made in the fall project, was also modified when they did not have a stop function to prevent those to be removed by pulling the handlebar grip back. They are designed with the handlebar grip mounted on an outer sleeve which can be moved back and forth with spring return. This is done to avoid a possible offset. This meant that one could accidentally pull the handlebar grip of the walker frame and the wires to the force sensors would then fall apart. Grips may also be rotated so the user experience did not seem very robust when handling the walker. A screw was screwed through the outer sleeve and into a rectangular hole in the inner sleeve. This means that it's now possible to handle the walker as a regular walker and you don't need to be afraid for the cables to fall apart. The screw can easily be removed with a standard screwdriver for any maintenance.

All these changes have brought the walker closer to a product that may be put into series production. There are still tasks left like putting in place mechanical braking, weight reduction, replacement of components to make it possible to pack up and to raise the IP level for the engine solution.
The walker now has the control strategy we want, and the results presented in this report support this.

The work has been challenging, especially to be working alone in a development period has been demanding. The project has required that we have had to make decisions under time pressure to make progress and this has been both challenging and educational. These are experiences that are useful to bring into the workplace.

Innhold

1 Innledning

1.1 Bakgrunn for oppgaven

Oppgaven går ut på å videreutvikle en modernisert rullator som ble laget ved Høgskolen i Sør-Trøndelag våren 2011[xx]. Prosjektet startet som et samarbeid mellom rullatorprodusenten Torpo og SINTEF og ble gitt som bacheloroppgave til ingeniørutdanningen for automasjon ved HiST.
Prototypen er en modernisert rullator med elektromotorer for fremdrift, et koblingsskap med batteri og en mikrokontroller, samt sensorer som detekterer kraften brukeren dytter rullatoren med. Sensorene som detekterer kraft ble laget i faget TTK4551 Teknisk Kybernetikk Fordypningsprosjekt, høsten 2011[xvii].

I helsevesenet vil det i fremtiden være behov for robotiserte hjelpemidler som kan være et supplement i omsorgsarbeid. Denne oppgaven vil gi et bidrag til feltet robotiserte hjelpemidler i form av en rullator som hjelper eldre i det daglige og som har smarte og intelligente løsninger som gjør håndteringen lettere.

Bakgrunn for oppgaven er å utvikle en intelligent rullator som løser et problem som ikke tidligere har blitt løst i Norge.

1.2 Motivasjon

Norge har en økende aldrende befolkning. Behovet for hjelpemidler og smarte helserelaterte roboter vil være økende i fremtiden. I følge Folkehelseinstituttet er omkring 15 % av befolkningen 65 år eller eldre, og denne andelen vil øke frem til år 2020[i]. I år 2100 er denne andelen estimert til å være det dobbelte. Dette er et forsiktig anslag, og SINTEF har anslått at Norge vil mangle 5000 sykehjemsplasser om 20 år, hvis man skal ha samme sykehusbehandling som man har i dag[ii].

Med dette som bakgrunn er det klart at forskning og utvikling av smarte hjelpemidler bør i større grad gjennomføres før en slik bølge slår inn. Et slikt produkt vil bidra både til en enklere hverdag for eldre, slik at de kan være selvstendige lengre, og det kan hjelpe dem som jobber med eldre med f.eks. avlastning ved tunge løft.

Dette prosjektet retter seg inn mot hjelpemidler som skal hjelpe eldre med å få en enklere hverdag og økt selvstendighet.

Hvis denne rullatoren kan være med å bidra til at eldre holder seg lengre aktive og selvstendige vil det kunne gi besparinger i et samfunnsøkonomisk perspektiv.

1.3 Begrensninger

Fokuset i denne oppgaven har vært å forbedre prototypen. SINTEF har satt krav om at kompleksiteten på løsningene skal holdes relativt lavt slik at komponentkostnadene ikke skal bli for dyre. Analysen av systemet i denne rapporten har gitt en ramme for hvilke oppgaver som må løses for å komme nærmere et fullverdig produkt. Gjennom dette prosjektet har det blitt fokusert på frigang for hjulene og reguleringsalgoritmen.

1.4 Bidrag

Gjennom litteraturstudiet er det gjennomgått en del av de mest aktuelle og spennende systemene som har en aktiv og adaptiv reguleringsstrategi. Oppgavene som er løst i dette prosjektet er en videreutvikling av høstprosjektet. I det prosjektet fikk rullatoren et adaptivt brukergrensesnitt med håndtak som måler kreftene brukeren påtrykker rullatoren med. For å komplementere rullatoren med de kravene som legges til grunn i denne oppgaven, har den fått hjul som triller uavhengig av motorakselen og som gjør at den kan brukes i strømløs tilstand. Motorene trenger ikke å være koblet inn for å kjøre rullatoren. Dette var en oppgave som ikke var løst når SINTEF tok over rullatoren fra prosjektgruppen på HiST. Disse hjulene gjør at regulatoralgoritmen nå fungerer på en mer intelligent måte. Funksjonene til rullatoren er nå slik den var ment, og rullatoren gir nå en moderat støtte ved å hjelpe til i oppoverbakker og ved tung last. Det er testet ut flere algoritmer for å finne ut hvilke som gir mest stabil fart og som gir mest intelligent virkemåte.

1.5 Målgruppe

Rapporten henvender seg til personer med interesse for helseroboter og prototypeutvikling av slike. Språket er prøvd holdt på et forståelig nivå, men en viss bakgrunn fra elektronikk og reguleringsteknikk hos leseren er å foretrekke.

1.6 Disposisjon av oppgaven

Kapittel 1 er innledning til oppgaven og bakgrunn for arbeidet.

Kapittel 2 er en litteraturstudie hvor det har blitt søkt etter lignende systemer og løsninger som kan være relevant for oppgaven. Dette kapittelet svarer på oppgave 1 i oppgaveteksten.

Kapittel 3 er et teorikapittel hvor rullatorsystemet er blitt analysert ved bruk av Use Case-analyse. Her er det vist alle scenarier rullatoren kan komme ut for, hvilke som sammenfaller og hvilke krav disse setter til rullatorens funksjoner og til reguleringsalgoritmen. Dette kapittelet svarer på oppgave 2 i oppgaveteksten. Det er også beskrevet hva slags teknisk utstyr som har blitt brukt i arbeidet med rullatorsystemet.

Kapittel 4 tar for seg forslag til endringer og løsninger for rullatoren på bakgrunn av litteraturstudie og Use case-analysen. De enkelte forslagene er delt opp i hvert sitt underkapittel der det er begrunnelser for valg og løsninger. Dette kapittelet svarer på oppgave 3 i oppgaveteksten. Kapittel 4.1 svarer på oppgave 4 og gir den tekniske kravspesifikasjonen for at rullatoren kan benytte seg av den foreslåtte reguleringsstrategien.

Kapittel 5 tar for seg implementeringen av hjulene med frinav og endringene i reguleringsalgoritmen. Dette kapittelet svarer på oppgave 5 i oppgaveteksten.

Kapittel 6 tar for seg uttesting og resultater.

Kapittel 7 er diskusjon av resultater og prosjektet

Kapittel 8 er konklusjon av prosjektet.

Kapittel 9 tar for seg forslag til videre arbeid.

1.6 Beskrivelse av rullatoren

Rullatoren er en Torpo Troja rullator . Det meste av hardware og software ble bygget som et bachelorprosjekt ved Høgskolen i Sør-Trøndelag våren 2011[xx]. Da den ble tatt over av SINTEF, var den relativt lik som nå, men hadde håndtak hvor man måtte presse ned med tomlene oppå håndtakene for å gi input til systemet. Hjulene var også montert på faste aksler.

Rullatoren har en motor og et posisjonskort på hvert av de to bakhjulene, en trykksensor i hvert av de to håndtakene, en instrumentkasse som inneholder det meste av elektronikken, en 12 VDC ladekontakt og en Av/På-bryter, som vist i Figur 1.

Rullatoren fungerer slik at den gir moderat støtte ved at den har aktiv fremdrift i oppoverbakker eller når den er lastet tungt ved at to motorer kobler inn og hjelper rullatoren fremover. Hvis rullatoren kjører i fra brukeren, løser de to håndtakene som er montert på en hylse ut ved hjelp av to fjærer, og rullatoren mister sin input. Da kobler motorene ut og rullatoren stopper. På denne måten har den et adaptivt brukergrensesnitt og en aktiv fremdrift.

Figur 1 Komponentene på rullatoren

1.7 Forkortelser og begreper

Binært – Signal som enten er 1 eller 0,

C-kode – Et av verdens mest brukte programmeringsspråk.

Debugger – Dataprogram som lar deg kjøre programmet, linje for linje, og undersøke verdier av variablene, eller se på verdiene som sendes inn til funksjoner i sanntid.

Forrigling – Koblings – og programmeringsmetode for å forhindre at uønskede tilstander kan være koblet inn samtidig.

FSR – Force Sensitivity Resistance

Mikrokontroller – Komponent som inneholder hele sentralenheten (CPU) i et datasystem. Enkelt sagt er en mikrokontroller en liten, nesten komplett datamaskin med CPU, hukommelse, IO-enheter og interne og eksterne databusser.

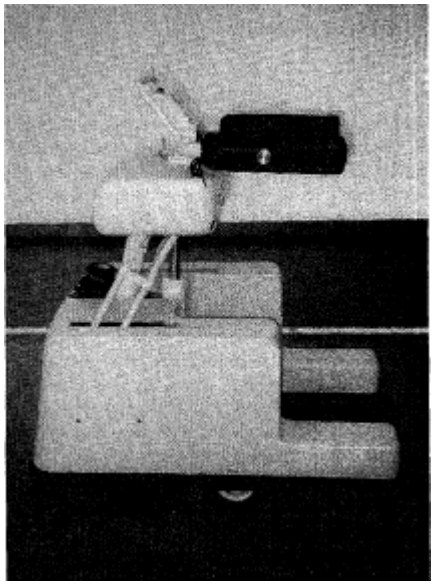
UART – Universal Asynchronous Receiver/Transmitter, kommunikasjonsprotokoll.

2 Litteraturstudie

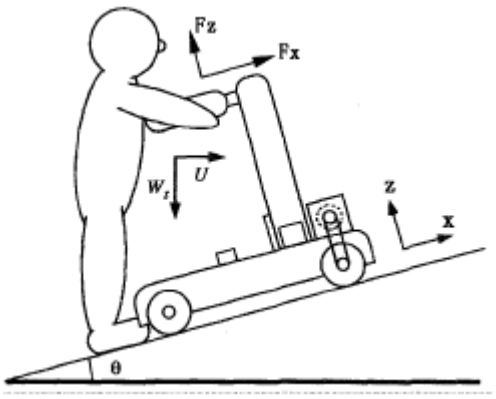
Her er en oversikt over eksisterende produkter og prototyper innenfor feltet «smarte helseroboter» som har blitt brukt som inspirasjon til den adaptive og aktive rullatoren. Studiet inneholder aktive hjelpemidler i form av rullatorer som har blitt utviklet ved flere forskjellige universiteter i verden. Det er også undersøkt rundt konsepter med virkemåter som vi vil at rullatoren skal ha.

2.1 Walking Support System

Hitachi (1998)[iii] har utviklet et rullatorsystem som gir aktiv støtte i bakker. Systemets design er bygd opp slik at man kan hvile kroppen på to armlener, se Figur 2. Designet minner om det som i norsk helsevesen kalles en «prekestol». En «prekestol» er en høy gåstol som tar av for kroppsvekten ved at man kan hvile armene over en ramme som ser ut som en halvsirkel. Rullatoren har aktiv hjelp ved gange i oppoverbakke, støtte i oppreist posisjon ved stans, og hjelp til å reise seg opp og ned når man f.eks. skal opp og ned av sengen eller av og på et toalett. Som man kan se på figur 1 er hele den øvre delen montert med løftesylindre som hjelper brukeren opp og ned. Til fremdrift har de som inngangssignal brukt kraftsensorer med to frihetsgrader. De måler altså kraft i horisontal -og vertikal retning. Man dytter eller drar helt enkelt i håndtakene på rullatoren og så går den fremover eller bakover. For å svinge er det differansen i kraften som man påfører håndtakene som måles. Systemet er designet for kun å gi moderat støtte, så på flatmark triller rullatoren som en vanlig rullator. Det står ikke beskrevet hvordan denne mekanismen fungerer, men det er naturlig å tro at de bruker hjul med frinav eller et clutchsystem i tillegg til en vinkelsensor som kobler ut fremdriften når den ikke er i en helning. Da rullatoren er bygget slik at man legger begge underarmene oppå rullatoren hevder de at brukerens vektkomponent nedover vil bidra signifikant til feil i kraftmålingene. De har derfor kompensert for vektorkomponenten av tyngdekraften som vil komme i tillegg når man går i oppoverbakke, se Figur 3. Vinkelen på bakken måles ved hjelp av vinkelsensorer og brukes i regulatorloven. De hevder at dette hadde en effektiv virkning på kontroll av rullatoren og den føltes naturlig helt opp i 10 graders helning.



Figur 2 Walking Support System fra Hitachi (1998)[iii]

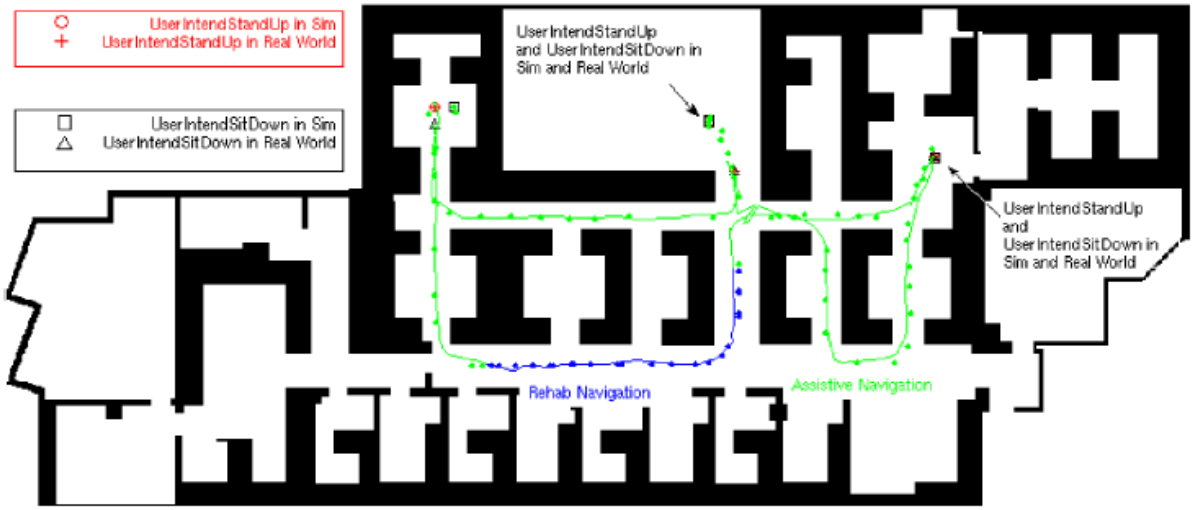


Figur 3 Vektorkomponentene fra brukerens vekt, Hitachi (1998)[iii]

2.2 Mobility Agent for Motor Function Rehabilitation and Ambulation Support

J. V. Mir ´o (2007), ved The Faculty of Engineering and IT, University of Technology Sydney (UTS), i NSW Australia [iv] har bygget en rullator som har aktiv fremdrift, låsing av hjulene når brukeren skal sette seg ned, og den har en form for ruteplanlegging, se Figur 4. Til fremdrift har de valgt strekklapper på den vertikale aksen til rullatorrammen som input. Ved å gi kraft fremover eller bakover måles momentet i rammen og rullatoren kjører i ønsket retning. I tillegg har de installert kraftsensorer i håndtakene som måler kraft i vertikalretning for å detektere om brukeren reiser seg eller setter seg ned. En høy kraft i disse sensorene impliserer at brukeren skal reise seg og da kjører rullatoren et lite stykke fremover for å hjelpe brukeren opp. Til deteksjon om brukeren står oppreist ved håndtakene, brukes det to IR sensorer med kalibrert område på 20 til 150 cm. Dette er også en sikring mot at rullatoren beveger seg hvis noen kommer borti et av håndtakene. Ruteplanleggeren fungerer ved at rullatoren må tas med til et forhåndsinnstilt punkt for så å kjøre en forhåndsprogrammert rute til et annet punkt. Dette er laget for å aktivisere eldre i en fast rute. Langs ruten bruker rullatorkontrolleren en «Partially Observable Markov Decision Process» [v], eller POMDP som overordnet beslutningsstrategi sammen med en lasersensor som skal forhindre kollisjoner. Laseren har et søkeområde på 240° og er montert foran på rullatoren, nede ved gulvet. POMDP-algoritmen tar kun hensyn til tilstanden rullatoren er i, og ikke bakover i tid. Dette gjør at man tar minst mulig hensyn til brukerens inngripen for å hindre fall og kollisjoner og prøver å kjøre mest mulig langs ruten. Kjører rullatoren langs ruten og den plutselig får en endring ved at brukeren kjører vekk fra ruten, retter den seg inn igjen i stedet for å følge brukerens handlinger. Hvordan ruteplanleggeren er implementert sier ikke artikkelen noe om. Forskerne gjorde forsøk av systemet med en rute i et kontorlandskap og testet det opp mot en fullt observerbar Markov Decision Process og fant at POMDP-algoritmen ga gode resultater, tett opp mot den optimale ruten, se resultat i Figur 5 på neste side.

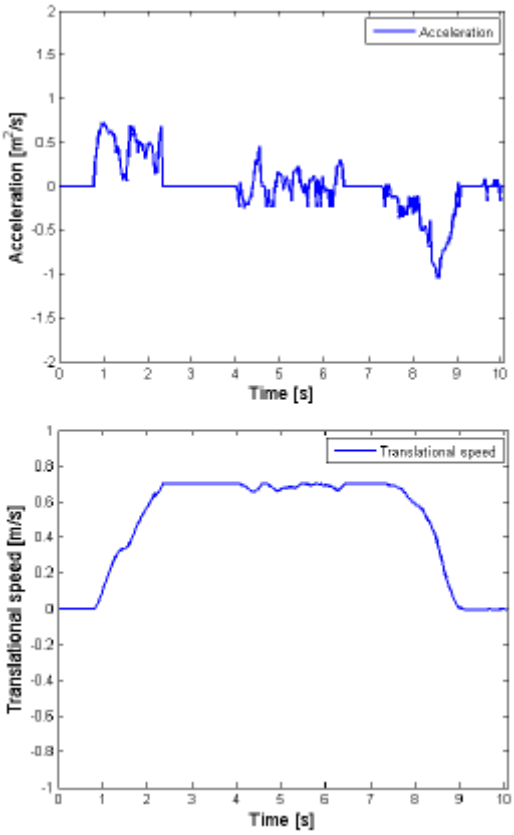
Figur 4 Rullator laget ved UTS, Australia (2007)[iv]



Figur 5 POMDP rullatortest. Simulert rute som prikker, testet rute i heltrukne linjer[iv].

2.3 Walkmate

Fei Shi (2010), ved Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China har utviklet et rullatorsystem de har kalt *Walkmate*[vi], se Figur 7. Dette er en rullator som også har aktiv fremdrift med adaptiv input fra brukeren. De har også bygget håndtak som benytter seg av FSR (Force Sensing Resistors) sensorer, likt vårt system, men kan måle kraft i begge retninger (fremover og bakover). Rullatorsystemet er også utstyrt med en mulighet for å senke rullatoren slik at den kan brukes som en stol til hvile o.l. Ved bruk av rullatoren har man to moduser den kan stilles inn på; en hvor kraft genererer hastighet, slik som vårt system er bygget opp, og en hvor akselerasjon er utgangssignalet til motorene. Kraften man trang for å få rullatoren opp i den fastsatte makshastigheten var mindre enn den man trang ved hastighetsmodusen, og kraften man trang for å opprettholde denne hastigheten var også lavere. Dette er logisk og samsvarer med virkeligheten når man dytter noe med hjul med tanke på friksjon. Dytter man noe tungt er det lettere å opprettholde farten når den først har kommet i bevegelse. Ved å bruke akselerasjon viste det seg at de fikk en mye mer stabil fart ved vanlig gange, enn ved bruk av hastighet, se Figur 6 for resultat.
Akselerasjonsmodusen ga stabil fart og bedre manøvrering, mens hastighetsmodusen hadde raskere respons.

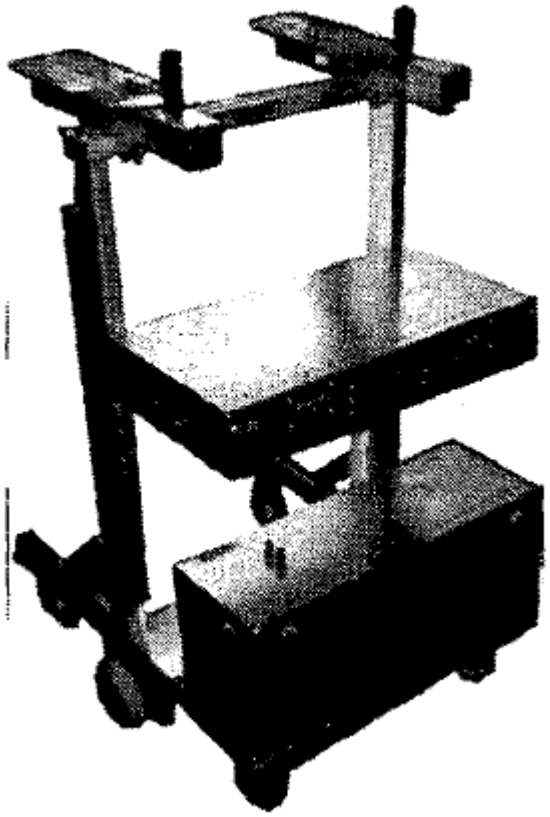


Figur 6 Øverst: Kraft målt ved vanlig gange. Nederst: Hastighet på rullatoren ved akselerasjonsmodus. Shanhai Jiao Tong University[vi]

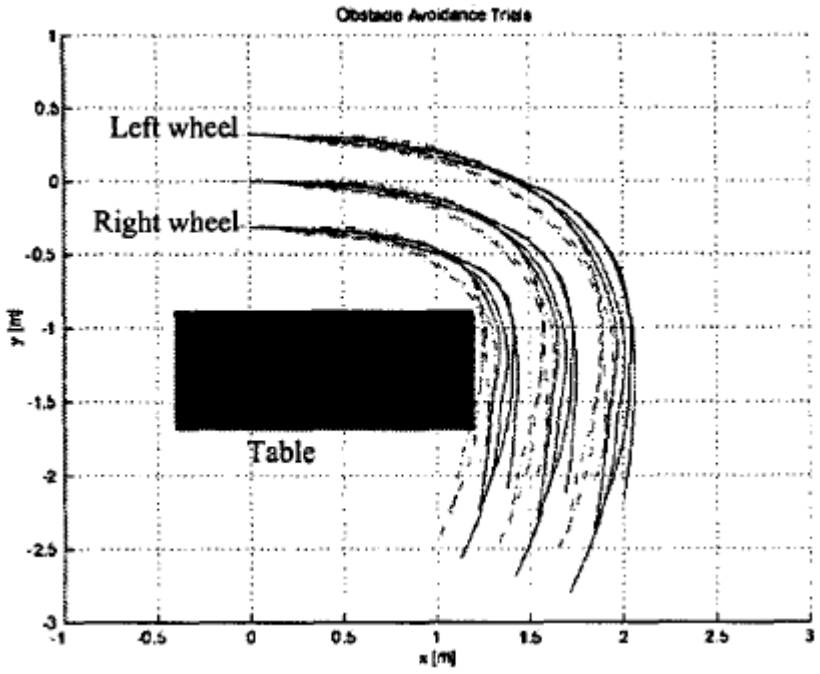
Figur 7 Walkmate[vi]**, China (2010)**

2.4 A Mobility Aid for the Support to Walking and Object Transportation of People with Motor Impairments

Sabati A.M. m. fl. (2002) ved Scuola Superiore Sant 'Anna, Pisa, Italy[vii] har utviklet en rullator som også har aktiv fremdrift, se Figur 8. Rullatoren har to moduser, en hvor man styrer farten med to joysticker som input og en modus der brukeren styrer rullatoren foran seg ved hjelp av to ultralydsensorer, kalt *follow-me-mode*. Joystickene er ikke av tradisjonelle typer, men håndtak hvor de har montert på strekklapper som måler kraften brukeren bøyer de med fremover eller bakover. I *follow-me-mode* kjører rullatoren fremover når brukeren går mot den innenfor et område og stopper når brukeren stopper. For å detektere at det er brukeren som står ved rullatoren må man ha på seg et belte som ved hjelp av infrarøde sensorer detekterer brukeren. I tillegg har de laget et system for å unngå kollisjoner der de bruker 7 ultralydsensorer som er montert foran og på sidene av rullatoren, 20 cm opp fra bakken. For å tolke hindringer har de brukt et kalmanfilter for å unngå bord, stoler o.l. Testene deres viser at rullatoren fint kunne kjøre rundt et bord, se Figur 9 for plotting av resultater.



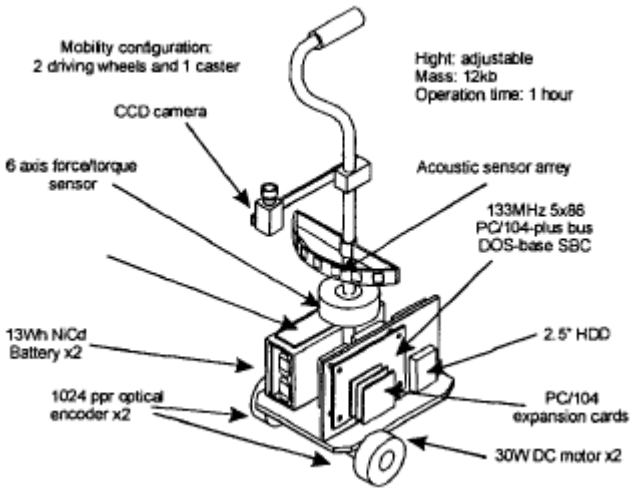
Figur 8 Rullator fra Scuola Superiore Sant'Anna, Italy (2002)[vii]



Figur 9 Resultat av kjøring rundt hinder med kalmanfiltrering[vii]

2.5 The Smart Cane

Dubowsky Steven (2000) ved Massachusetts Institute of Technology (MIT) [viii] har laget et system som både hjelper til fysisk og som overvåker helsen til brukeren. Som prototype har de laget en motorisert gå-stokk, se Figur 10. Den har en sensor som måler kraft og moment i 6 akser for fremdrift og en akustisk sensor som skal hindre kollisjoner. For å overvåke hvor brukeren er, har de montert et kamera som leser av sjekkpunkter i taket. Denne informasjonen sendes så trådløst til en pc hvor man kan se hvor brukeren er og evt. hvor langt brukeren har gått den dagen. I tillegg overføres brukerens helsetilstand trådløst, som puls o.l. i et eget system. Smart Cane-systemet har fire moduser som man kan velge mellom. Det er vanlig kjøring der brukeren styrer fart og retning selv, kjøring hvor både rute og fart er forhåndsinnstilt, kjøring der ruten er innstilt, men man kan styre farten selv og en modus der man følger en rute, men har mulighet for å avbryte ruten underveis.



Figur 10 The Smart Cane Prototype PAMM System, MIT (2000)[viii]

2.6 Elektriske sykler

Rullatoren skal ha en adaptiv reguleringsstrategi. Dette er løsninger man finner igjen i elektriske sykler , og det er derfor undersøkt hvordan dette er løst.
Det finnes i dag to typer elektriske sykler. En med handgass, som på en moped, og en som kun hjelper til i motbakker. Virkemåten med at de kun hjelper til i motbakke, eller ved mye last, er noe av det samme vi ønsker å oppnå med rullatoren. Løsningen hvor sykkelen kun hjelper til i motbakke, har en momentsensor i pedalnavet, se Figur 11. Denne sensoren er så tilkoblet et kontrollkort som styrer pådraget til motoren. Det kan så stilles inn på hvilket moment man vil at motoren skal legge inn på[ix]. En slik innkoblingsgrense er aktuelt å bruke på rullatoren.

Figur 11 Torque Sensor i pedalnavet[ix]

2.7 Expresso posttralle

Expresso er en tysk leverandør av elektriske trallesystemer[x]. De leverer modeller i flere varianter, men alle har elektriske motorer med individuell drift og håndtak med FSR-sensorer som styrer hvert sitt hjul, se Figur 12. Systemet kan brukes til mange forskjellige oppgaver og er ment å avlaste rygg og muskulatur ved håndtering av store ting samt frakte tunge ting i trapper. Trallen tilpasser seg brukerens hastighet og har to former for bremsing. Motoren regenererer energi når den kjører i nedoverbakke som en form for brems og den stopper helt når brukeren slipper håndtakene.

Figur 12 Expresso posttralle[x]

2.8 Sammendrag av litteraturstudie

Et fellestrekk for alle de beskrevne prototypene er at de alle er bygget som «proof-of-consept» og fremstår som dyre og komplekse løsninger. De er som oftest ment for innendørsbruk og for brukere som i mer eller mindre grad har bevegelseshemning. Vårt system ser mer i retning av et produkt som kan serieproduseres og som ikke skal avvike for mye fra et standardprodukt. Da må pris, design og ikke minst kompleksitet vurderes nøye. Vårt system skal også kunne brukes utendørs så krav til robusthet skal også tas med. Men tankegangen fra de forskjellige systemene er verdt å ta med seg videre.
Erfaringer som kan tas med videre, og som kan ha relevans for vårt system, er blant annet styrestrategien fra *Walkmate*[vi] der de bruker akselerasjon som utgangssignal og fikk med det en veldig stabil hastighet. Fra systemet til Hitachi[iii] er det å gi moderat støtte noe som skal tas med videre. Rullatoren gir da kun aktiv fremdrift i motbakker og har frigang på flatmark.
Fra systemet som er utviklet ved universitetet i Sydney[iv] kan det å måle kraft vertikalt for å detektere at brukeren skal reise seg for så å låse hjulene, være noe å vurdere. Dette kan løses med strekklapper og en målebro for å detektere at brukeren skal reise seg eller sette seg.
Litteraturstudiet har gitt flere interessante ideer til hvordan man kan løse forskjellige utfordringer. Disse vil bli vurdert senere i rapporten.

3 Teori

Den første delen av kapittelet tar for seg en strukturert systemanalyse av rullatorens brukerscenarier. Den siste tar for seg teknisk informasjon om det utstyret som er brukt til å programmere mikrokontrolleren med og om inngangskortet som er brukt til å logge resultatene med.

Kapittel 3.1 går gjennom hvordan en strukturert analyse, ved hjelp av verktøyet UML (Unified Modeling Language) - Use case, gjennomføres. I kapittel 3.2 vil de enkelte brukerscenariene bli beskrevet og analysert ved hjelp av Use Case. Rullatorsystemet er analysert for å belyse de brukerscenariene den kan komme opp i, og for å avdekke forbedringsområder. Kapittel 3.3 lager et sammendrag av hva analysen avdekker og hvilke utfordringer som må løses. Kapittel 3.4 tar for seg det tekniske utstyret som har blitt brukt under prosjektperioden.

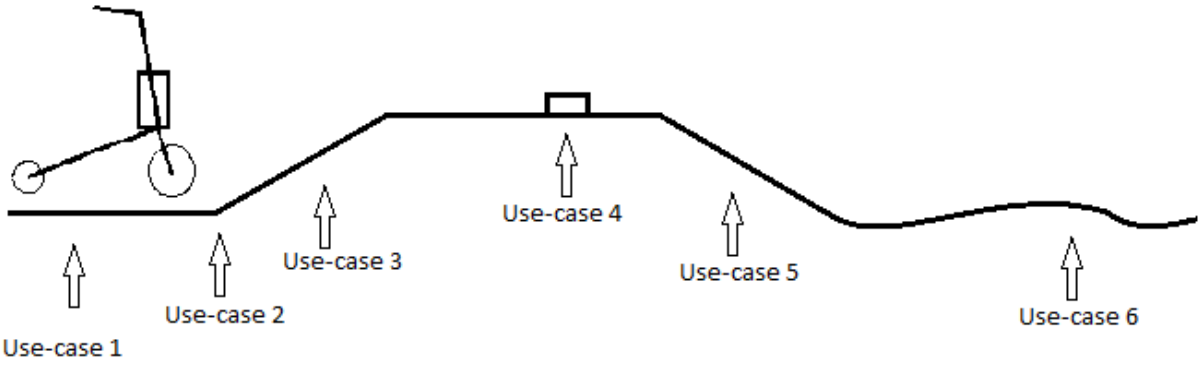
3.1 UML 2.0 (Unified Modeling Language) – Use case, en grafisk beskrivelse

UML 2.0 er et standardisert generelt modelleringsspråk innenfor feltet objekt-orientert softwareutvikling som ble utviklet av Object Management Group i 1997[xi]. UML 2.0 kan settes opp i diagrammer og deles inn i to innfallsvinkler, statisk og dynamisk fremstilling. Den statiske viser objektene, operasjonene og sammenhengene i systemet. Det dynamiske diagrammet viser korrelasjoner og endringer i de interne tilstandene til objektene. Det finnes mange forskjellige UML-verktøy og her er Use case tatt i bruk for å vise systemets funksjoner og begrensninger. Use case er et dynamisk diagram og er en måte å bygge opp systemet grafisk på. Ved å tegne opp alle brukerscenariene systemet kan komme opp i, finner man fort ut hvilke begrensninger og funksjoner systemet har. Use case-analyse kan da brukes til å finne ut hvilke oppgaver det er mest hensiktsmessig å løse for å få et fungerende system. I tillegg kan man finne frem til handlinger som fører frem til samme scenario, noe som kan forenkle programmeringsbiten. Use case er et hjelpemiddel for programmering, men fremgangsmåten kan også brukes til analyse av fysiske systemer. Som et resultat av analysen får man ut hvilke krav man må sette til systemet. Etter en ferdig analyse kan det vise seg at systemet må utvides og nye funksjoner må installeres. Et Use case bygges opp for å beskrive hvilke aksjoner et system kan utføre ved at en ekstern bruker, kalt aktør, påvirker systemet. Når alle brukerscenarier er beskrevet, sitter man igjen med alle krav som systemet krever og alle funksjoner som systemet tillater.
I denne analysen er Use case brukt som et grafisk verktøy som beskriver de forskjellige scenariene. De er i tillegg beskrevet med tekst for å beskrive grundig hvilke utfordringer hvert scenario står ovenfor. Diagrammene leses på den måten at det beskriver aktørens aksjon, systemets respons og funksjon eller feilhåndtering. Teorien bak analysen er hentet fra boken *Doing Hard Time (Douglass, 2000) – Developing Real-Time Systems with UML, Objects, Frameworks, and Patterns.*

3.2 Brukerscenarier

I Figur 13 er alle scenariene rullatoren kan komme opp i, presentert. Det er ment til å avdekke hvilke utfordringer de forskjellige scenariene lager og hvilke som sammenfaller og kan gjøre ting enklere. Med dette menes at programmet kan lages så intelligent at det tolker ulike sitasjoner likt og gir optimal respons på dette.

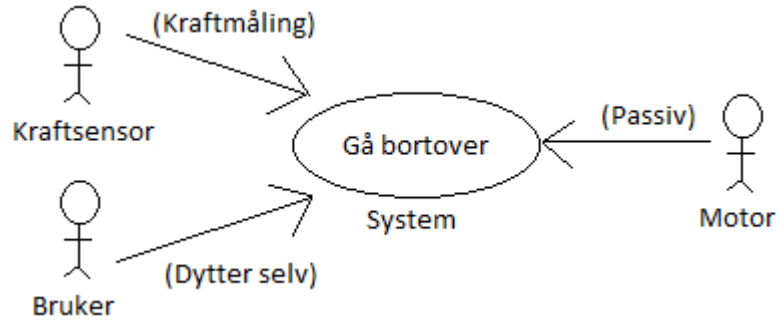
Rullatoren har i prinsipp to hovedmoduser. En der den fungerer som en vanlig rullator og en der rullatoren gir aktiv fremdrift.
De forskjellige Use-casene er laget ut i fra både der man har statiske situasjoner, hvor input og output er stabile, og der hvor man har endringer. For å gi en grafisk fremstilling av hvilke senarioer rullatoren kan komme ut for er det laget en oversikt som deler opp disse sett fra et praktisk perspektiv, se Figur 13. I tillegg er det laget use-caser rullatoren kan komme opp i hvor den er fullastet, brukeren skal reise eller sette seg og hvor rullatoren er strømløs.



Figur 13 Oversikt over brukersenarioer

3.2.1 Use-case 1 Normal situasjon

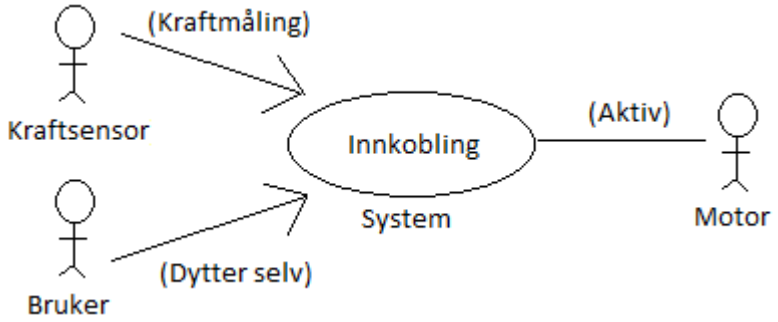
En normal situasjon defineres til å være når rullatoren brukes på flatmark, uten last. Det impliserer at hjulene skal ha mulighet for å rulle fritt uten at motorene er koblet inn og at bremsene kan brukes som på en vanlig rullator. Da vil man ikke trenge aktiv fremdrift. For å se hvordan aktørene forholder seg til systemet, se Figur 14.



Figur 14 Gå bortover-senarioet

3.2.2 Use-case 2 Innkobling

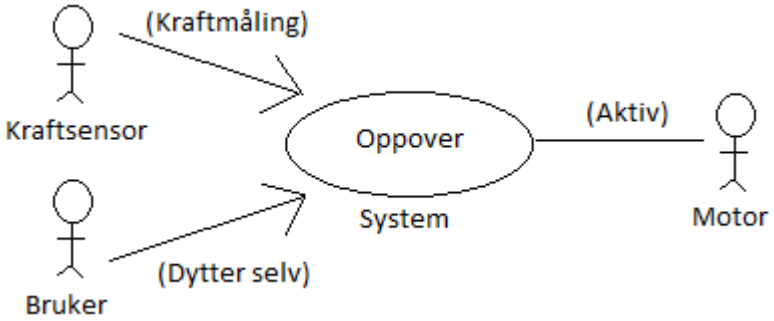
Innkobling av motorene vil skje i overgangen mellom der man ikke trenger aktiv støtte og der man trenger det. Dette defineres som et eget Use-case da man her har selve intelligensen i systemet og en helt egen situasjon, se Figur 15. I overgangen mellom passive og aktive motorer er det målingen av input gjennom en kraftsensor som styrer om motorene skal være aktive eller passive. Dette kan ses på binært med at de kobles ut eller inn. Kravet til algoritmen vil her være at man definerer en grense for innkobling som man kommer frem til gjennom grundig uttesting.



Figur 15 Innkobling-senarioet

3.2.3 Use-case 3 Oppoverbakke

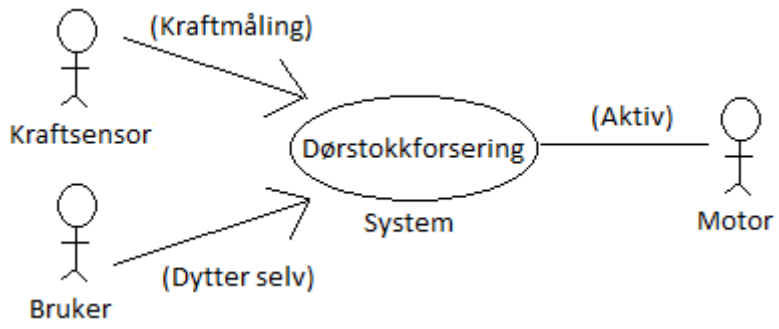
I oppoverbakke er rullatorens oppgave å gi brukeren aktiv fremdrift ved hjelp av deteksjon av kreftene brukeren dytter rullatoren med, se Figur 16. Rullatoren stopper når brukeren ikke gir noe kraft i håndtakene så rullatoren vil ikke kjøre i fra brukeren. Farten vil være lineær med kraften brukeren gir så sant den er over innkoblingsgrensen, og rullatoren vil holde en jevn fart ved jevn kraft oppover bakken. Farten vil være uavhengig av hvilke last man eventuelt har belastet rullatoren med i vertikal retning, og vi oppnår det vi vil med rullatorsystemet.



Figur 16 Oppover-senarioet

3.2.4 Use-case 4 Dørstokkforsering

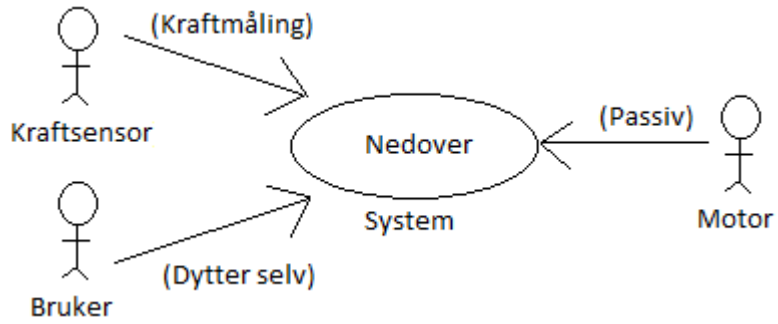
Når rullatoren møter en dørstokk og brukeren øker kraften mot håndtakene vil rullatoren gi aktiv fremdrift og hjelpe brukeren over dørstokken, se Figur 17. Hvordan dette vil fungere i praksis vil være situasjonsbestemt. I noen sitasjoner vil rullatoren fungere optimalt og i andre situasjoner vil rullatoren stå og spinne hvis hindringen er for stor. I en slik situasjon må brukeren løfte forhjulene over hindringen. Når bakhjulene møter dørstokken og kraftinputen stiger over innkoblingsgrensen, vil den i de fleste tilfeller ikke ha noen problemer med å kjøre over, da driften sitter i disse hjulene og diameteren på bakhjulene er relativt stor. Her kan kravet til vekten på batteriet gjøre seg gjeldene da den i dagens løsning er relativ høy. Dette kan virke problematisk hvis dette blir kombinert med lasting av rullatoren. En bør jo her ta høyde for at brukeren greier å løfte f.eks. sine egne matvarer, men at et tillegg i vekt fra batteriet kan gi problemer. Vekten på batteriet i dagens system er oppgitt til å være 6,4 kg[xii].



Figur 17 Dørstokk-senarioet

3.2.5 Use-case 5 Nedoverbakke

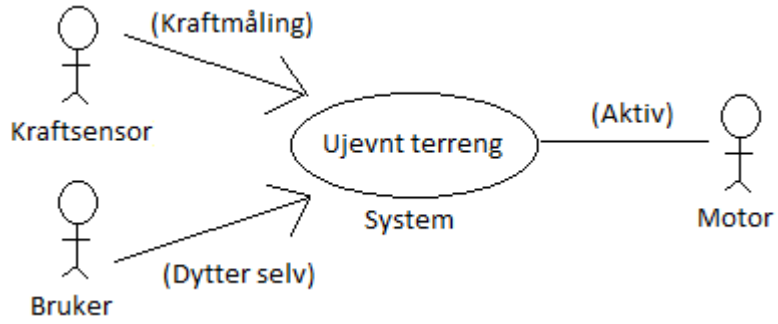
I nedoverbakke bør rullatoren fungere som en vanlig rullator og ikke ulikt Use-case 1, se Figur 18. Det betyr at det ikke vil være aktiv fremdrift, noe som vil være naturlig da tyngdekraften tar seg av fremdriften. I tillegg bør det være vanlig bremsefunksjon som gir trygghet og støttefunksjon.



Figur 18 Nedover-senarioet

3.2.6 Use-case 6 Ujevnt terreng

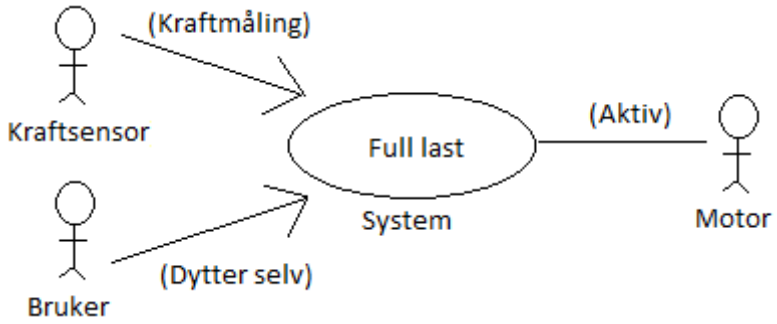
Ved bruk i ujevnt terreng og f.eks. snødekke eller grus vil behovet for aktiv fremdrift være der, se Use-caset i Figur 19. Dette senarioet blir mye likt Use-case 2 og Use-case 3, innkobling og kjøring i oppoverbakke. Reguleringsstrategien bør være den samme.



Figur 19 Ujevnt terreng-senarioet

3.2.7 Use-case 7 Full last

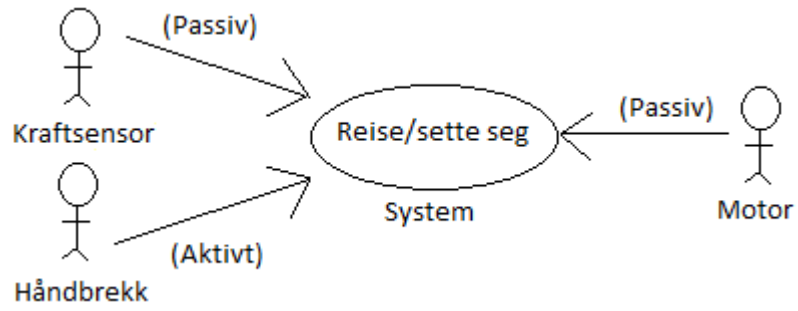
Ved full last kan kreftene man trenger for å dytte rullatoren bortover tilsvare kreftene man trenger for å dytte rullatoren i oppoverbakke, se Figur 20. Man vil da kunne trenge aktiv fremdrift i denne situasjonen og det vil skje hvis man har en kraftterskel i programmet som overstiges. Rullatoren vil da gi aktiv fremdrift og man kan laste så mye som rullatoren tåler i vertikal retning uten at brukeropplevelsen i horisontal retning vil endre seg nevneverdig.



Figur 20 Full last-senarioet

3.2.8 Use-case 8 Reise seg opp, sette seg ned

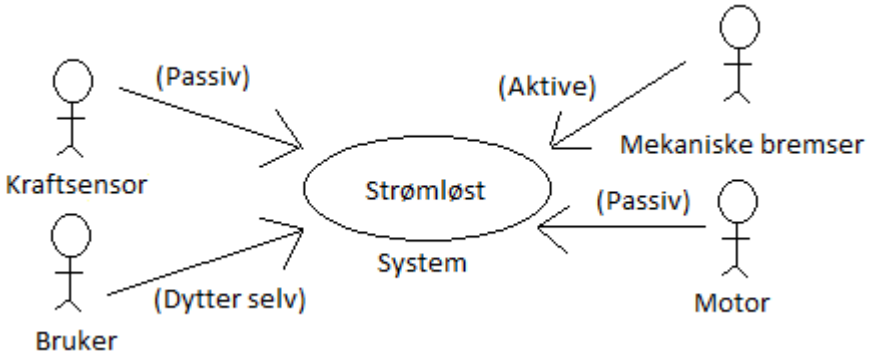
Når brukeren skal reise seg opp, eller sette seg ned bør ikke rullatoren kunne bevege seg, se Figur 21. Et mekanisk håndbrekk bør da kunne aktiveres slik at den ikke kan flytte seg før brukeren er klar. Det vil da også være behov for å koble ut motorene slik at rullatoren ikke beveger seg hvis sensorene for fremdrift blir belastet. Dette kan løses på flere måter, f.eks. ved at når håndbrekket kobles inn kobler motorene ut, som en ren I/O-løsning, eller ved bruk av strekklapper som måler analogt den vertikale belastningen på rullatoren. Man kan da sette en grense som den analoge verdien må over for å koble ut motorene, men den mest robuste løsningen vil nok være en I/O-løsning.



Figur 21 Reise/sette seg-scenariet

3.2.9 Use-case 9 Strømløst

Rullatoren har et batteri som selvfølgelig har sin begrensning og når den går over i strømløs tilstand skal rullatoren fungere som en vanlig rullator med frigang på hjulene, se Figur 22. I tillegg må bremsene være mekaniske slik at man ikke er avhengig av strøm for å få bremset eller låst hjulene. Det er viktig at denne løsningen blir like robust som en original rullator slik at ikke brukeren opplever denne innarbeidede løsningen som annerledes. Bremser omhandles som tidligere nevnt ikke i denne oppgaven.



Figur 22 Strømløst-scenariet

3.3 Sammendrag av Use case-analysen

Use case-analysen har beskrevet alle de brukerscenarier rullatoren kan komme opp i og belyst de utfordringer det er ved de forskjellige.
Når man ser på Use case-analysen opp mot rullatorløsningen, som HiST-studentene har laget, er det to oppgaver som utpeker seg. En mekanisk oppgave som er muligheten for at hjulene skal kunne rulle fritt og en programmeringsoppgave som går på å lage en terskelgrense for innkobling av motorene.

3.4 Teknisk utstyr

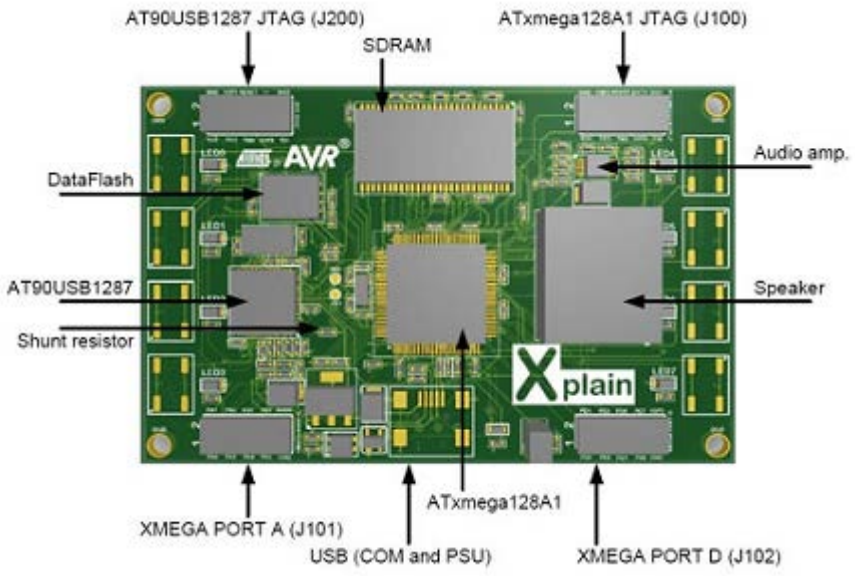
3.4.1 Mikrokontrollerkort ATxmega128A1

Mikrokontrollerkortet i rullatoren er fra produsenten Atmel og heter ATxmega 128A1[xv], se Figur 23. Dette kortet har mulighet for mange forskjellige funksjoner og man har for eksempel egne pinner som er dedikert for analoge signaler inn på kortet og egne pinner for UART-kommunikasjon [xiii]. Det sitter åtte lysdioder på toppen av kortet og disse kan brukes til flere ting, blant annet signalisere ulike tilstander på inn- og utganger. Det går også an å koble på et touchpanel på mikrokontrolleren, slik at man har mulighet til å styre mikrokontrolleren via denne.

For å programmere kontrolleren er det brukt et program som heter AVR Studio. Her skriver man i C-kode og overfører programmene til mikrokontrolleren ved bruk av en debugger, kalt AVR JTAGICE mkII[xiv]. Med denne kan man også simulere i sanntid på mikrokontrolleren. Dette kan være svært nyttig hvis man for eksempel skal utføre feilsøking.

Her er noen spesifikasjoner for kortet [xv]:

- Lavt strømforbruk
- Høy ytelse 8/16-bit AVR mikrokontroller med 128KB flash program minne
- Maks I/O-pinner: 78
- UART-porter: 8
- AD/DA-porter: 8
- Driftsspenning: 0-12 MHz @ 1.6-3.6V
  0-32 MHz @ 2.7-3.6V



Figur 23 ATxmega128A1 Mikrokontrollerkort[xv]

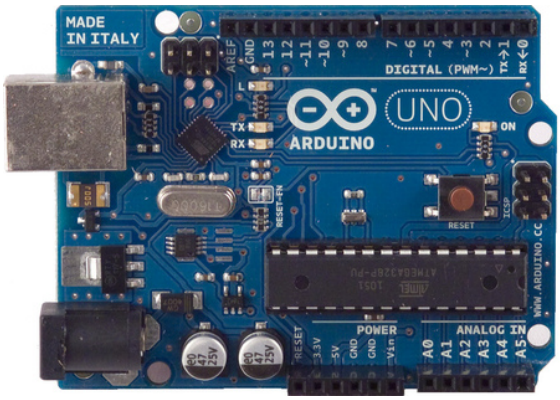
3.4.2 Loggekort Arduino UNO

Arduino UNO er et kort som har en ATmega328-mikrokontroller[xvi], se Figur 24. Kortet har 14 digitale inn/utganger, 6 analoge innganger, en 16 MHz krystall oscillator, en USB-tilkobling, en kontakt for ekstern spenningsforsyning og en reset-knapp. Dette kortet ble brukt fordi det oppstod problemer med å få lest av sanntidsinformasjon gjennom JTAG'en. Da var det enklere å ta et eksternt kort for å logge signalene. Signalene som ble logget var fra inngangssignalet fra høyre FSR-sensor og pådraget til høyre motor. Disse målingene ble brukt for å analysere inn – og utgangssignalene til systemet. Man kunne da se når motorene koblet inn og hvor stabilt utgangssignalet var med tanke på tuning av regulatoren.

De analoge inngangene kan maks måle 5 VDC. Da signalet til motorene ble målt til 7 VDC, måtte det kobles inn en spenningsdeler for å få dette signalet under 5. For enkelhetsskyld ble det koblet inn like motstander slik at dette signalet ble delt på to og logget med en av inngangene på kortet. Resultatene ble så skrevet til en fil i to kolonner slik at de ble enkle å visualisere i MATLAB.
For komplett kode til loggekortet, se vedlegg B.

Her er noen spesifikasjoner for kortet [xvi]:

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz



Figur 24 Arduino UNO[xvi]

4. Forslag til løsning / design

Med litteraturstudie og use case analysen som underlag er det to oppgaver som skal løses/prøves ut:
Montering av hjul som ruller kun en vei og videreutvikling av algoritmen. Algoritmen må testes ut og tunes slik av rullatoren oppfører seg slik vi vil ved å gi aktiv fremdrift i bakker, ved full last, ved ujevnt terreng og snø o.l. Disse oppgavene er valgt for å forbedre rullatoren og komme nærmere et intelligent produkt. De valgte tekniske løsningene vil kunne gi rullatoren en mer naturlig funksjon og vil være energibesparende. Den vil bli energibesparende i form av at motoren ikke lengre trenger å være koblet inn til en hver tid og at algoritmen hindrer den i å koble inn på flatmark.
Kapittel 4.1 tar for seg hjulene med frinav som sikrer at man kan bruke rullatoren uten strøm. Kapittel 4.2 tar for seg krav til reguleringsalgoritmen, dynamikken til rullatoren og endringer av algoritmen. Kapittel 4.3 tar for seg forbedring av håndtakene som ble laget i høstprosjektet[xvii].

4.1 Hjul med frinav

Tanken bak hjul med frinav kommer fra Use Case-analysen der man vil at rullatoren kan brukes i strømløs tilstand og at den kun skal gi moderat støtte. For at dette skal kunne fungere i praksis må hjulene kunne rulle fritt når det ikke er nødvendig med motorbidrag og kunne bli dratt rundt av motoren når dette trengs. I arbeidet med å finne slike hjul ble det undersøkt etter systemer som allerede hadde denne funksjonen. Det ble først sett på hjullagre som kunne monteres rett inn i de eksisterende hjulene. Disse viste seg ikke å være en god løsning, både med tanke på serieproduksjon, da det ville blitt et veldig komplekst system, og med tanke på pris[xviii]. Etter å ha undersøkt diametere og priser viste det seg at det ikke var hensiktsmessig å benytte dette. Det ble så søkt etter hjul i systemer hvor slike lagre allerede ble brukt og det viste seg at rulleskøyter og rulleski hadde akkurat denne funksjonen. De triller den ene veien, og låses automatisk fra å trille bakover.
For å få et robust og egnet hjulsystem ble det valgt hjul for terrengrulleski, også kalt «skike»[xix], se Figur 25 på neste side. Disse har en indre diameter for 8mm. aksel, mens den originale akseldiameteren til motoren er på 12,67mm (0,5"). Det må derfor monteres en overgangshylse for å få diameterne til å samstemme.

Ved å implementere hjul med frinav fjernes helt muligheten for å bremse rullatoren med motorene. Dette gjør at man må montere mekaniske bremser slik man har på en original rullator. Dette vil ikke bli gjort i dette prosjektet. I programmet som er tatt over etter HiST-prosjektet[xx] hadde de bremsene montert til to potensiometre og ligningen for utgangen så da slik ut:

$$Utgang = Pådrag - Brems \qquad (1)$$

Den nye løsningen vil da ikke inneholde brems så alt som har med bremsing å gjøre gjelder da ikke i programmet. Det vil ikke bli tatt bort, men kommentert ut hvis man senere vil gå tilbake og implementere en form for elektrisk brems.

Figur 25 Skikehjul[xix]

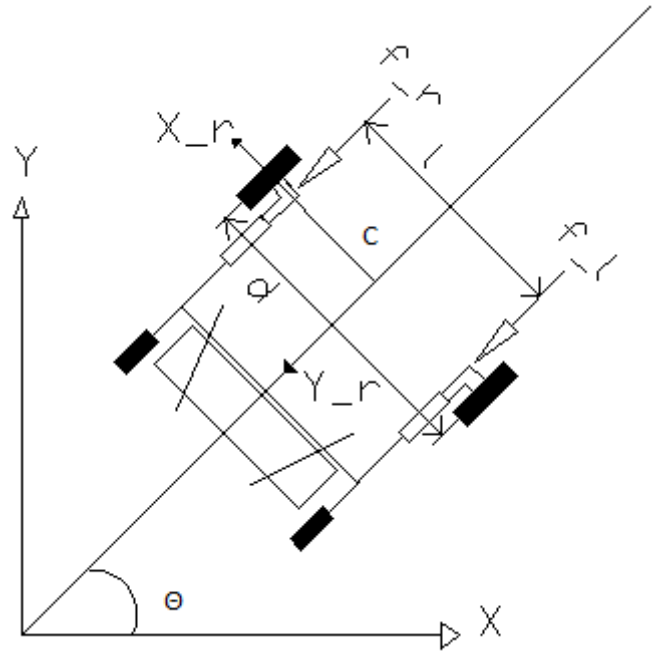
4.2 Reguleringsalgoritme

Før dynamikken og reguleringsalgoritmen blir drøftet er det tre ting som må ligge til grunn for kontroll av en aktiv rullator.

1. *Sikkerhet:* Primærmålet til en algoritme som skal styre et aktivt robotsystem må være sikkerheten.
2. *Manøvrering:* En god manøvrering vil være med å forbedre brukeropplevelsen.
3. *Intelligens:* Som et aktivt og adaptivt hjelpemiddel bør rullatoren ha en form for intelligens i støttefunksjonene og valgmulighetene.

4.2.1 Dynamikken til rullatoren

For å få en oversikt over dynamikken og frihetsgradene til rullatoren er det her satt opp som et vanlig bevegelsessystem for en vogn. Bevegelsene til et slikt system er den retningen den blir dyttet og ingen rotasjon, da det ikke er laget noe program for sving. I rullatorens tilfelle settes koordinatsystemet mellom hjulene med x og y-retning, som vist i Figur 26.



Figur 26 Rullatoren satt opp i et koordinatsystem

Definerer og setter så opp ligningen for bevegelsen til rullatoren.
Definerer D_r som dempekoeffisient. Fart og akselerasjon defineres som $\dot{q}$ og $\ddot{q}$. Kraften som tilføres håndtakene defineres som f og i dette tilfelle blir det da $F = (f_l + f_r)$ Her ignoreres friksjonen i håndtakene.

Dynamikken for rullatoren kan da settes opp slik:

$$M(\ddot{q} - D) = F \qquad (2)$$

$$M = [m], q = [Y_r], D = [D_r]$$

Hvor

M er massen
q er posisjonen
D er dempningskoeffisienten

4.2.2 Endringer i reguleringsalgoritmen

Reguleringsalgoritmen som ble laget av studenter ved Høgskolen i Sør-Trøndelag våren 2011[xx] er bygget opp etter pensum de har lært der. Algoritmen har blitt brukt som en testalgoritme under prosjektperioden og har derfor kun blitt endret delvis for å teste ut nye reguleringsstrategier. Den er i hovedsak blitt brukt for å teste ut proof-of-consept løsninger.

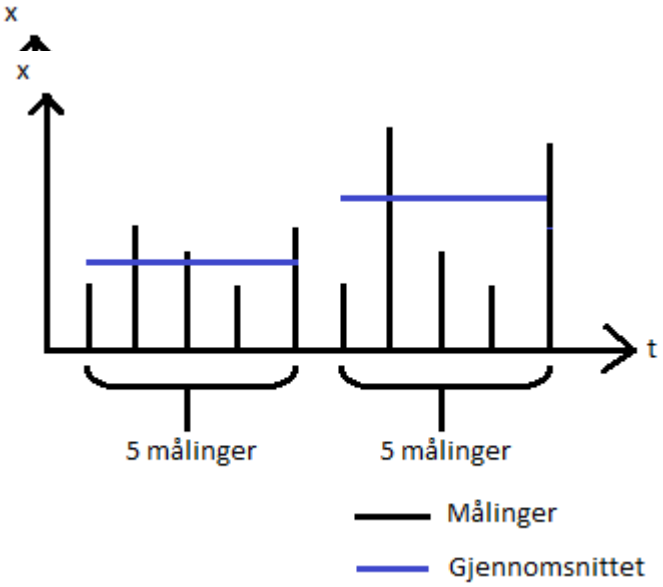
Reguleringssløyfene er bygget opp på følgende måte: Mikrokontrollerkortet sender en hastighetsreferanse til motorkontrollerne som så sender denne videre til motorene. Tilbakemeldingen på hastigheten kommer så fra et tacho-meter og motorkontrollerne regulerer et evt. avvik i hastigheten. Skulle rullatoren få for høy hastighet i forhold til farten brukeren ønsker å gå i, løser fjærreturen i håndtakene ut og sensorene mister sin input. Rullatoren vil da stoppe og dette blir da den «ytre» reguleringssløyfen. Motorkontrollerne og tachometerne fungerer da som den «indre» reguleringssløyfen og sammen danner de en form for kaskadekobling. Reguleringen inneholder et P-ledd og det er filtreringen av input og forsterkningen, kjent som *Kp* i tradisjonell reguleringsteknikk, som lar seg endre.

Endringer som har blitt gjort er på bakgrunn av litteraturstudie og use-case analysen. Før gikk motorene når kraftsensorene målte over 0 % og hadde et lineært område opp til 100 %. Nå har det blitt satt en terskelgrense som signalet fra kraftsensorene må over før motorene slår inn. Filtret som tar opp inputsignalet er også endret for å prøve ut et filter med glemmefaktor for å få et mer stabilt motorpådrag.

4.2.3 Gammel løsning

I HiST-prosjektet[xx] tok de inn inngangssignalet i noe de kalte ”Midningsfilter” hvor de logget fem målinger, fant gjennomsnittet av disse for så å sende dette signalet til utgangen. Tankegangen bak dette var at rullatorsystemet skulle være helt sikker på at brukeren ville kjøre før den ga signal. Dette fungerte ganske bra, men ved kjøring virket hastigheten noe ustabil og ga en følelse av rykk-og-napp-kjøring. Programmet fungerte heller ikke optimalt da den mellomlagret disse fem målingene og man hadde ikke sikret seg mot at disse ble slettet når man skrudde på rullatoren. Dette gjorde at når man skrudde på rullatoren begynte den å bevege på seg hvis det var lagrede verdier i filtret.

Man ville derfor aldri være helt sikker på hvordan rullatoren oppførte seg, og den oppførte seg litt forskjellig hver gang man startet den. Hvordan disse fem målingene lagres er vist i et fiktivt eksempel i Figur 27.



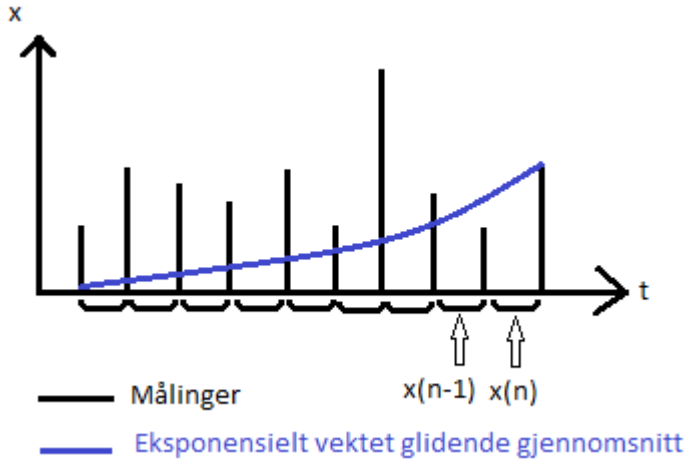
Figur 27 Gammelt "Midningsfilter"

4.2.4 Nytt filter med glemmefaktor

For å få en bedre reguleringsstrategi er forslaget å bytte ut det gamle filtret med et filter med glemmefaktor. Inspirasjonen for dette filteret er hentet fra faget TTK4195 Systemidentifikasjon og Adaptiv Regulering og kan kalles et *adaptivt eksponentielt vektet glidende gjennomsnittsfilter*, og som har følgende ligning:

$$\bar{x}_n = ax_n + bx_{n-1} \qquad (3)$$
$$b = 1 - a$$

Det er adaptivt i form av at målingen x_n kommer fra sensorene som brukeren påvirker med å tilføre kraft. Her er $\bar{x}_n$ summen av nyeste og forrige måling ganget med to glemmefaktorer, a og b. Målingene blir satt til 0 i starten slik at man unngår at rullatoren har noen verdi lagret fra forrige gang den var i drift. A og b er faktorer som brukes til å vektlegge hvilke av målingene man vil at skal gi størst bidrag. Grunnlaget for at ligningen skal fungere er at summen av disse to faktorene er under 1 slik at man ikke får en forsterkning. Hvis man vil ta høyde for enda eldre målinger kan denne rekken bare utvides med enda en konstant, c, og ta med x_{n-2} i ligningen osv. På denne måten kan man styre hvor fort inngangssignalet skal bygge seg opp før det gir signal til motorene.
Dette gjøres for begge håndtakene før de blir addert og delt på to for å få gjennomsnittet. Dette resultatet sendes så til utgangen for pådrag til motoren. Man kan så styre hvor raskt man vil at motorene skal akselere ved å vektlegge den første eller andre målingen ved og justere a og b. Hvordan disse målingene påvirker filteret kan man se i et fiktivt eksempel i Figur 28.



Figur 28 Filter med glemmefaktor

Fremgangsmåten for å finne faktorene a og b må gjøres gjennom en prøv-og-feil metode for å finne de innstillingene som gir best brukeropplevelse. For at rullatoren ikke skal reagere på små impulser og at akselerasjonen ikke skal oppleves for brå, bør den eldste av de to målingene vektlegges. Det må da vedvarende kraft til for at rullatoren skal bevege seg og akselerasjonen vil bygge seg opp jevnt. Dette bør være slik for at den tiltenkte brukergruppen, eldre, som erfaringsmessig ikke liker at ting går for fort eller oppfører seg brått, får en god brukeropplevelse.

4.2.5 Innkoblingsgrense for motorene

Reguleringsstrategien til rullatoren skal være slik at den kun hjelper til når den målte kraften kommer over en viss grense. Det gjør at motorene ikke unødvendig kobler inn når man går på flatmark, med lav friksjon, og belaster batteriet. Ved gange på flatmark vil en naturlig gange gi kraftimpulser inn i systemet gjennom håndtakene. Disse impulsene vil ha relativ kort lengde og er ikke ønsket å gi noe pådrag til motorene. Disse impulsene lar seg ikke filtrere vekk med forslaget som presenteres her, men vil gi små, nærmest ubetydelige bidrag når man legger vekt på den eldste av målingene i glemmefaktorfilteret.

Utgangen på mikrokontrollerkortet har et område fra 0-255 bit. Denne er ganget med en verdi slik at utsignalet maks kan gi en bit-verdi på 30 ut til motorene. Dette er en omregning og et valg som ble gjort av prosjektgruppen ved HiST da de synes at dette var en normal makshastighet på rullatoren. Denne verdien kan regnes om til omdreininger på motoren og videre til hastigheten rullatoren får ved maks pådrag. Økes denne verdien får rullatoren en høyere makshastighet og motsatt ved å senke den. Dette gjelder også for inputsignalet så verdiene som kommer fra sensorene i håndtakene blir ganget med en faktor for å holdes mellom 0 og 30.

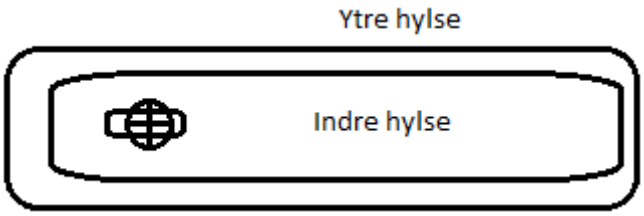
Da den nye løsningen på sensorene[xvii] ble testet opp mot den gamle[xx] viste det seg at FSR-sensorene er veldig ømfintlige ovenfor deteksjonsflaten[xxi] og at den nye ikke ga like god respons. Det gjorde at dette signalet måtte ganges med et litt større tall for å få like god respons på kraften som ble tilført. Dette gjør at grensen for innkobling av motorene må settes litt høyere da inputverdien stiger raskere i verdi enn den ville gjort med en lavere faktor. Disse endringene må gjøres gjennom en prøv-og-feil strategi for å finne den verdien som gjør brukeropplevelsen til rullatoren best mulig. Endringene gjøres i programmets deler «drift» og «tomgang», se vedlegg A for full kode.

4.3 Stopperfunksjon til håndtakene

Håndtakene som ble laget i høstprosjektet[xvii] ble laget som en hylseløsning hvor holken man holder i sitter på en bevegelig del og sensoren sitter i en fast del montert vertikalt på enden av rammen, inne i holken. Håndtaket lot seg demontere ved å dra holken bakover da man enklere kunne justere på sensorene og demontere rammen under kabling. Det ble ikke laget noe stopperfunksjon, noe som har resultert i at det ikke har vært så enkelt å håndtere rullatoren. Håndtakene har kunnet roteres rundt den indre hylsen og løsningen har virket lite robust. Ved å dra av håndtakene har kabelen til sensorene løsnet fra tilkoblingen og man må demontere rammen for å kunne dra kabelen tilbake for å kunne koble de sammen igjen.

Håndtakene bør derfor forbedres slik at en stopperløsning hindrer at man kan dra hylsene fra hverandre. Dette skal være en løsning som kan skrus ut og inn slik at håndtaket kan demonteres ved eventuelle endringer. Skisse over planlagt løsningen kan ses i Figur 29.

En skrue låser da den ytre holken som man holder i, og hindrer den i å kunne dras av og fra å kunne roteres. En slik løsning vil være mye mer robust og det vil bli enklere å håndtere rullatoren når man f.eks. skal svinge.
Da rullatoren ikke lar seg trille bakover på grunn av de nye hjulene med låsing bakover, vil det i mange situasjoner være behov for å snu rullatoren. Det er derfor viktig at håndtakene oppfattes som robuste og gode å holde i.



Figur 29 Stopperfunksjon til håndtakene

5 Implementering

Implementeringen er delt i to deler. En mekanisk del og en programmeringsdel. Den mekaniske har vært utskiftning av hjul og stopperfunksjon til håndtakene. Programmeringsbiten har vært implementering av innkoblingsgrense og nytt filter med glemmefaktor.
Kapittel 5.1 tar for seg implementering av de nye hjulene, 5.2 tar for seg det nye filteret med glemmefaktor, 5.3 tar for seg innkoblingsgrensen og 5.4 tar for seg stopperfunksjonen til håndtakene.

5.1 Implementering av hjul med frinav

Hjulene er kjøpt inn fra firmaet *X-skating* i Tyskland og er hjul for terrengrulleski, kalt *Skike*[xix]. Disse ble valgt da de hadde den funksjonen vi var ute etter og samme ytre diameter som de gamle hjulene, slik at det ikke skulle bli noe konflikt med hastigheten. Som nevnt i kapittel 4.1 var den indre diameteren forskjellig fra de gamle hjulene noe som gjorde at det måtte lages en overgang for å få de til å passe. Denne overgangen måtte monteres på utsiden av akselen fra motoren for å få godt nok hold i konstruksjonen da de nye hjulene nå er montert litt lengre ut i fra rammen, se Figur 30. Overgangene er laget i aluminium og har en sett-skrue som kan skrus ut ved demontering.
De nye hjulene ferdig montert kan ses i Figur 31. Det mekaniske arbeidet ble utført ved verkstedet til Instituttet for Teknisk Kybernetikk.

Figur 30 Overgangshylse til nye hjul

Figur 31 Ferdig monterte hjul med frinav

5.2 Implementering av filter med glemmefaktor

Det nye filteret er skrevet i kode med grunnlag fra ligning (3). Det gamle programmet[xx] var slik at det ikke sikret seg mot å slette gamle verdier i filtret slik at rullatoren beveget seg et lite stykke under oppstart, som tidligere nevnt i kapittel 4.2.3. Dette er også rettet opp i som man kan se av de to første linjene av koden under.

Slik ser den spesifikke delen av koden ut i dag:

```
delsvar=0;

delsvar2=0;

void Glemmefaktor(int *delsvarP, int *delsvar2P, int
*delsvar3P, int *delsvar4P, int samplesP, int *ResultVoltP,
int *ResultVolt2P, int *Brems)

{

ADCA.CH0.CTRL |= ADC_CH_START_bm;                //høyre FSR

  *delsvarP = 0.3*(ADCA.CH0RES*0.204)+0.7*(*delsvarP);

ADCA.CH1.CTRL |= ADC_CH_START_bm;                //venstre FSR

  *delsvar2P = 0.3*(ADCA.CH1RES*0.204)+0.7*(*delsvar2P);

}
```

Av de to linjene hvor delsvarene fra venstre og høyre FSR blir utregnet ser vi at den eldste av de to målingene blir mest vektlagt. Når rullatoren startes opp er målingen som leses inn på pinne 2 og 4 på Port A på mikrokontrollerkortet, kalt ADCA.CH0RES og ADCA.CH1RES, lik 0. Når håndtakene blir påtrykket kraft leses signalet inn her og vektlegges med en faktor på 0,3. Ved neste sampling blir disse delsvarene til delsvarP og delsvarP2, og nye målinger tas inn. Den forrige blir så vektlagt med en faktor på 0,7 og sånn bygges pådraget opp. Man vil da få et jevnere signal og forhåpentligvis et mer stabilt motorpådrag.

5.3 Implementering av innkoblingsgrense

For å finne en innkoblingsgrense som gjorde at motorene ikke koblet inn før den oversteg en grense, ble dette gjort ved prøving og feiling. Denne grensen måtte være høy nok til at motorene ikke koblet inn på flatmark ved lav friksjon, men ikke så høy at kraften som blir tilført i f.eks. relativt slake bakker blir kuttet bort. Etter testing og utprøving ble denne grensen satt til 13 % av full skala for inngangen. Det vil si en tallverdi på 4 i programmet da bit-verdien blir regnet om slik at den går mellom 0 og 30.

Den nye innkoblingsgrensen kan ses som pseudokode i koden under, for full kode, se vedlegg A.

```
void Drift(int ResultFremP)

{

  if(ResultFrem>4)

(Vedlegg A)

}

void Tomgang(int ResultFremP)

{

 if(ResultFremP<=4)

(Vedlegg A)

}
```

5.4 Implementering av stopperfunksjon til håndtakene

Designet og utførelsen på stopperfunksjonen ble implementert slik det var planlagt og tilfredsstiller de krav som var satt. Det ytre håndtaket er nå låst i et spor i den indre hylsen slik at de ikke kan dras bakover. Håndtakene er robuste og lar seg demonter med en standard stjernetrekker. Den ferdige løsningen kan ses i Figur 32. Verkstedet til Instituttet for Teknisk Kybernetikk har utført de mekaniske endringene.

Figur 32 Ferdig montert stopperfunksjon til håndtakene

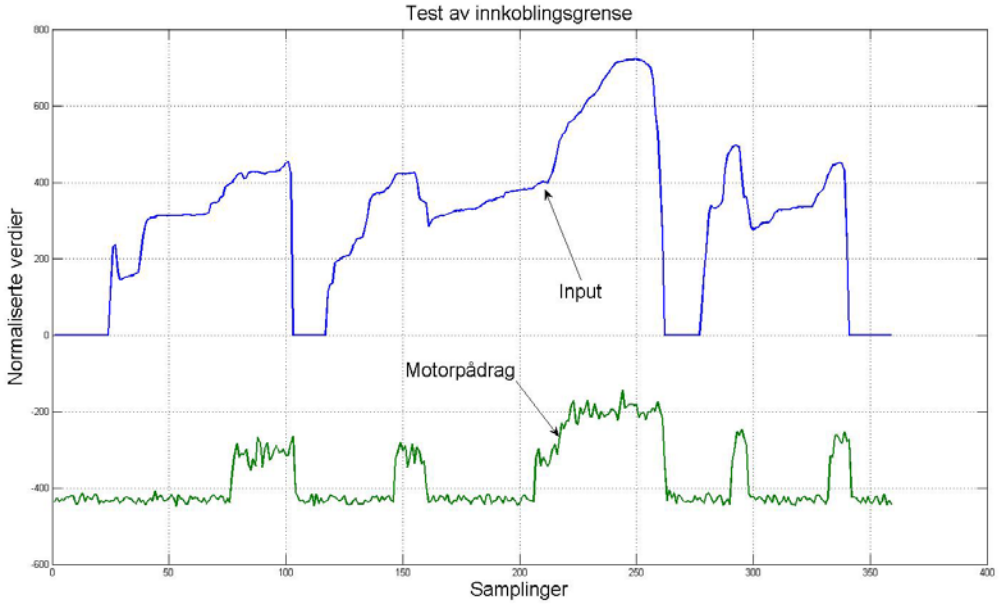
6 Uttesting / Resultater

I dette kapittelet vil man se hvilke tester som er gjort, hvorfor de er gjort og resultatene fra de testene som er gjort. De enkelte testene er gjort med bakgrunn i de brukerscenariene som ble analysert i Use Case-analysen. Det er i tillegg gjort en test hvor man ser motorens innvirkning på den kraften som trengs for å kjøre rullatoren i en bakke. Testene er utført i samme bakke og tidsintervallet på testene er prøvd og holdt relativt likt. Loggingen er utført med kortet Arduino UNO[xvi] som er beskrevet i kapittel 3.4.2. Det er som beskrevet i kapittel 3.4.2 kun høyre FSR og høyre motor som er målt og visualisert her. Y-aksen er ikke skalert ut i fra noen fysiske mål og er kun numeriske verdier som er logget fra Arduino UNO kortet. Det ble også prøvd å måle hastigheten med en Android-telefon som hadde et akselerometer hvor resultatene ble logget i en fil for så å bli integrert opp i Matlab. Dette viste seg å være meget vanskelig da man summerer opp og drar med seg eventuelle feil inn i neste resultat. Det er derfor ikke referanser til hastigheter i resultatene. Det ble heller prøvd å gå i samme hastighet når man skulle ha flere resultater inn i samme graf. Det fallet man kan se i input etter at rullatoren er satt i bevegelse svarer til kreftene som trengs for å akselerere noe opp i hastighet er høyere enn kreftene som trengs for å holde noe som er i bevegelse, i gang.

6.1 Test av innkoblingsgrense

Denne testen er utført for å vise innkoblingsgrensen i praksis og svarer til Use-case 2 og Use-case 6. Man får her testet innkoblingsgrensen, men også simulert hvordan rullatoren vil oppføre seg ved kjøring i ujevnt terreng. Det er påført stigende trykk på håndtaket for å vise når motorene kobler inn. For å se når motorene kobler inn se Figur 33. Det er ikke henvist til noen verdi i Y-aksen på når motorene kobler inn, da dette kun er numeriske verdier hentet fra Arduino UNO kortet.

6.1.1 Resultat fra test av innkoblingsgrense



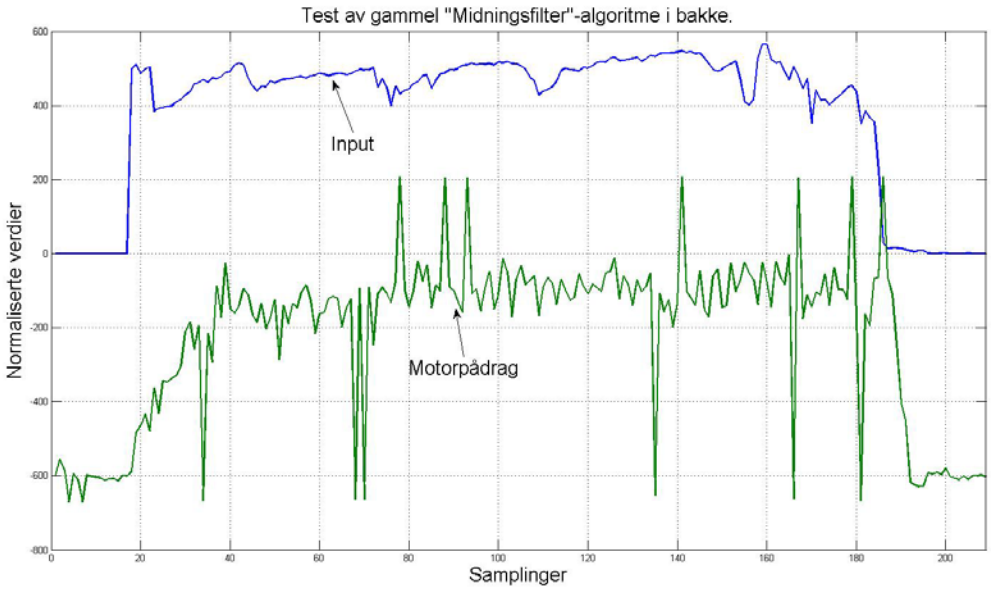
Figur 33 Test av innkoblingsgrense

Da dette er normaliserte verdier og trykket er påtrykt med håndmakt er det ikke mulig og si noe eksakt om hvor mange grams trykk det skal til for at motorene kobler inn. Det som er viktig å lese av Figur 33 er at man må over en innkoblingsgrensegrense før motorene kobler inn. Denne grensen er som sagt funnet gjennom testing hvor rullatoren hadde en god funksjonalitet og brukeropplevelse. Dette svarer til den ønskede og praktiske oppførselen fra systemet og resultatet viser at algoritmen fungerer godt.

6.2 Test av gammelt «Midningsfilter»

For å teste den nye algoritmen opp mot den gamle er det utført en test for å visualisere hvordan pådraget til motoren ser ut med det gamle filteret. Resultatet kan ses i Figur 34 under.

6.2.1 Resultat fra test av gammelt «Midningsfilter»



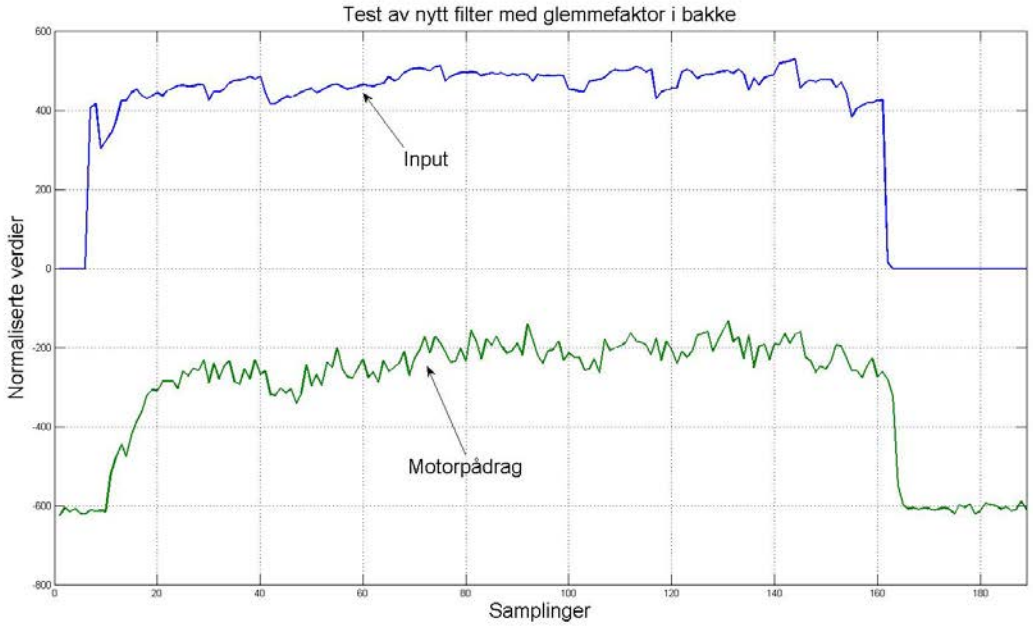
Figur 34 Test av gammelt "Midningsfilter"

Som vi ser av Figur 34 er motorpådraget med det gamle filteret relativt stabilt, men gir store utslag ved små endringer i input. Dette gir ikke en god brukeropplevelse og rullatoren gjør rykk-og-napp bevegelser uten at man har endret i stor grad på inngangen.

6.3 Test av nytt filter med glemmefaktor

Det nye filteret er testet i samme bakke som det gamle for å visualisere hvordan pådraget til motoren ser ut med det nye filteret. Dette er rullatorens hovedfunksjon og svarer til Use Case 3. Resultatet kan ses i Figur 35 under.

6.3.1 Resultat av test av nytt filter med glemmefaktor



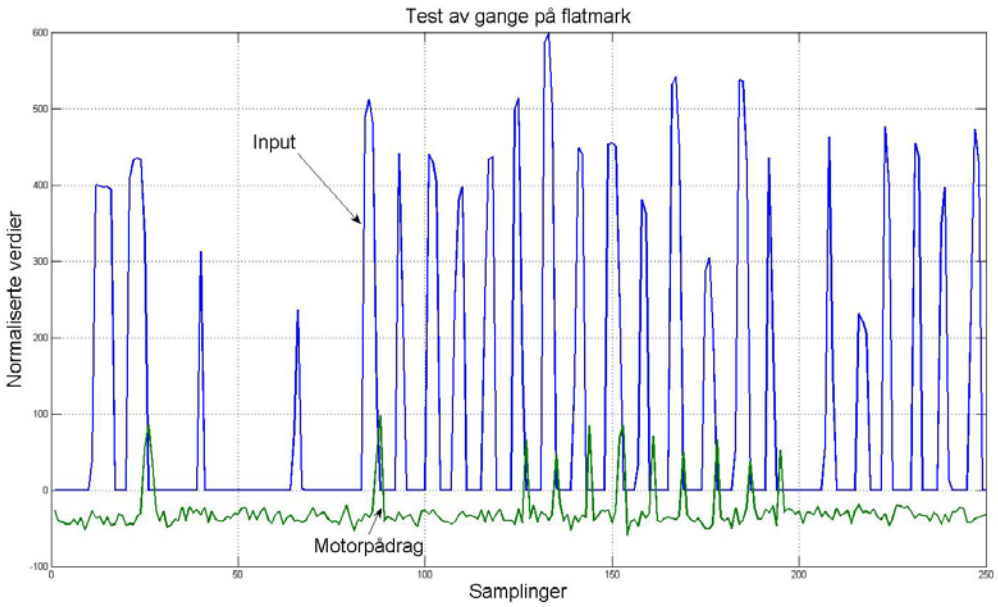
Figur 35 Test av nytt filter med glemmefaktor

Som vi ser av Figur 35 er det nye filteret mye mer stabilt og gir et mer stabilt motorpådrag. Man kan se på sprangresponsen at motorpådraget bygges gradvis opp og holdes stabilt. Brukeropplevelsen er bedre og rullatoren har en jevn og behagelig fart.

6.4 Test av gange på flatmark

Denne testen er utført for å se hvordan rullatoren reagerer når man går på flatmark, uten friksjon slik som Use Case 1 beskriver. Impulsene på håndtakene ble utført noe overdrevet for å fremprovosere effekten av rykk-og-napp bevegelsene man har når man går. Resultatet kan ses i Figur 36 under.

6.4.1 Resultat fra test av gange på flatmark



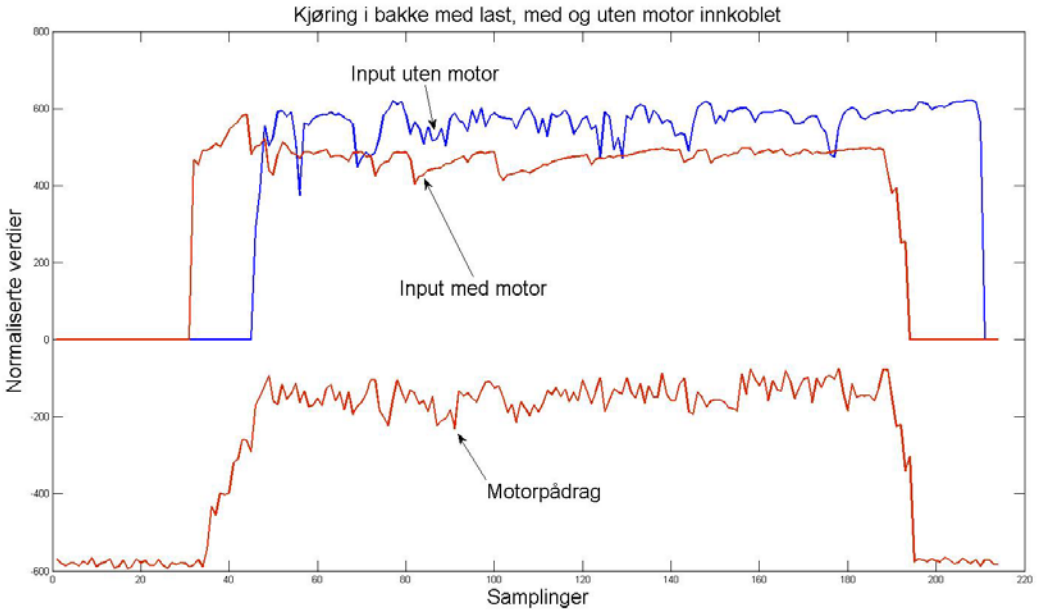
Figur 36 Test av gange på flatmark

Som vi ser av Figur 36 kobler motoren inn når impulslengden kommer over en viss lengde. Dette er egentlig ikke ønskelig, men når man går,vil dette pådraget være så lite at det ikke gir noe bidrag til farten på hjulene siden disse har frinav.

6.5 Test av motorenes innvirkning

Testen er utført for å vise motorenes innvirkning på den kraften som trengs for å kjøre rullatoren i en bakke. Det er forsøkt å gå med jevn fart opp en bakke, med og uten motorene innkoblet. Testen er utført med rullatoren lastet med tunge gjenstander og resultatet kan ses i Figur 37 under.

6.5.1 Resultat fra test av motorenes innvirkning



Figur 37 Test av motorenes innvirknig

I Figur 37 ser man resultatet av to forsøk hvor den blå streken er en test uten motorene koblet inn, og de to andre er input og output med motorene innkoblet. Kraften man trenger uten motorene koblet inn vil bli større enn det vi ser her, forutsatt at farten man går med er lik. Siden motorpådraget er lineært med kraftinputen vil ikke disse grafene avvike mye ved samme fart. Hadde man gått fortere med motorene innkoblet kunne det resultatet ligget høyere enn testen uten motorene innkoblet. Dette er utført som et praktisk eksperiment ,så god repeterbarhet er vanskelig å oppnå. Med det menes at det er vanskelig å gå opp en bakke med samme hastighet, og kraftinput, når man ikke har hastighetsmåling av rullatoren. Det testen beviser er at motoren lar deg gå i samme hastighet opp en bakke med last, men at man trenger mindre kraft for å få det til.

7 Diskusjon

Diskusjonsdelen er delt inn i de oppgavene som er løst i dette prosjektet og blir diskutert hver for seg. En kort kommentar er lagt til etter hvert resultat i kapittel 6 og det vil i dette kapittelet bli diskutert generelt rundt alle resultatene.

7.1 Hjul med frinav

De nye hjulene er montert litt ut i fra rullatoren noe som gjør at løsningen ikke er like god som en original rullator med tanke på momentet dette skaper ut fra rammen. Det mest ideelle hadde vært og hatt hjulene så nærme rammen som mulig, men det lar seg ikke gjøre med de akslene som originalt følger med de motorene som er montert. En vil derfor ikke kunne legge så mye press på håndtakene i vertikal retning som man kan med en original rullator. Når det kommer til vekten man kan belaste forhjulene med er det ikke noe avvik fra en original rullator, så kapasiteten for lasting av rullatoren med matvarer o.l. er ikke forringet med utskiftning til de nye hjulene. Prisen på dette hjulsystemet var også rimelig og tilfredsstiller SINTEF' krav om at komponentprisene skulle holdes lave for ikke å gjøre totalprisen høy med tanke på serieproduksjon.

7.2 Filter med glemmefaktor

I sammendraget av litteraturstudie ble det foreslått å benytte en regulator med akselerasjon som utgang. Under arbeidet med å se på algoritmen ble det under tester funnet ut at motorpådraget var noe ustabilt. Det ble da lett etter et nytt filter som gav en mer stabil oppbygning av inputsignalet og som man kunne styre mer selv. Ved tester av det nye filteret med glemmefaktor viste det seg at pådraget ble stabilt og behovet for å bruke tid på å utvikle en ny regulator ble mindre. Dette hadde nok vært spennende og prøvd ut, men tidspress og noe manglende kompetanse innenfor programmering gjorde at denne oppgaven ikke ble gjennomført. Resultatet fra det nye filteret, som kan ses i Figur 35, viser at motorpådraget har blitt mye mer stabilt og farten er derfor jevnere enn den var med det gamle. Det ble eksperimentert med forskjellige vektingsfaktorer, og det viste seg at en større vekting på den eldste av de to målingene som blir mellomlagret, ga best resultat. Filteret gir ikke bare stabil fart, men gir også en behagelig akselerasjon opp til den farten som er ønsket.

7.3 Innkoblingsgrense

Det å finne innkoblingsgrensen viste seg og ikke være så enkelt. Programmet HiST-studentene hadde laget var det ikke så logisk å endre på som det skulle virke som. De har satt en grense for innkobling og en grense for hva som skal være tomgang. Disse to ble prøvd å settes til en høyere verdi, men da koblet ikke motorene inn. Etter mye testing og utprøving viste det seg at bare å øke den med en i verdi ga et ok resultat til at den ikke begynte å bevege seg fra 0 % av full skala for inputen. Resultatet at denne innkoblingsgrensen kan ses i Figur 33.

7.4 Gange på flatmark

Testen som ble utført ved gange på flatmark ble gjort med ganske overdrevne rykk-og-napp-bevegelser for å se hvordan algoritmen håndterte dette. Som Figur 36 viser kobler motorene inn når lengden på input går over en viss lengde. Da bidraget fra motorene er såpass lite, vil ikke dette gi noe bidrag til farten på hjulene da disse vil ha høyere hastighet enn det akslene roterer med. Ved gange på flatmark og lav friksjon vil systemet i mange tilfeller ikke registrere noe input da fjærene i håndtakene vil holde holkene ute.

7.4 Stopperfunksjon til håndtakene

Stopperfunksjonen til håndtakene ble montert i hovedsak for å hindre at ledningen til sensoren ikke skulle falle ut da man håndterte rullatoren. Rullatorkonseptet er i hovedsak laget for å føre rullatoren fremover og med hjulene som låser bakover lar den seg ikke rulle bakover. For å snu den må den enten svinges rundt eller løftes rundt 180 grader. De er også montert for å få et brukergrensesnitt som ikke avviker for mye fra et originalt produkt. Da håndtakene ble montert i høstprosjektet var dette for å teste ut om konseptet fungerte, og det ble ikke lagt vekt på å lage en fast løsning da det var muligheter for at alt måtte demonteres igjen. Løsningen med en skrue som lar seg skru ut løser begge disse utfordringene og gir rullatoren et mer robust brukergrensesnitt. Håndtakene er fortsatt en bevegelig del, noe som ikke er likt originale håndtak, men dette er bevisst for og ikke få en offset på sensorene slik at rullatoren kan begynne å kjøre uten at noen betjener den. Mer om sensorene i håndtaksløsningen kan leses om i rapporten fra høstprosjektet[xvii].

8 Konklusjon

En kan konkludere med at implementeringen av hjul med frinav åpnet en del dører for hvordan styrestrategien og intelligensen til rullatoren kunne videreutvikles. Dette gjorde at man kunne implementere moderat støtte ved at rullatoren kun hjelper til når brukeren trenger det. Det gav også muligheten for at rullatoren kan brukes i avslått tilstand og hvis rullatoren går tom for strøm. Man må også konkludere med at implementeringen av hjul med frinav fjernet muligheten for elektrisk brems og at man nå må lage en mekanisk bremseløsning i en videre versjon.

Filteret med glemmefaktor har gjort brukeropplevelsen mer stabil og motorpådraget oppfører seg nå ikke lengre ustabilt, se Figur 35. Akselerasjonen opp til den farten man ønsker er også behagelig og rullatoren er ikke så brå i overgangene som før. Tuningen av glemmefaktorene ble gjort gjennom en prøv-og-feil-metode og kan enkelt endres på i ettertid hvis det er ønskelig.

Innkoblingsgrensen fungerer godt i praksis, men virker ikke like logisk i programmet. Grensen ble satt til det man trodde skulle være en logisk grense, men da viste det seg at inputsignalet ikke ble høyt nok til at motorene koblet inn. Etter mye testing ble den høynet med 1 fra den gamle løsningen og som Figur 33 viser, fungerer dette godt. En kan konkludere med at denne funksjonen har gjort rullatorsystemet til et mer intelligent system, som hjelper brukeren uten at de selv trenger å gjøre noe.

Stopperfunksjonen til håndtakene har gjort brukergrensesnittet mer robust. Dette gjør at håndteringen og bruken av rullatoren ligner mer på en original rullator enn før. Stopperfunksjonen kan enkelt demonteres med en standard stjernetrekker ved eventuell demontering. Denne løsningen er enkel og billig, helt i tråd med SINTEF sine krav.

En kan konkludere med at dette prosjektet har svart på de oppgavene som var gitt i oppgaveteksten fra SINTEF. Rullatoren har blitt utviklet videre til et bedre produkt enn da den ble tatt over fra HiST. De valg og løsninger som er gjort er gjort på bakgrunn av en grundig analyse over rullatorens brukerscenarier og ut i fra de ideene som kom gjennom litteratursøket. Rullatoren fremstår nå som et bedre og mer intelligent produkt enn da det ble startet på.

9 Videre arbeid

Det videre arbeidet bør i stor grad dreie seg om mekaniske endringer og vektreduksjon. Arbeidet som er gjort i dette prosjektet har fått det et stykke på vei, men det er fortsatt en vei frem til et produkt som eventuelt lar seg serieprodusere. Noen av ideene er ganske enkle å gjennomføre og innebærer bare programmering, mens andre krever innkjøp av nytt utstyr. Her er ideer for videre arbeid på rullatoren:

9.1 Mekaniske endringer

- Montere brems. Dette kan gjøres ved å montere på igjen de originale mekaniske bremsene, men det kreves nok litt justering for å få de til å fungere da de nye hjulene sitter lengre ut fra rammen enn de originale. Bremsehendlene til denne løsningen må i tillegg koble ut motordriften slik at man ikke kan bremse og gi motorpådrag samtidig. En slik enkel forrigling vil ikke være vanskelig å lage da det finnes flere ledige innganger på mikrokontrollerkortet.
- Bytte ut batteriet med et lettere ett. Dagens blybatteri har en vekt på 6,4 kg noe som er hovedvekten i systemet. Det gjør at rullatoren er tung å håndtere over dørstokker og ved generell håndtering i trapper o.l. Hvis størrelsen også reduseres kan instrumentkassen, som i dag virker veldig stor, reduseres.
- Montere den originale handleposen. Ved å senke instrumentkassen kan man få plass til den originale posen. Det bør også ses på en anordning slik at rullatoren lar seg klappe sammen når man ser på opphenget til instrumentkassen. Med dagens løsning er det ikke mulig og slå sammen rullatoren da kassen er skrudd fast i begge sider mot rammen.
- Lage rullatoren mer værbestandig. Dette kan gjøres ved å kapsle inn motorene og posisjonskortene med noe som har høyere IP-grad det det som er i dag. Dette gjelder spesielt under motorene der posisjonskortene er åpne mot bakken og det kan sette seg fast snø o.l.

9.2 Elektriske/program endringer

- Montere ny av/på-knapp. Dagens knapp er dårlig og man må dunke i den for å få rullatoren til å starte. Et forslag er å montere den oppe ved håndtakene slik at systemet enkelt lar seg skru av og på. Her må man tenke på plassering hvis rullatoren skal ha mulighet for å slås sammen.
- Programmere inn revers slik at den kan rygge. Det ble vurdert i høstprosjektet om man skulle montere kraftmålere som målte i begge retninger for så å bruke denne målingen til revers når brukeren dro rullatoren mot seg. Dette ble reflektert over og vi kom frem til at det ikke lot seg gjøre da man i nedoverbakke vil få utslag på denne måleren ved at man holder igjen rullatoren. Dette kan f.eks. løses med et gyroskop som indikerer at rullatoren er i nedoverbakke. Det enkleste her er nok å lage et separat system med en reversknapp som kjører rullatoren bakover. Denne reversknappen kan f.eks. sitte oppå «knekken» foran håndtakene slik at man kan manøvrere rullatoren samtidig.
- Implementere deteksjonssensorer som sikrer at brukeren står ved rullatoren ved bruk. Dette kan gjøres med f.eks. Infrarøde avstandssensorer.

9.3 Oppsummering av videre arbeid

Kort oppsummert foreslår vi at det arbeides videre med:

- Montere mekanisk brems som i tillegg kobler ut fremdriften.
- Bytte ut batteriet med et lettere ett og reduser størrelsen på instrumentkassen.
- Montere den originale handleposen og se på muligheten for å klappe sammen rullatoren.
- Lage rullatoren mer værbestandig.
- Montere ny av/på-knapp.
- Implementere revers.

Vedlegg A: Programkode for rullatoren

```
/*****************************************************************************
**
 *

 *                      -= Program for drift av rullator =-

 *
*****************************************************************************/

/***************************************
 *  HEADER&INCLUDE FILER OG DEFINISJONER *
 ***************************************/

#include <avr/io.h>
#include <stdbool.h>
#include <util/delay.h>

#define __AVR_ATxmega128A1__  1

#define BSCALE_VALUE -1
#define BSEL_VALUE 11

#define LEDPORT PORTE
#define BUTTONS PORTF
#define MOTORV USARTD0
#define MOTORH USARTD1
#define USART_PORT PORTD

/***************************************
 *                Pekere                *
 ***************************************/
void Deakselerasjon(int ForResP,int ResultFremP);
void Dodband(int ForResP,int ResultFremP);
void Gjsnitt(int ResultVoltP,int ResultVolt2P,int *ResultFremP,int Brems);
void Drift(int ResultFremP);
void Tomgang(int ResultFremP);
void Glemmefaktor(int *delsvarP,int *delsvar2P,int *delsvar3P,int
*delsvar4P,int samplesP,int *ResultVoltP,int *ResultVolt2P,int *Brems);
void Adc(int *ResultA,int *ResultB,int *Brems);
/***************************************
 *            Variabler til ADC          *
 ***************************************/
int ResultVolt;
int ResultVolt2;
int ResultFrem;
int ResultMellom;
int delsvar;
int delsvar2;
int delsvar3;
int delsvar4;
int samples;
int Resultat;
int ForRes;
int ForRes2;
int td=0;
int td2=2;
int Avvik;
int Brems;
int Brems2;
```

```
/****************************************
 *                MAIN                  *
 ****************************************/
int main(void)

{
 _delay_ms(5000);                        // Enabler lysdioder
 USART_PORT.DIRSET   = PIN3_bm;   // Pinne 3 (TXD0) as output.
 USART_PORT.DIRSET   = PIN7_bm;   // Pinne 2 (TXD1) as output.

 // USARTD0, 8 Data bits, No Parity, 1 Stop bit
 MOTORV.CTRLC = (uint8_t) USART_CHSIZE_8BIT_gc | USART_PMODE_DISABLED_gc |
false;
 // Setter Baud-Rate'n
 MOTORV.BAUDCTRLA = BSEL_VALUE;
 MOTORV.BAUDCTRLB = (BSCALE_VALUE << 4) | (BSEL_VALUE >> 8);
 // Enable both RX and TX
 MOTORV.CTRLB = 0x18;
 // ADC-SETUP

 // USARTD0, 8 Data bits, No Parity, 1 Stop bit
 MOTORH.CTRLC = (uint8_t) USART_CHSIZE_8BIT_gc | USART_PMODE_DISABLED_gc |
false;
 // Setter Baud-Rate'n
 MOTORH.BAUDCTRLA = BSEL_VALUE;
 MOTORH.BAUDCTRLB = (BSCALE_VALUE << 4) | (BSEL_VALUE >> 8);
 // Enable both RX and TX
 MOTORH.CTRLB = 0x18;
 // ADC-SETUP


 PORTA.DIR = 0;
      //Konfigurerer PORTA som input
 ADCA.CTRLA |= 0x1;                                          //Enabler
ADC
 ADCA.CTRLB = ADC_RESOLUTION_8BIT_gc;                      //8-bit
konvertering
 ADCA.REFCTRL = ADC_REFSEL_VCC_gc | 0x20;           //Setter VCC/1.6
referanse
 ADCA.PRESCALER = ADC_PRESCALER_DIV8_gc;             //Peripheral clk/8
 ADCA.CH0.CTRL = ADC_CH_INPUTMODE_SINGLEENDED_gc;   //Single ended
//Single ended
 ADCA.CH1.CTRL = ADC_CH_INPUTMODE_SINGLEENDED_gc;   //Single ended
 ADCA.CH2.CTRL = ADC_CH_INPUTMODE_SINGLEENDED_gc;   //Single ended
 ADCA.CH3.CTRL = ADC_CH_INPUTMODE_SINGLEENDED_gc;   //Single ended
 ADCA.CH0.MUXCTRL = ADC_CH_MUXPOS_PIN4_gc;          //PORTA:PINNE4
 ADCA.CH1.MUXCTRL = ADC_CH_MUXPOS_PIN2_gc;                 //PORTA:PINNE2
 ADCA.CH2.MUXCTRL = ADC_CH_MUXPOS_PIN5_gc;                 //PORTA:PINNE5
 ADCA.CH3.MUXCTRL = ADC_CH_MUXPOS_PIN7_gc;                 //PORTA:PINNE7

/****************************************
 *       STARTUP SEKVENS FOR MOTOR      *
 ****************************************/

 MOTORV.DATA = 0b01001001; //Akselerasjonen(rampen) til motoren
 _delay_ms(td2);
 MOTORV.DATA = 0b00000111;
 _delay_ms(td2);
  MOTORV.DATA = 0b00110001; //Reversed
 _delay_ms(td);
```

```
 MOTORV.DATA = 0x28;       //Clear position
 _delay_ms(td2);
 MOTORH.DATA = 0b01001001; //Akselerasjonen(rampen) til motoren
 _delay_ms(td2);
 MOTORH.DATA = 0b00000111;
 _delay_ms(td2);
 MOTORH.DATA = 0x28;       //Clear position
 _delay_ms(td);

delsvar=0;
delsvar2=0;
delsvar3=0;
delsvar4=0;

/***************************************
 *            HOVEDPROGRAMMET           *
 ***************************************/


 for(;;)
 {

Glemmefaktor(&delsvar,&delsvar2,&delsvar3,&delsvar4,samples,&ResultVolt,&Re
sultVolt2,&Brems);

  Gjsnitt(ResultVolt,ResultVolt2,&ResultFrem,Brems);

  // Deakselerasjon(ForRes,ResultFrem);

  Dodband(ForRes,ResultFrem);

  Drift(ResultFrem);

  Tomgang(ResultFrem);

  ForRes=ResultFrem;

 }
}


/***************************************
 *                Pekere                *
 ***************************************/


void Deakselerasjon(int ForResP,int ResultFremP)
{
  while((ForResP-ResultFremP)>2)
  {
   LEDPORT.DIR = 0xF;              //Setter farten
   MOTORV.DATA = 0b01000001;
   MOTORV.DATA = 0b00000000;
   MOTORV.DATA = ForResP;
   _delay_ms(2);

   MOTORH.DATA = 0b01000001;    //Setter farten
   MOTORH.DATA = 0b00000000;
   MOTORH.DATA = ForResP;
   _delay_ms(2);
   ForResP--;
```

```
  }
}

void Dodband(int ForResP,int ResultFremP)
{
  if((ForResP-ResultFremP)<3)
  {
  LEDPORT.DIR=0x00;
  MOTORV.DATA = 0b00100001;  //Får rullatoren til å flytte seg
  _delay_ms(td2);
  MOTORV.DATA = 0b00000000;
  MOTORV.DATA = 0b00000010;
  _delay_ms(td);

  MOTORV.DATA = 0b01000001;  //Setter fart
  _delay_ms(td2);
  MOTORV.DATA = 0b00000000;
  MOTORV.DATA = ForResP;
  _delay_ms(td);

  MOTORH.DATA = 0b00100001;  //Får rullatoren til å flytte seg
  _delay_ms(td2);
  MOTORH.DATA = 0b00000000;
  MOTORH.DATA = 0b00000010;
  _delay_ms(td);


  MOTORH.DATA = 0b01000001;  //Setter fart
  _delay_ms(td2);
  MOTORH.DATA = 0b00000000;
  MOTORH.DATA = ForResP;
  _delay_ms(td);
  }
  else if (-3<(ForResP-ResultFremP))
  {
  LEDPORT.DIR=0x00;
  MOTORV.DATA = 0b00100001;  //Får rullatoren til å flytte seg
  _delay_ms(td2);
  MOTORV.DATA = 0b00000000;
  MOTORV.DATA = 0b00000010;
  _delay_ms(td);

  MOTORV.DATA = 0b01000001;  //Setter fart
  _delay_ms(td2);
  MOTORV.DATA = 0b00000000;
  MOTORV.DATA = ForResP;
  _delay_ms(td);

  MOTORH.DATA = 0b00100001;  //Får rullatoren til å flytte seg
  _delay_ms(td2);
  MOTORH.DATA = 0b00000000;
  MOTORH.DATA = 0b00000010;
  _delay_ms(td);


  MOTORH.DATA = 0b01000001;  //Setter fart
  _delay_ms(td2);
  MOTORH.DATA = 0b00000000;
  MOTORH.DATA = ForResP;
  _delay_ms(td);
  }
```

```
}

void Drift(int ResultFremP)
{
  if(ResultFrem>4)
  {
  LEDPORT.DIR=0xFF;
  MOTORV.DATA = 0b00100001;   //Får rullatoren til å flytte seg
  _delay_ms(td2);
  MOTORV.DATA = 0b00000000;
  MOTORV.DATA = 0b00000010;
  _delay_ms(td);

  MOTORV.DATA = 0b01000001;   //Setter fart
  _delay_ms(td2);
  MOTORV.DATA = 0b00000000;
  MOTORV.DATA = ResultFremP;
  _delay_ms(td);

  MOTORH.DATA = 0b00100001;   //Får rullatoren til å flytte seg
  _delay_ms(td2);
  MOTORH.DATA = 0b00000000;
  MOTORH.DATA = 0b00000010;
  _delay_ms(td);


  MOTORH.DATA = 0b01000001;   //Setter fart
  _delay_ms(td2);
  MOTORH.DATA = 0b00000000;
  MOTORH.DATA = ResultFremP;
  _delay_ms(td);
  }
}

void Tomgang(int ResultFremP)
{
 if(ResultFremP<=4)
 {
 LEDPORT.DIR = 0x00;
 ResultFremP=0;

 MOTORV.DATA = 0x28;               //Clear position
 _delay_ms(td2);
 MOTORV.DATA = 0b01000001;         //Setter fart
 _delay_ms(td2);
 MOTORV.DATA = 0b00000000;
 MOTORV.DATA = ResultFremP;
 _delay_ms(td);

 MOTORH.DATA = 0x28;               //Clear position
 _delay_ms(td2);
 MOTORH.DATA = 0b01000001;         //Setter fart
 _delay_ms(td2);
 MOTORH.DATA = 0b00000000;
 MOTORH.DATA = ResultFremP;
 _delay_ms(td);
 }
}
```

```
void Glemmefaktor(int *delsvarP,int *delsvar2P,int *delsvar3P, int
*delsvar4P, int samplesP,int *ResultVoltP,int *ResultVolt2P,int *Brems)
{



  ADCA.CH0.CTRL |= ADC_CH_START_bm;                         //høyre FSR
  *delsvarP = 0.3*(ADCA.CH0RES*0.204)+0.7*(*delsvarP);
  ADCA.CH1.CTRL |= ADC_CH_START_bm;                         //venstre FSR
  *delsvar2P = 0.3*(ADCA.CH1RES*0.204)+0.7*(*delsvar2P);
  ADCA.CH2.CTRL |= ADC_CH_START_bm;                         //høyre brems
  *delsvar3P = 0.1*(ADCA.CH2RES-105)*0.9+0.9*(*delsvar3P);
  ADCA.CH3.CTRL |= ADC_CH_START_bm;                         //venstre brems
  *delsvar4P = 0.1*(ADCA.CH3RES-100)*0.8+0.9*(*delsvar4P);

 *ResultVoltP=*delsvarP;
 *ResultVolt2P=*delsvar2P;
 *Brems=(*delsvar3P)+(*delsvar4P);
}


void Gjsnitt(int ResultVoltP,int ResultVolt2P,int *ResultFremP,int Brems)
{
      *ResultFremP=(((ResultVoltP+ResultVolt2P)/2)+0.5)-((Brems/2)+0.5);
      if(*ResultFremP<0)
      *ResultFremP=0;
      if(*ResultFremP>30)
      *ResultFremP=30;
}
```

Vedlegg B: Programkode for Arduni UNO-loggekort

```
int firstSensor = 0; // first analog sensor
int secondSensor = 0; // second analog sensor
int inByte = 0; // incoming serial byte
void setup()
{
// start serial port at 9600 bps:
Serial.begin(9600);
pinMode(2, INPUT); // digital sensor is on digital pin 2
pinMode(1, INPUT);
establishContact(); // send a byte to establish contact until receiver
responds
}
void loop()
{
// if we get a valid byte, read analog ins:
if (Serial.available() > 0)
  {
  // get incoming byte:
  // inByte = Serial.read();
  // read first analog input, divide by 4 to make the range 0-255:
  firstSensor = analogRead(1);
  // delay 10ms to let the ADC recover:
  delay(10);
  // read second analog input, divide by 4 to make the range 0-255:
  secondSensor = analogRead(0);
  // send sensor values:
  Serial.print(firstSensor);
  Serial.print('\t');
  Serial.print(secondSensor);
  Serial.println();
  delay(190);
  }
}
void establishContact() {
while (Serial.available() <= 0
  {
  Serial.print('A'); // send a capital A
  delay(300);
  }
}
```

Vedlegg C: Prisliste

Reverse Lock Wheel - Set for a pair Powerslide Nordic Skates kr. 713 NOK (94.90 EUR)

Referanseliste

[i] Nasjonalt folkehelseinstitutt (2011), «Eldre - andelen eldre over 65 år i befolkningen». Tilgjengelig fra: http://www.fhi.no/eway/default.aspx?pid=233&trg=MainLeft_6039&MainArea_5661=6039:0:15,4576:1:0:0:::0:0&MainLeft_6039=6041:70828::1:6043:7:::0:0, lastet ned 4.6.2012

[ii] Svein Tønseth, (2009), «Dramatisk press på framtidas sykehus». Tilgjengelig fra http://www.sintef.no/Presserom/Forskningsaktuelt/Dramatisk-press-pa-framtidas-sykehus/ , lastet ned 19.5.2012

[iii] Nemoto, Y. (1998), POWER-ASSISTED WALKING SUPPORT SYSTEM FOR ELDERLY. Tilgjengelig fra http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=745229, lastet ned 13.2.12.

[iv] Jaime Valls Mir ´o, Vivien Osswald, Mitesh Patel and Gamini Dissanayake (2007), *Robotic Assistance with Attitude: a Mobility Agent for Motor Function Rehabilitation and Ambulation Support*, The Faculty of Engineering and IT, University of Technology Sydney (UTS), NSW, Australia. Tilgjengelig fra http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5209523, lastet ned 13.2.12.

[v] Wikipedia (2011), *Partially observable Markov decision process.* Tilgjengelig fra: http://en.wikipedia.org/wiki/Partially_observable_Markov_decision_process, lastet ned 15.2.2012

[vi] Fei Shi (2010), *Based On Force Sensing-Controlled Human-Machine Interaction System For Walking Assistant Robot*, Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. Tilgjengelig fra: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5554167, lastet ned 14.2.2012.

[vii] Sabati, Angelo M., Vincenzo Genovese, Elena Pacchierotti (2002), *A Mobility Aid for the Support to Walking and Object Transportation of People with Motor Impairments*, Scuola Superiore Sant 'Anna, Pisa, Italy. Tilgjengelig fra: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1043942&tag=1, lastet ned 16.2.2012.

[viii] Dubowsky Steven (2000), *PAMM - A Robotic Aid to the Elderly for Mobility Assistance and Monitoring: A "Helping-Hand" for the Elderly*, Massachusetts Institute of Technology (MIT). Tilgjengelig fra: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5209523, lastet ned 17.2.2012.

[ix] Pedelecs (2006) - the electric bike community. Tilgjengelig fra: http://www.pedelecs.co.uk/forum/electric-bicycles/1172-home-made-pedelec-pedal-torque-sensor-2.html (lastet ned 28.9.2011)

[x] Expresso (2012) «Touch2move». Tilgjengelig fra: http://www.expresso.de/media/_laender/pdf/en-gb/product-catalogs/touch2move_ingenious_transport_solutions.pdf, lastet ned 4.6.2012.

[xi] OMG (2012) «UML® Resource Page». Tilgjengelig fra: http://www.uml.org/, lastet ned 4.6.2012.

[xii] Altitec (2011) «GELE/AGM/VRLA». Tilgjengelig fra: http://altitec.no/12v-22ah-hoy-strom-181x77x167-m5.html, lastet ned 8.5.2012.

[xiii] Wikipedia (2012) «Universal asynchronous receiver/transmitter» Tilgjengelig fra: http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter , lastet ned 30.5.2012.

[xiv] Atmel (2012) «AVR JTAGICE mkII». Tilgjengelig fra: http://www.atmel.com/tools/AVRJTAGICEMKII.aspx, lastet ned 4.6.2012.

[xv] Atmel (2011) «Key parameters for ATxmega128A1». Tilgjengelig fra: http://www.atmel.com/devices/ATXMEGA128A1.aspx, lastet ned 30.5.2012.

[xvi] Arduino UNO (2012) http://arduino.cc/en/Main/ArduinoBoardUno, lastet ned 4.6.2012.

[xvii] Skarra, Sigurd (2011) NTNU, ”Nye sensorer for en mer brukervennlig rullator”.

[xviii] Alibaba (2012) Unidirectional bearings. Tilgjengelig fra: http://www.alibaba.com/product-gs/435161726/unidirection_bearing_629_2rs.html, lastet ned 23.2.2012.

[xix] X-Skating (2006), Reverse Lock Wheel – Set for a pair Powerslide Nordic Skates. Tilgjengelig fra: http://www.xskating.com/Reverse-Lock-Wheel-Set-for-a-pair-Powerslide-Nordic-Skates::421.html, lastet ned 12.4.2012.

[xx] Austad, Semb, Flatgård, Nordkvelde (2011) HiST, «Modernisert rullator».

[xxi] Interlink Electronics (2010) Force Sensing Resistors. Tilgjengelig fra: http://www.digikey.com/Web%20Export/Supplier%20Content/InterlinkElectronics_1027/PDF/Interlink_Electronics_Integration_Guide.pdf?redirected=1, lastet ned 1.6.2012