



Smarthjem - Et strømstyringssystem

Maja Charlotte Otnes

Carl Joachim Rørvik

Master i datateknologi

Innlevert: mai 2014

Hovedveileder: Tor Engebret Onshus, ITK

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

SmartHjem - Et strømstyringsystem

Maja Charlotte Otnes & Carl Joachim Rørvik

Masteroppgave



NTNU – Trondheim
Norwegian University of
Science and Technology

21. mai 2014

Til minne om

Ellen og Oivind Rørvik

Oppgavebeskrivelse

Denne oppgaven skal se nærmere på hvordan nye strømstyringssystemer kan bidra til utviklingen av smartere hjem. Det skal utvikles et forslag til et system som skal kunne ta smarte og kostnadseffektive valg for forbrukeren, samt gjøre forbrukerne mer bevisste over sitt eget strømforbruk.

Det skal undersøkes om oppvarmingen i boligen kan reguleres med en MPC-regulator, og om dette vil kunne bidra til å redusere strømkostnadene tilknyttet til oppvarming, samtidig som brukerens krav til komfort blir overholdt.

Et nettbasert brukergrensesnitt til systemet skal utvikles for at forbrukeren skal kunne sette innstillinger, men også for å gjøre informasjonen om forbruk og kostnad lettere tilgjengelig. Brukeren skal kunne se kostnadsestimater for ulike brukerinnstillinger knyttet til oppvarming. I tillegg skal brukeren kunne se og styre diverse andre elektriske artikler i boligen.

Oppsummering og konklusjon

I denne masteroppgaven har det blitt sett på strømstyringssystemer i boliger, og på hvordan fremtidens smarte hjem vil kunne endre måten vi bruker strøm på i det daglige. Det har blitt laget et forslag til et strømstyringssystem, SmartHjem, som illustrerer noen av mulighetene et slikt system kan gi. I brukergrensesnittet til SmartHjem kan brukere styre elektriske artikler, se historikk over forbruk, sette innstillinger for oppvarming og estimere fremtidige strømkostnader.

SmartHjem kommuniserer med reguleringsystemet for oppvarmingen i boligen. Dette styresystemet er en MPC-regulator som varmer opp boligen på en kostnadseffektiv måte, samtidig som den overholder temperaturkravene til brukeren. MPC-regulatoren tar hensyn til strømprisene, ute-temperaturene, brukerens tilstedeværelse og krav til komforttemperatur.

Det har vist seg at MPC-regulatoren gjør det mulig å ta smartere avgjørelser ved styring av oppvarmingen i hjemmet. SmartHjem er et forslag til hvordan et smart hjem system kan presenteres og anvendes av forbrukerne. Hensikten med systemet er å gi brukeren mulighet å styre elektriske artikler i hjemmet over internett, få god oversikt over strømforbruk og kostnader, samt oppmuntre til å spare strøm og penger.

Resultatene i denne oppgaven viser at det er mulig å bruke strøm smartere, og at det kan bidra til å redusere strømutgiftene. Ved å gjøre informasjon om strømforbruket lettere tilgjengelig gjennom et brukergrensesnitt på nett vil brukerne kunne bli inspirert til et mer bevisst strømforbruk.

Koden til dette systemet kan lastes ned fra GitHub via linken <https://github.com/cjrorvik/smarthjem>. En video som introduserer websystemet kan sees på YouTube via <http://youtu.be/uROMFZXKkI>.

Abstract

This master thesis looks closer into controlling power systems in people's homes, and how smart homes in the future will change the way we use electricity in the daily life. It has been made a suggestion to a power control system, SmartHjem, which illustrates some of the opportunities this kind of a system can give. In the interface of SmartHjem, the user can control electrical articles, look at historical consumption data, activate new heating settings and estimate the related future cost.

SmartHjem communicates with the regulator system for heating control in the home. This control system is a Model Predictive Controller (MPC) that heats up the home in a cost effective way, and at the same time complies with the user requirements. The MPC-regulator takes into account, the hourly electricity prices, the outdoor temperatures, if the user is home or not, and user comfort requirements. It has been shown that the MPC makes it possible to make smarter decisions about the management of heating in the home. SmartHjem is a suggestion of how a smart home system can be presented and used by the consumers. The purpose of this system is both to allow the user to control electrical appliances in the home over the Internet, get a good overview of energy consumption and costs, and to encourage saving energy and money.

The results in this thesis shows that it is possible to use the electricity in people's home in a smarter way, which will contribute in a reduction of the electricity expenses. And by making the information about the power consumption more available through Internet, users will be inspired to a more deliberate consumption.

This system's code can be found at GitHub on <https://github.com/cjrorvik/smarthjem>. And a video introducing the web system can be watched at YouTube at <http://youtu.be/uROMFZXKmkI>.

Forord

TTK4900 Masteroppgave er en del av mastergraden ved Norges Teknisk-Naturvitenskaplige Universitet (NTNU) i Trondheim. Denne oppgaven ble skrevet i perioden januar til mai 2014 av Maja Charlotte Otnes fra Teknisk Kybernetikk og Carl Joachim Rørvik fra Datateknikk.

Anerkjennelser

Først og fremst vil vi gjerne takke vår veileder Tor Onshus for å ha gitt oss muligheten til å velge denne oppgaven, Nord Pool Spot for tilgang og bruk av deres strømprisdata- og statistikk, og Bjarne Foss for veiledning og hjelp til reguleringsdelen av oppgaven.

Vi vil også takke de som hjalp oss underveis i oppgaven med tilbakemeldinger, og til slutt Trondheim og NTNU for en flott studietid!

Trondheim, 21. mai 2014

Maja Charlotte Otnes

Carl Joachim Rørvik

Innledning

Motivasjon

Strøm og elektrisitet er noe vi alle er avhengige av døgnet rundt. Oppvarming, elektriske artikler, digitale enheter, lading av batterier og belysning er blitt en stor og naturlig del av hverdagen. En tenker ikke lenger over hvordan alt henger sammen, men det forventes at det til en hver tid fungerer.

Fra tidlig alder læres vi opp til at en må huske på å ikke sløse med strømmen, og at en gjerne skal fokusere mer på å spare strøm. Samtidig er realiteten slik at strømmen leses av en gang i kvartalet, og en regning dukker opp i postkassen. Det blir opp til en selv å tenke over hvordan strømforbruket har vært, og hva som førte til nettopp denne månedens strømutfgifter.

Vi lever i dag i en digital hverdag der mye er avhengig av at vi har tilgang på strøm. Vi ønsket derfor å se nærmere på hvordan vi kan benytte de digitale mulighetene rundt oss til å bli mer bevvist på eget strømforbruk. Hvis det er mulig å styre elektriske artikler i hjemmet over nett, få automatiske sparetips eller nyttig statistikk, så må det da vel bli lettere å være oppmerksom på det vi lærte om sparetips som unge? Dersom det i tillegg er mulig å bruke teknologi til å hjelpe forbrukeren med å bruke strøm smartere vil resultatet forhåpentligvis bli reduserte strømkostnader. Med dette ønsker vi å bidra til at strøm og elektrisitet også er med på den digitale utviklingen fremover.

Leserens guide

Denne oppgaven er delt inn i fire hoveddeler som hver inneholder flere kapitler.

Den første delen er et forstudie, der kapittel 1 tar for seg litt historikk om strøm i Norge, og kapittel 2 som ser nærmere på eksisterende løsninger av smarte hjem. Forstudiet er med på å bygge opp grunnlaget for produktet i denne oppgaven, samt gi bakgrunnsinformasjon om temaet.

Del 2 består av en beskrivelse av det nye smart hjem systemet, hvilke krav som bør oppfylles og arkitekturen som er lagt til grunn.

I del 3 forklares det nærmere om modelleringen og MPC- regulatoren som styrer oppvarmingen i smart hjem systemet. Her forklares det grunnleggende teori, antagelser som har blitt gjort og ulike måter å løse optimaliseringsproblemene på.

Del 4 består av oppgavens avslutning. Den er satt sammen av resultatene fra det nye smart hjem systemet, et diskusjonskapittel og en konklusjonsdel, samt forslag til videre arbeidsoppgaver for systemet. Til slutt i oppgaven finnes kilder og diverse vedlegg.

En CD medfølger også denne rapporten. Her ligger kildekoden til prosjektet, introduksjonsfilm av systemet, utvalgte referanse kilder og ulike vedlegg.

Innhold

Oppgavebeskrivelse	i
Oppsummering og konklusjon	ii
Abstract	iii
Forord	iv
Innledning	v
Figurer	xii
I Forstudie	1
1 Strøm i Norge	3
1.1 Historie om strøm i Norge	3
1.2 Strømproduksjon i Norge	4
1.3 Strømpriser	5
1.4 Strømforbruk i norske husstander	7
1.5 Fremtidens strømforvaltning	8
2 Smarte hjem	11
2.1 Tilbydere av smarte hjem løsninger	12

2.2	Sammenligning av tilbydere	15
II	SmartHjem	17
3	Beskrivelse	19
3.1	Formål	19
3.2	Brukergrensesnitt	20
3.3	Regulering av oppvarming	21
3.4	Aktuelle datakilder	21
4	Krav	23
4.1	Funksjonelle krav	23
4.2	Kvalitetskrav	25
5	Arkitektur	27
5.1	Teknologier	28
5.2	Websystemet	34
5.3	Kommunikasjon i systemet	42
III	Regulering	47
6	Modellering	49
6.1	Modelleringsteknikker	50
6.2	Teori	51
6.3	Termisk modell av en bolig	53
6.4	Modell av oppvarmingssystem	54
6.5	Antagelser og forenklinger	55
7	MPC	61
7.1	Optimaliseringsproblemet	63
7.2	Kvadratisk programmering	72
7.3	Blandet heltall programmering	73

7.4	Beregningstid	75
IV	Avslutning	79
8	Resultater	81
8.1	Mitt hjem	81
8.2	Oppvarming i SmartHjem	84
8.3	MATLAB resultater	85
9	Diskusjon	95
9.1	Oppvarmingregulering med MPC	96
9.2	Redusering av beregningstiden	98
9.3	Livet i et smart hjem	100
9.4	Fremtiden og miljøet	101
9.5	Hvem vil tjene på smartere hjem?	102
9.6	Systemet i praksis	102
10	Konklusjon	105
11	Videre arbeid	107
11.1	SmartHjem websystemet	107
11.2	Oppvarmingsalternativer	109
11.3	Smartere varmtvannsbereder	109
11.4	Termisk modell av boligen	110
11.5	Optimaliseringsproblemet	111
	Referanser	113
	Vedlegg	119

Figurer

1.1	Strømpris(NOK/MWt). 21.-25. april 2014, Trondheim . . .	6
1.2	Fordeling av elektrisitetsforbruk for norske husholdninger. .	7
1.3	Illustrasjon av Smart Grid	9
5.1	Oppsett av Socket.IO mellom klient og server	33
5.2	InteractiveHomeModel.js	35
5.3	HeatingModel.js	35
5.4	EldeviceInfoBoxModel.js	36
5.5	CurrentUsageModel.js	36
5.6	HeatingView.js	38
5.7	EldeviceInfoBoxView.js	38
5.8	CurrentUsageGraphView.js	39
5.9	Utils	40
5.10	Overordnet klassesdiagram av websystemet	41
5.11	Kommunikasjonsprotokoller i systemet	42
5.12	Meldingsflyt i systemet	43
6.1	Ulike modelleringsteknikker	51
6.2	Husmodell	53
6.3	Minstekrav til termisk komfort i et rom, gitt ved overflate- temperatur, romtemperatur, lufthastighet og temperaturfor- skjeller	56

6.4	Illustrasjon av varmeoverføringsprosesser	58
7.1	Illustrasjon av MPC prinsippet	62
7.2	Oppbygging og bruk av optimaliseringen	72
7.3	Illustrasjon av Branch and Bound-algoritmen	75
7.4	Illustrasjon av inngangsblokkering	78
8.1	Smart Hjem-huset	82
8.2	Lampe i Smart Hjem	83
8.3	Forbruk av lampe i Smart Hjem	83
8.4	Total forbruk og kostnad i Smart Hjem	84
8.5	Referanserresultat	86
8.6	Vekting av strømkostnad	88
8.7	Vekting av temperaturendring	89
8.8	Inngangsblokkering	90
8.9	Temperaturforløp med og uten integralvirkning	92
8.10	Termostat	93
8.11	Blandet heltall programmering	94
9.1	Smart Hjem i praksis	103

Del I

Forstudie

Kapittel 1

Strøm i Norge

1.1 Historie om strøm i Norge

Mot slutten av 1800-tallet begynte produksjonen av strøm ved bruk av vannkraftverk i Norge. Før den tid hadde spinnerier, verksteder og fabrikker vært i nærheten av de større elvene for å dra nytte av kraften i det rennende vannet, men elektrisk energi var ennå ikke introdusert. Etter hvert som kunnskapen om vannkraftverk og vekselstrøm tok seg opp, ble det mulig å produsere strøm et sted i landet, og så frakte denne videre rundt. Ved starten av 1900-tallet var det flere norske byer som hadde fått installert elektrisk gatelys. Norsk Hydro ble en av de store aktørene innenfor norsk kraftproduksjon, og i 1911 sto, på den tid, verdens største kraftstasjon på Vemork ferdig.

Før krigen var det hovedsaklig privat industri som sto for det meste av kraftproduksjonen i landet, og staten hadde kun en eierandel på rundt 10%. Etter krigen tok staten mer ansvar for å etablere et felles norsk statnettverk, og i år 2000 var eierandelen på rundt 30% [1]. Fram til midten av 1960-tallet var det et stort fokus på å bygge store og effektive vannkraftverk, og det

ble lagt stor vekt på den økonomiske gevinsten og sysselsettingen dette ga. Etter hvert ble det også tatt mer hensyn til natur og miljø i områdene rundt utbyggingen av de store kraftverkene.

I 1990 ble energiloven vedtatt. Hovedmål med denne loven er å sørge for at det er fritt marked i Norge for kjøp og salg av elektrisk energi [2]. Dette gir en energiutbygging som er mindre styrt av politiske vedtak og prognoser. I loven gjelder det leveringsplikt på energi, men kundene kan selv velge hvem de vil ha som leverandør. I tillegg er det også åpent for import og eksport av energi til og fra andre land. Nord Pool er en felles nordisk kraftbørs for kjøp og salg av elektrisk energi, og prisene her varierer fra time til time hele døgnet.

1.2 Strømproduksjon i Norge

Vannkraft har vært, og er fremdeles, helt klart den største bidragsyteren til norsk kraftproduksjon. Opp mot 97% av all norsk kraftproduksjon kommer fra vannkraft, mens det resterende kommer fra vindkraft, fossilt eller biobrennsel. De siste 10 årene har produksjonen ligget på omlag 127 Twt/år [3, 4]. Likevel er ikke produksjonen nok til å dekke Norges forbruk av strøm, og vi må derfor importere strøm fra andre land. Strøm er en ferskvare, og må derfor brukes i det den produseres. Norge eksporterer av den grunn strøm når tilsiget er stort, gjerne om våren som følge av snøsmelting i fjellene, og om høsten når det regner mye.

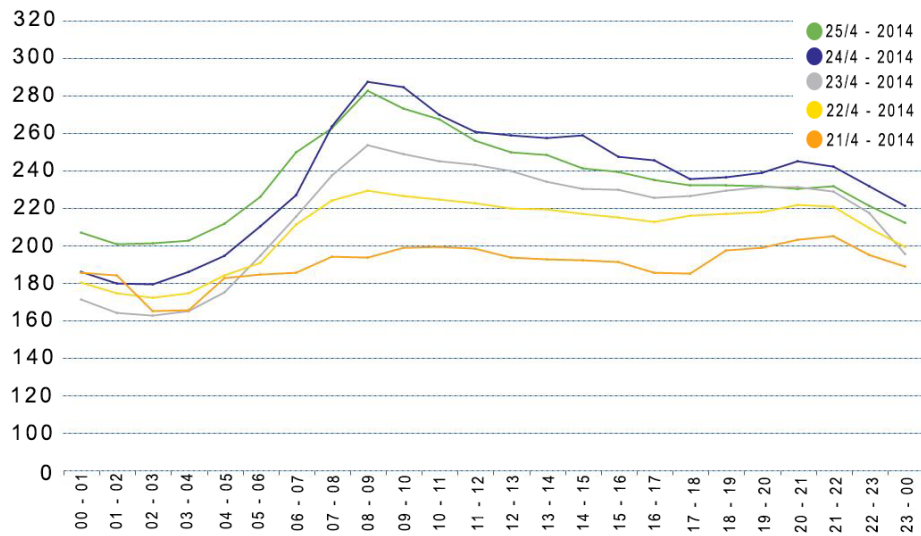
Det er Norges vassdrags- og energidirektorat (NVE) som har ansvar for å sikre en hensynsfull forvaltning av vassdragene i Norge, samt tilrettelegge for effektiv kraftomsetning og energibruk. De setter også regler for hvor store vannmengder det er lov til å oppbevare i vannmagasinene. Dette for å forhindre naturkritiske situasjoner, samt sørge for at det opprettholdes orden på kraftproduksjonen.

1.3 Strømpriser

Utgangspunktet for strømprisene norske huster betaler for i dag er kursen på strømbørsen Nord Pool. I tillegg kommer et påslag som leverandøren legger på for å sikre fortjeneste. Dette påslaget avhenger av både selskap og privat avtale mellom forbruker og selskap.

Det finnes mange ulike strømleverandører som har ulike tilbud. Med spotpris kan kundene få strøm til den samme prisen som leverandørene betaler mot et påslag. Prisene per kWt vil dermed variere avhengig av den reelle markedsværdien på strøm fra Nord Pool. Alternativet til spotpris er å velge en fastprisavtale med strømleverandøren. I tillegg til å betale per kWt som brukes, er nettleie en utgift alle må betale for å få strømmen levert til husstand. I Norge er det ca. 140 ulike nettleieselskaper som driver hver sin del av strømmettet [5].

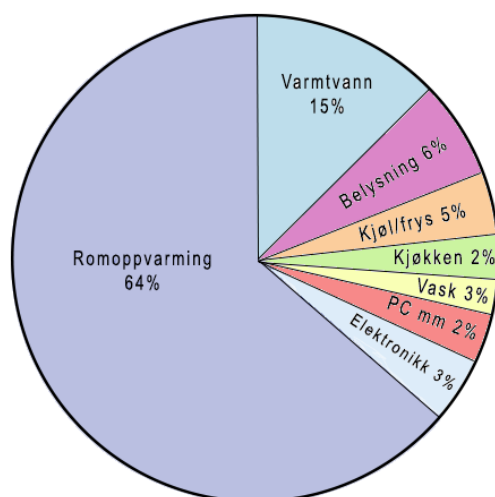
Strømprisene som settes er avhengig av tilbud og etterspørsel. Dersom det har vært mye nedbør i Norge over tid kan dette føre til lavere strømpriser da det blir nødvendig å produsere strøm for at ikke vassdragene skal bli overfylte. Motsatt kan lite nedbør og kulde føre til et større forbruk og lavere produksjon, som så vil øke prisen. På tidspunkt der folk flest har et lavt strømforbruk, for eksempel på natten, vil det gjerne også gi lavere strømpriser, se figur 1.1 [6].



Figur 1.1: Strømpris(NOK/MWt). 21.-25. april 2014, Trondheim

1.4 Strømforbruk i norske husstander

I norske husstander er det hovedsaklig oppvarming som bidrar til den største andelen av strømregningen. Dette gjelder både romoppvarming og varmtvannsoppvarming, som regnes å være mellom 70-75% av forbruket [7]. De resterende kostnadene er knyttet til annet bruk, som belysning, kjøkken, elektronikk og lignende, se figur 1.2.



Figur 1.2: Fordeling av elektrisitetsforbruk for norske husholdninger.

Det har vist seg at det meste av forbruket i vanlige norske husholdninger gjerne er om morgenen og ettermiddagene da folk som regel er hjemme i husene sine. Det er da folk flest gjerne ønsker et varmt hjem, bruker varmtvann, lager mat og skrur på lys. Det er naturlig at etterspørselen av strøm øker i disse periodene.

1.5 Fremtidens strømforvaltning

AMS

Innen 1. januar 2019 skal alle norske husstander ha tatt i bruk Avanserte Måle- og Styringssystem (AMS). Dette er målere som hver time sender informasjon om forbruket til nettselskapet, og gir dermed et mer detaljert bilde av strømforbruket som danner grunnlag for strømregningen. Dette gjør at det blir mulig å lage en rekke nye tjenester som kommer forbrukerne tilgode med tanke på styring av strømforbruket.

Smartere målere vil også trolig føre til et mer fleksibelt strømforbruk, noe som vil gi bedre utnyttelse av kraftsystemet. I tillegg vil et mer effektivt strømforbruk gi positive innvirkninger på miljøet, da det kan bli et mindre behov for å benytte seg av fossile energikilder [8].

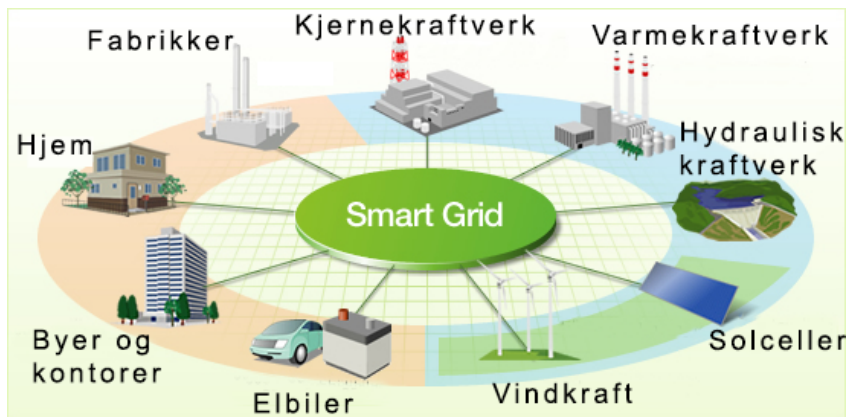
For kraftselskapene blir det mange muligheter til å lage nye produkter og tjenester i årene fremover. Norges vassdrags- og energidirektorat, som har ansvaret for gjennomføringen av AMS prosjektet, legger ikke opp til noen krav om valg av teknologier, så der blir det opp til kraftselskapene selv å sørge for at ting fungerer som det skal.

Smart Grid

Som det har blitt nevnt tidligere er historien om elektrisk strøm i Norge over 100 år gammel. Strømnettet ble laget på en tid der det var betydelig færre husstander, og det fantes også langt færre elektriske artikler. For at dagens strømnett ikke skal overbelastes ved tider der det er behov for et høyt forbruk, har man i stor grad begynt å tenke fremover. Arbeidet mot neste generasjons strømnett, Smart Grid, er allerede i gang [9, 10].

Smart Grid er en samlebetegnelse for bedre utnyttelse av energiinfrastrukturen, integrasjon av IKT i alle ledd i det elektriske kraftnettet, og mer energifokuserte sluttbrukere. Som en del av Smart Grid er utrulling av AMS, nevnt i forrige seksjon.

Før i tiden ble strømmen hovedsaklig produsert av noen få store kraftstasjoner. Etter hvert som man har fått mer kunnskap om energiproduksjon har det blitt mer aktuelt med en desentralisering av strømproduksjonen, til fordel for produksjon av strøm ved mindre kraftstasjoner som vindkraft, bølgekraft og solceller. For å inkludere dette i det totale strømnettverket kreves det en god organisering og planlegging. I tillegg vil det om noen år bli mer aktuelt med plusshus, et hus som produserer mer strøm enn det kan bruke selv, og som også vil kunne være med på å selge strøm på markedet.



Figur 1.3: Illustrasjon av Smart Grid

Kapittel 2

Smarte hjem

Smarte hjem kan defineres som et hjem der elektroniske apparater selv kan justere sine innstillinger for å best tilfredsstille forbrukerens behov, sikkerhet og økonomi. Apparatene kan også bli styrt av brukerne over internett, enten via mobiltelefon, nettbrett eller en nettside. Fra et forbrukerperspektiv kan et smart hjem lønne seg økonomisk, i tillegg til å gi brukeren mer oversikt over sitt eget forbruk.

Det er mange muligheter for produkter og tjenester som leverandører kan tilby innenfor smarte hjem. Mer intelligente systemer, abonnementer for å kunne tilby vekterutrykning ved innbrudd, tilkallelse av brannvesen ved brann, utvikling av nye nettløsninger for kontroll av enheter og visualisering av forbruk.

Per dags dato er det flere leverandør som tilbyr slike tjenester, men likevel er det ikke allemannseie. I dette kapittelet vil det bli sett nærmere på flere norske og internasjonale smarte hjem tilbydere og hva de kan levere til forbrukerne.

2.1 Tilbydere av smarte hjem løsninger

VIVA

Viva er en norsk leverandør av smarte hjem tjenester. Ved å benytte smartkontakter, sensorer og en bredbåndstilkoblet styringsenhet kan hjemmet bli smartere. Panelovner tilkoblet smart-



kontakter kan automatisk dag- og nattsenkes, og over tid vil ovnene lære seg hushandens rutiner. Lys slås automatisk av når huset er tomt, og bevegelsessensorer tar hensyn til bevegelse og mørke. Alt kan styres via en egen applikasjon til mobil. Prisen varierer basert på hvor mange smartkontakter og sensorer forbrukeren ønsker [11].

Smartly



Smartly er en del av Lyse-konsernet, og de arbeider med løsninger for smartere styring av varme, lys, innbrudds- og brannalarm. Automatiske målere og timeprising på strøm gjør det lettere å ha et billigere og bedre energiforbruk. Alt kan styres med en fjernkontroll i hjemmet, eller via nettbrett og smarttelefonen. Gjennom Enova støtter myndighetene kjøp av varmestyring fra Smartly, så det er mulig å få igjen opp til 35 prosent av investeringskostnadene [12].

Verisure

Verisure er det nye navnet på det som tidligere var Securitas Direct, og er et selskap som hovedsaklig har fokusert på alarm i hjemmet. Med Verisure har selskapet tatt en ny retning mot smartere hjem. Samtidig som de beholder alarmtjenesten, har de nå begynt å tilby smartkontakter, smarte brannalarmer, samt termometre. Alle enhetene kan konfigureres via smarttelefon og nettbrett. Selskapet har over 1,5 millioner kunder fordelt på 12 land, men en stor andel av kundene benytter kun alarmtjenestene [13].



Nest



Nest er et amerikansk selskap som tilbyr intelligente termostater og brannalarmer. Termostaten programmerer seg selv basert på når folk er hjemme, i tillegg til at den også kan styres via telefonen.

Selskapet reklamerer med at termostaten kan lære seg så mye at kostnaden knyttet til oppvarming og nedkjøling reduseres med 20%. Temperaturen justeres også slik at det blir tatt høyde for når etterspørselen etter strøm er stor og prisen høy, gjerne om morgenen og ettermiddagen. Samtidig tar termostaten hensyn til værprognoser for gjeldende posisjon, aktivitetssensorer for automatisk-borte-funksjon, luftfuktigheten i rommet og selvfølgelig temperaturen. Selskapet ble kjøpt opp av Google i januar 2014 [14].

Phillips Hue

Phillips Hue har spesialisert seg på smarte lyspærer som kan styres via mobiltelefon. Via en applikasjon tilkoblet internett kan lysene dimmes, skrues av eller på, samt endres til valgfri farge [15].



Belkin WeMo

Belkin WeMo er et sett med smartkontakter som kan styres over Wi-Fi og tilhørende WeMo-applikasjon. I tillegg til at brukeren kan skru av og på kontakten, blir også informasjon om enhetens strømforbruk sendt direkte til applikasjonen. Det kan også settes tidsplaner som angir når enheten skal være aktiv eller ikke [16].

Revolv

Revolv er et selskap som har spesialisert seg på å automatisk kontrollere elektroniske enheter i et smart hjem, i stor grad uavhengig av merke.



Enheter som Nest termostat, Phillips Hue, Belkin WeMo, Sonos Hi-Fi kan la seg styre av Revolv systemet [17].

2.2 Sammenligning av tilbydere

Som beskrevet over finnes det allerede en rekke tilbydere av smarte hjem løsninger. De ulike selskapene har hver sine områder de fokuserer på, men det handler hovedsakelig om å gjøre kundene tryggere, mer oppmerksomme og gi en større kontroll over sitt eget strømforbruk.

Ved å studere de ulike løsningene er det tydelig at det er mange muligheter for smart hjem systemer. Noen leverandører tilbyr tilsvarende løsninger, og noen har spesialisert seg innenfor noen områder. Tabell 2.1 oppsummerer og sammenligner de ulike smart hjem systemene som er presentert i dette kapitlet.

Selskap	Tilbyr	Etablering	Per måned	Antatt besparelse	Tilgjengelig
VIVA	Smartkontakter, sensorer, bredbåndstilkoblet styringsenhet. Intelligente ovner.	1 790 kr	79 kr	rundt 150 kr per måned	Norge
Smartly	Sensorer, termostat, røykvarslere, alarm, forbruksstyring. Koblet seg direkte på strømssystem i hjemmet, som en del av AMS.	16 970 kr	447 kr	15 - 20 % av oppvarming årlig	Norge
Verisure	Smartkontakter, røykvarslere, innbruddsalarm.	2 790 kr	429 kr	-	Verden
Nest	Intelligent termostat, røyk- og brannvarsler. Værbasert. Automatisk justering dersom ikke hjemme.	1 500 kr	-	10 - 20% mindre oppvarming og avkjøling	USA
Phillips Hue	Digitale lys i forskjellige farger. Timerbasert	1 490 kr	-	-	Verden
Belkin WeMo	Diverse smartkontakter. Forbrukskontroll via app	360 kr	-	-	Verden
Revolv	Sammenkobling av smartenheter til et brukersystem	1 800 kr	-	-	Verden

Tabell 2.1: Sammenligning av smarte hjem tilbydere

Del II

SmartHjem

Kapittel 3

Beskrivelse

I denne oppgaven har det blitt utviklet et forslag til en ny smart hjem løsning, og dette systemet har fått navnet SmartHjem. Dette kapittelet vil beskrive hva som er formålet med å utvikle SmartHjem, samt se på noen elementer som er viktige å ha med for å gjøre systemet mest mulig reelt.

3.1 Formål

I forstudiet har ulike løsninger av smarte hjem systemer blitt sammenlignet, og dette har gitt et godt utgangspunkt for hva et nytt system bør inneholde. I tillegg har kunnskap om hvordan produksjonen, distribusjonen og salg av strøm i Norge bidratt til tanker om mulighetene for et slikt system. Formålet med produktet er å gjøre allmennheten mer bevisst på sitt eget strømforbruk, og bidra med å redusere strømforbruket.

Kun en av smart hjem løsningene som har blitt presentert i forstudiet ble levert av et kraftselskap. Så det er nærliggende å tenke at flere kraftselskap bør kunne tjene på å levere tilsvarende løsninger for smarte hjem. Det vil

være en fordel fordi de da kan tilby sine kunder lavere strømutgifter, noe som vil gjøre deres kunder mer fornøyde. Dersom systemet utnytter seg av strømmen når den er billigere og et lavere forbruk når den er dyrere, vil det kunne bidra til å jevne ut forbruket noe i løpet av dagen. Et jevnere forbruk er ønskelig for kraftselskapene, da de må ha nok kapasitet til å håndtere det høyeste forbruket.

Det er et fåtall av de smarte hjem tilbyderene som legger opp til en tjeneste der målet er å gjøre brukerne mer oppmerksomme på hva ulike elektriske artikler faktisk bruker av strøm. Det at forbrukeren har lett tilgang på informasjon om eget forbruk er et viktig element for å gjøre det mulig å redusere strømutgiftene merkbart.

Det ble sett nærmere på hvilke elementer som ville gjøre det lettere for folk å styre strømtilkoblede enheter, men også kanskje det viktigste, nemlig hvilken kostnad som var tilknyttet til hvert produkt. Kostnader til oppvarming av hjem utgjør den største delen av strømregningen hos de fleste. Det ble derfor viktig å gjøre styringen og kostnadsoversikten av dette til en viktig del av systemet.

3.2 Brukergrensesnitt

I et brukergrensesnitt er det viktig å gi forbrukeren mulighet til å styre enheter, bli informert om feil, få nye oppdateringer og simulere forskjellige innstillinger. I tillegg bør det fremgå av en logg i systemet hva strømmen er brukt på, og når dette skjedde. Dette vil være kontrollrommet av systemet og til elektriske artikler i huset, i tillegg til å være det stedet hvor brukeren får informasjon om sitt eget forbruk.

Et slik system ville måtte være tilgjengelig til en hver tid, enten fra mobil, nettbrett eller en datamaskin. Det burde kunne være mulig å logge seg

på systemet fra andre siden av jorden og for eksempel skru av lyset eller senke temperaturen.

3.3 Regulering av oppvarming

Oppvarmingen av en bolig utgjør den største andelen av strømutgiftene, og en smart regulering av oppvarmingen har potensiale til å redusere disse utgiftene. Temperaturen i en bolig kan variere noe uten at det går ut over komforten, så en regulator kan ha noe frihet til å ta valg som vil redusere utgiftene. Brukeren bør kunne sette kravene til oppvarmingen gjennom brukergrensesnittet, som for eksempel temperaturkrav når brukeren er hjemme og borte fra huset. Systemet må kunne lære seg huset å kjenne, slik at systemet blir best mulig tilpasset hver enkelt husstand.

3.4 Aktuelle datakilder

SmartHjem vil benytte informasjon om forbruket til elektriske artikler, strømpriser og værddata. De følgende punktene vil redegjøre for hvordan dette vil bli tatt i bruk i denne oppgaven.

Elektriske artikler

SmartHjem vil måtte kunne trenge informasjon om forbruket til de elektriske artiklene i hjemmet. Teknologien for å gjøre dette mulig finnes allerede, men vil i denne oppgaven bli erstattet av gjennomsnittlig forbruksdata fra ulike kilder [18, 19, 20].

Strømpriser

Som nevnt tidligere i oppgaven vil strømprisene bestemmes av tilbud- og etterspørselsmodellen, og variere fra time til time. Dette er viktig å ta i betraktning når det skal gis anbefalinger til folket om forbruk, men også som en del av regnestykket når kostnaden for forbruk skal regnes ut. Det var derfor svært viktig å ha dette med i beregningene i systemet.

I virkeligheten vet man ikke hva de fremtidige strømprisene blir over tid, men i SmartHjem er det nødvendig å ha et estimat på dette. I denne oppgaven ble det derfor tatt et valg om å bruke strømpriser per time for Trondheim i 2013, levert av NordPool [21] som utgangspunkt i beregningene. Det er dermed mulig for å få en indikasjon på hvordan strømprisene vil kunne varierer i løpet av en uke, som vist i figur 1.1.

Værdata

Været og utetemperaturene varierer også fra time til time. Dette fører til ulik behov for oppvarming. En lav utetemperatur gjør også at innetemperaturen faller fortere og effekten av oppvarmingen må økes for å opprettholde ønsket temperatur, slik som en høyere utetemperatur i andre enden kan gi et lavere oppvarmingsbehov.

Til denne oppgaven ble det sett på ulike tjenester som leverer predikert værdata frem i tid. Slike data er sjelden helt nøyaktige, og de gir en lavere sannsynlighet for å være korrekt lenger frem i tid enn i nær fremtid. Likevel gir slike værvarsler en indikasjon på hvordan været trolig vil forandre seg. Etter å ha sett på ulike tilbydere av slik gratis værdata, ble det valgt å benytte OpenWeatherMap [22], en tjeneste som tilbyr et enkelt API det var lett å koble seg opp i mot, og som gav de dataene det var behov for til systemet.

Kapittel 4

Krav

For et IT-system settes det normalt opp en rekke krav som brukes til å evaluere suksessen til systemet. I dette kapitlet vil det bli sett nærmere på de funksjonelle- og kvalitetskravene for SmartHjem, og hva som ville være viktig å få med i en implementasjon.

4.1 Funksjonelle krav

De funksjonelle kravene er et sett med krav som må bli implementert for å tilfredstille funksjonalitet til det grunnleggende systemet. Kravene som er listet opp i tabell 4.1 er klassifisert på formen Funksjonelt Krav (X, Y). Der X har en verdi som skal illustrere viktigheten av at nettopp dette kravet blir tilfredsstilt, og Y er antatt vanskelighetsgrad for å få implementert kravet. X og Y kan tilegnes ulike verdier; Høy (H), Middels (M) eller Lav (L).

Funksjonelle krav	
FK1	Kunne simulere temperaturer i hjemmet. (H, H)
FK2	Få oversikt over totalforbruk og kostnad per enhet i hjemmet. (H, M)
FK3	Kunne skru av og på enheter. (H, M)
FK4	Kunne sette komforttemperatur og en nedre temperaturgrense i huset. (M, H)
FK5	Kunne velge om fokus skal være på komfort eller økonomi (M, M)
FK6	Bruker bør kunne sette sin tilstedeværelse i hjemmet (M, M)

Tabell 4.1: Funksjonelle krav

FK1 Kunne simulere temperaturer i hjemmet

- I systemet skal det være mulig å simulere temperaturforløpet i hjemmet frem i tid, basert på ulike brukerinnstillinger. Simuleringen skal også vise de antatte fremtidige kostnadene som er knyttet til oppvarmingen.

FK2 Få oversikt over totalforbruk og kostnad per enhet

- Det skal være mulig å kunne se detaljer for hver enkelt elektrisk artikkel i hjemmet, se forbruket dens og kostnad over tid.

FK3 Kunne skru av og på enheter

- Fra systemet skal det være mulig å styre elektriske artikler i hjemmet, hovedsaklig ved å kunne skru disse av eller på.

FK4 Kunne sette temperaturen

- Fra systemet skal det være mulig å sette den ønskede komforttemperaturen i hjemmet. I tillegg skal det kunne settes en nedre grense som temperaturen i huset ikke skal gå under.

FK5 Fokus på komfort eller økonomi

- I systemet bør det være mulig å kunne velge om det er viktig at temperaturgrensene overholdes fullt ut, eller om økonomien er viktigst. Ved å fokusere på økonomi skal systemet finne den beste måten å spare penger på, samtidig som de satte temperaturgrensene hovedsakelig blir fulgt.

FK6 Sette tilstedeværelse

- Det bør være mulig for brukeren å sette opp en tidsplan over når beboerne forventer å være hjemme i huset, slik at huset kan tilpasse temperaturen utfra dette.

4.2 Kvalitetskrav

Kvalitetskrav til et system blir også sett på som ikke-funksjonelle krav. Dette er krav som brukeren gjerne forventer at er tilfredsstilt for at de skal føle seg trygge når de bruker systemet, at de føler at systemet er sikkert og at de er fornøyd med brukeropplevelsen de sitter igjen med.

Til denne oppgaven har ISO/IEC 25051-standarden [23] blitt brukt som referanse for kvalitetskravene. Dette er en internasjonal standard for kvalitetskrav. Det benyttes også for kvalitetskravene en Kvalitetskrav (X,Y) terminologi, der som for de funksjonelle kravene betyr at X er hvor viktig kravet er, og Y sier noe om antatt vanskelighetsgrad å implementere i systemet.

Kvalitetskrav	
KK1	Systemet bør ha en oppetid på 95%. (H, M)
KK2	Systemet må ha en 100% sikker tilkobling. (H, H)
KK3	Nye brukere bør lett kunne bruke systemet. (M, M)
KK4	Kunne bruke systemet på ulike enheter. (M, M)

Tabell 4.2: Kvalitetskrav

KK1 Systemet bør ha en oppetid på 95%.

- Det bør være mulig for brukere å få tilgang til systemet minst 95% av tiden. Dersom brukersystemet er nede, bør ikke dette påvirke de elektriske artiklene i huset. Det er ikke essensielt at systemet er tilgjengelig til enhver tid, ved for eksempel oppdateringer. Likevel er det forventet at det hovedsaklig er kjørende for en god brukeropplevelse.

KK2 Systemet må ha en 100% sikker tilkobling

- Det er svært viktig at ingen uautoriserte brukere får tilgang til en annens system. Dette er viktig da det er snakk om styring av strøm som i verste fall kan være brannfarlig.

KK3 Nye brukerne bør lett kunne bruke systemet

- Det bør være en lav terskel for nye brukere å ta i bruk systemet. Med dette menes at det skal være intuitivt for folk flest i alderen 18-80 år å styre enheter med systemet, kunne se forbruk og simulere kostnad for ulike brukerinnsstillinger.

KK4 Kunne bruke systemet på ulike enheter

- Det skal være mulig å bruke dette systemet enten brukeren benytter seg av mobil, nettbrett eller datamaskin koblet til internett.

Kapittel 5

Arkitektur

Utviklingen av brukersystemet var en kontinuerlig prosess, der det stadig dukket opp nye tanker, idéer og krav. Det ble gjort noen valg basert på tidligere erfaringer om utvikling av nettsted, i tillegg til at det ble tatt i bruk rammeverk for eksperimentering for å skape noe nytt og spennende. Da det ikke fra begynnelsen var en klar tanke om hvordan det endelige systemet skulle bli, ble det fokusert på å bygge opp systemet slik at arkitekturen var både moduler- og modifiserbar for endringer.

Det ble valgt å bruke teknologier som er støttet av en rekke enheter for å tilfredsstille uttrykket '*Write once - Run anywhere*', samt forsøke å opprettholde krav KK4 i del 4.2. Koden som er skrevet er hovedsaklig på engelsk da det gjør det lettere med tegnsetting, samt gjør det lettere å forklare problemer underveis med kode for internasjonale forum på internett.

Dette kapitlet tar for seg hvordan arkitekturen til det endelige systemet er bygget opp, designmønstre som ble brukt underveis, hvordan kommunikasjonen foregår og hvordan koden henger sammen.

5.1 Teknologier

Der er blitt benyttet en rekke teknologier i utviklingen av SmartHjem for å nå de ønskede kravene. Flere av teknologiene som har blitt brukt er svært utbredt innenfor dagens webutviklingen, er godt dokumentert og bra støttet på ulike nettverkstilkoblede enheter.

HTML5

HTML5 (Hyper-Text Markup Language) er et språk for å strukturere og presentere innhold til internettsider. Språket ble først definert i 1989, men har siden den gang blitt endret flere ganger. HTML5 er fortsatt under utvikling, men er ventet å ta over som standard W3C (World Wide Web Consortium) innen få år.



Samtidig som å være et skrivespråk, består HTML5 også av en rekke teknologier, funksjoner og API'er som gjør det mulig å håndtere multimedia og grafisk innhold på en nettside. Den nye funksjonen `<canvas>`, gir mulighet til å tegne grafikk, mens `<audio>` og `<video>` gir programmereren mulighet til å legge ut henholdsvis lydklipp eller videosnutter. Disse funksjonene kombinert med JavaScript og CSS gir muligheter for å lage noen svært komplekse nettsider.

En av de store fordelene med at dette blir et standardspråk er at det også blir gjeldende for mobile nettlekere. Som følge av dette kan en nå etterhvert holde seg til et språk som fungerer på alle enheter.

WebSocket ble innført i HTML5, og med det har det åpnet seg nye muligheter for å kommunisere mellom klient og server mer effektivt [24].

CSS3

CSS3 (Cascading Style Sheet 3) er det mest vanlige design-verktøyet til utvikling av HTML sider. Ved å definere ulike klasser og id'er i HTML- objektene kan en ved hjelp av CSS forandre på bakgrunnsfarge, tekstfarge, tekststørrelse, plassering på skjermen, størrelse og lignende.



I versjon 3 av CSS har det blitt lagt støtte for *media queries*, med muligheter for å tilpasse designet av siden til ulike skjermstørrelser. Innhold som er mindre viktig for siden kan skjules dersom det viser seg å ikke være plass [24].

JavaScript

JavaScript er et programmeringsspråk som gjerne brukes til å manipulere objekter på en nettside og gjøre det mer interaktivt. Selve scriptet blir utført på brukerens maskin, direkte i nettleseren, og krever derfor ingen egen nedlasting for å fungere. Språket er også nyttig i den forstand at det støttes av de aller fleste nettlesere, også til mobil. For å gjøre en nettside mer dynamisk, med animasjoner og overganger er JavaScript et godt egnet verktøy.



En annen stor fordel med dette programmeringsspråket er at det finnes mange ulike biblioteker som kan benyttes. Ved å kombinere disse inn i koden, gir det flere muligheter til å raskere å løse definerte problemer [24].

jQuery.js

jQuery er et raskt og kraftig bibliotek til JavaScript som gjør oppgaven med å traversere og manipulere HTML objekter mye enklere. Dette biblioteket støttes av de fleste nettlesere, slik at terskelen for å bruke det i et JavaScript-prosjekt er svært lav [24].

Twitter Bootstrap

Twitter Bootstrap er et verktøy for å lage design til nettsider og nettapplikasjoner. Det er et gratis åpent rammeverk som tilbyr maler for typografi, knapper, navigasjon og andre grensesnittkomponenter for HTML og CSS [25].

Highcharts

Highcharts er et verktøy som genererer grafer, og er skrevet i HTML5 og Javascript. Det tilbyr svært modulerbare grafer og diagrammer for interaktive nettsteder og applikasjoner.



For personlige og ikke-profit nettsteder er Highcharts gratis å bruke, men for businessapplikasjoner kreves en lisens. Kundebasen til selskapet inneholder mange anerkjente bedrifter som for eksempel IBM, NASA, Siemens, Sony og BBC [26].

MATLAB

MATLAB er et program for høynivå utregninger, visualisering og programmering. Data kan analyseres og det kan brukes avanserte verktøy for behandling av innkommende data. MATLAB inneholder er et stort sett av innebygde matematiske funksjoner, slik at det går raskt å starte utviklingen av mer utdypende operasjoner. MATLAB kan brukes til signalbehandling, kommunisering, video- og lyd-prosessering, og er svært utbredt blant ingeniører som krever ulike utregninger utført [27].



Simulink er en integrert del av MATLAB som blant annet støtter simulering og automatisk kodegenerering fra blokkdiagrammer [28].

Simple.js

Simple.js er et JavaScript-rammeverk som brukes til å opprettholde en *Model-View* arkitektur i et JavaScript prosjekt. Rammeverket bygger i stor grad på Backbone.js og Spine.js, som begge er biblioteker for å holde opprettholde en slik arkitektur. Fordelen med å benytte seg av en denne arkitekturen i JavaScript-prosjekter er for å sørge for at koden blir lettere å vedlikeholde, samt skaléres bra dersom systemet blir større. Arkitekturen baserer seg på at modellen har ansvar for å holde på data, samt hente og sende data til serveren, mens viewet representerer data fra modellen.

Modellen og viewet kommuniserer seg i mellom med *eventer*. Ved en slik arkitektur er det mulig å ha flere views på en nettside som lytter til forskjellige modeller, og som kan oppdatere seg uavhengig av hverandre. Dette gjør det enklere å legge til eller fjerne views, i tillegg til å sørge for at ikke hele siden behøver å oppdatere seg dersom det skjer en endring i en modell [24].

KineticJS

HTML5 tilbyr muligheten å lage figurer og tegninger i et canvas. Kineticjs er et rammeverk som utnytter dette ved å legge til rette for å også lage animasjoner, overganger, lagdeling av figurer, eventhåndtering med mer [29]. I denne oppgaven gjør dette rammeverket det mulig å designe plantegninger til boliger med tilhørende klikkbare elektriske artikler.



Node.js

Node.js er en JavaScript plattform for raskt å bygge og sette opp skalérbare nettverksapplikasjoner. Ved hjelp av dette kan det med noen få kodelinjer sette opp en webserver. Koden 5.1, lagret i for eksempel en server.js fil er alt som skal til for å komme i gang med en enkel web server.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');

console.log('Server running at http://127.0.0.1:1337/');
```

Kode 5.1: Oppsett av Node.js

Node.js benyttes i dag av store selskaper som Microsoft, The New York Times, eBay og LinkedIn til å for eksempel håndtere raske oppdateringer på deres online tjenester [24].

Socket.IO

Socket.IO er et JavaScript bibliotek for å lage sanntids webapplikasjoner. Det består av to deler for henholdsvis klient og server. Klienten kjører direkte i nettleseren, mens serveren kjører på en Node.js kobling. Figur 5.1 viser hvordan Socket.IO settes opp på server og klient. En av fordelene med Socket.IO er at API'et også er skrevet i JavaScript, noe som gjør langt enklere å bruke i et JavaScript system, enn dersom biblioteket hadde vært skrevet i et annet språk.

SERVER	CLIENT
<pre>var io = require('socket.io').listen(80); io.sockets.on('connection', function(socket) { socket.emit('news', { hello: 'world' }); socket.on('my other event', function (data) { console.log(data); }); });</pre>	<pre><script src="/socket.io/socket.io.js"></script> <script> var socket = io.connect('http://localhost'); socket.on('news', function (data) { console.log(data); socket.emit('my other event', { my: 'data' }); }); </script></pre>

Figur 5.1: Oppsett av Socket.IO mellom klient og server

Socket.IO tilbyr også kommunikasjon til en server i et annet domene (*Cross-Domain*), da Socket.IO, som tilbyr en WebSocket kobling, ligger som et sikkert overlag til en TCP-kobling. Det er med andre ord ikke mulig å koble seg direkte til en TCP-kobling fra en nettapplikasjon uten å ha et sikkert oppgradert overlag, som nettopp Socket.IO, med WebSocket, tilbyr [24].

5.2 Websystemet

SmartHjem-systemet er satt sammen av en rekke komponenter som hver har sitt ansvarsområde. Dette er viktig for lesbarheten, men også for å gjøre avhengighetene mellom de ulike delene minst mulig. Websystemet har en *Model-View* arkitektur, og modellene og viewene vil presenteres i de påfølgende avsnittene.

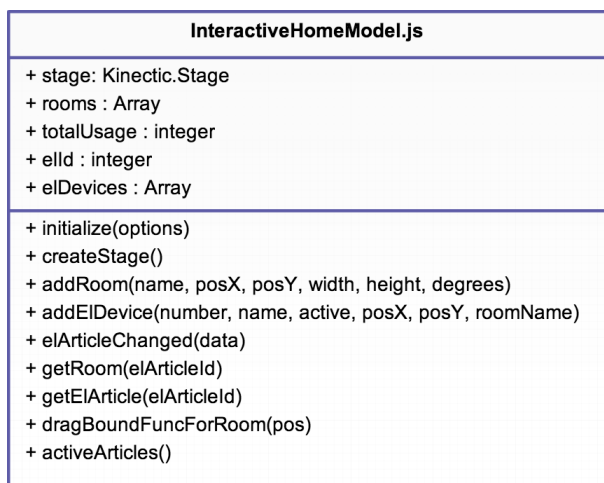
Modeller

Modellene har som ansvar å behandle og holde på data som viewet kan vise. Det er også modellene som håndterer kommunikasjon til server og sørger for at dataen er av rett type, og klare til å vises.

InteractiveHomeModel.js

I denne klassen genereres demonstrasjonshjemmet som inneholder ulike elektriske artikler. Disse plasseres i det Kinetic-tegnede hjemmet, og det settes tilstander og data til hver artikkel. Dette er data om hva som skal skje ved klikk på objektet, om artikkelen er av eller på, hvilket rom den tilhører, og dens posisjonen i rommet. I tillegg har denne klassen metoder for hva som skjer når objekter blir flyttet på, hvilke artikler som er på, og muligheter for å legge til nye rom og elektriske artikler. Hver artikkel som legges til inneholder informasjon om effektstyrke og til hvilke tider den er aktiv. Mer utdypende om dette i `eObjects.js`.

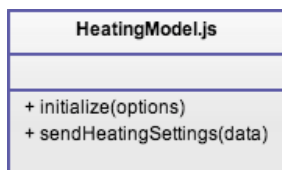
For oversikt over variabler og metoder i denne klassen, se figur 5.2.



Figur 5.2: InteractiveHomeModel.js

HeatingModel.js

HeatingModel sørger for at datainnstillinger om varme og tilstedeværelse i hjemmet, satt i HeatingView, blir omformet til en tekststreng, og deretter sendt til serveren over websocket-koblingen. I tillegg tar modellen i mot ferdig beregnet data og sender en beskjed til viewet om det. Dette er fordi modellen, og ikke viewet, skal håndtere kommunikasjon.



Figur 5.3: HeatingModel.js

EldeviceInfoBoxModel.js

EldeviceInfoBoxModel er modellen for hver enkelt elektrisk artikkel i hjemmet. Den består av et navn, en id, effekten til artikkelen, hvilket rom den er i, om den er av eller på ved nåværende tidspunktet, og hvilke timer den er tenkt å være på.

ElDeviceInfoBoxModel.js
+ elId : integer
+ name : String
+ effect : integer
+ room : String
+ active : boolean
+ activeHours : integer
+ initialize(options)

Figur 5.4: EldeviceInfoBoxModel.js

CurrentUsageModel.js

Denne klassen holder på alle de elektriske enhetene i huset. Modellen kan beregne summen av det totale antall på-timer for de elektriske artiklene ved å summere dette fra alle enhetene i metoden *getActiveHours()*.

CurrentUsageModel.js
+ elDevices : Array
+ totalEffect : integer
+ activeHours : Array
+ initialize(options)
+ getActiveHours()

Figur 5.5: CurrentUsageModel.js

Viewene

Viewene sørger for å visualisere dataene som modellene sender. De har ansvar for animasjoner, overganger mellom sidene og for at dataen blir riktig fremstilt.

HeatingView.js

Det er i HeatingView at brukeren av systemet velger i kalenderen de tidene brukeren tenker å være hjemme i løpet av uken. Det er også her det settes ønsket komforttemperatur ved tilstedeværelse, samt en nedre temperaturgrense som temperaturen kan falle til dersom ingen er hjemme eller sover. Deretter kan brukeren sette antall dager det ønskes å simulere kostnad for fremover i tid, samt vektingen mellom pris og komfort.

Når innstillinger er satt blir data først sendt til HeatingModel, før det blir sendt videre til Matlab for utregning. Resultatet blir så sendt tilbake fra Matlab, og visualisert i en graf og en tabell.

Når dataen kommer tilbake fra Matlab er det ingen garanti for at all dataen blir sendt i en melding, se 5.3, det betyr at det også er viktig i denne klassen å analysere det som kommer inn, og sørge for at tekststrengen, som skal settes sammen til et JSON-objekt, er fullstendig.

HeatingView.js
+ matlab : Object + currentComfort : integer + currentLimit : integer + currentNumOfDays : integer
+ initialize(options) + simulation() + getVisibleEvents() : Array + createCalendar() + controlMatlabData(data) + progressBar(daysToSimulate) + addTableElement(simulationObject, number) + generateOrUpdateTempSimulationGraph(dataString)

Figur 5.6: HeatingView.js

EldeviceInfoBoxView.js

Hver elektrisk artikkel i systemet har hver sin tilhørende boks som holder på informasjon om forbruk i kilowattimer og pris, samt mulighet for å skru av eller på. Fra denne boksen kan det også aktiveres en graf som viser forbruket til artikkelen i en graf, over en tidsperiode som kan velges av brukeren.

EldeviceInfoBoxView.js
+ template: String
+ initialize(options) + showBox() + closeEldeviceInfoBox() + showEldeviceGraph(elDeviceData)

Figur 5.7: EldeviceInfoBoxView.js

CurrentUsageGraphView.js

CurrentUsageGraphView viser data om det totale forbruket og kostnaden til dette. Dette gjør det lett for brukeren å se endring av totalforbruket over tid. Tidsintervallet som vises i informasjonsboksen settes ved seleksjon av ønsket område i grafen.

CurrentUsageGraphView.js
+ totalInfoContentTemplate: String
+ highChart: HighChart
+ initialize(options)
+ showGraphInfo()
+ createGraph(elDevicesData)

Figur 5.8: CurrentUsageGraphView.js

Index.html

Selv om man i applikasjonen kan navigere seg mellom tre ulike sider, skru av og på informasjonsbokser og grafer, ligger alt sammen på en nettside. Det er dette, som på engelsk kalles en *One-Page-Application*. Alt det som brukeren ser på siden blir vist og byttet ut ved hjelp av JavaScript.

Ved navigasjon i applikasjonen blir viewene og modellene initialisert, og etter det er det de som styrer siden. Dette resulterer i en rask og responsiv side, og det er enkelt å bytte ut og sette sammen deler av siden til en applikasjon. I tillegg opprettes det i denne klassen en WebSocket-forbindelse til serveren. Strømpriser og værdata blir hentet inn og lagret i en global variabel, slik at dette raskere kan aksesserer når det er behov for det.

Utils.js

Hensikten med denne klassen er å samle opp nyttefunksjoner som skal løse generelle oppgaver som ikke er knyttet til en spesifikk del av systemet. Mye av utregningene, særlig av grafene, foregår i millisekunder og det er greit å ha noen statiske variabler som holder på verdier som en dag, en uke og en time i millisekunder. Det er også i denne klassen at vær- og strømpriserdata blir hentet inn og tolket.

SP.Utils.js
+ SP.Utils.OneDay : integer + SP.Utils.OneWeek : integer + SP.Utils.OneHour : integer + SP.Utils.HoursInDay : integer
+ SP.Utils.parseFile(data) + SP.Utils.parseWeatherFile(data) - successFunc(data) - linearFunction(y1, y2) - extractTemperatures(hourTemperatures) + SP.Utils.elUsageTime(elObject) + SP.Utils.getPowerPricesFromNowTheNextDays(numOfDays) + SP.Utils.getWeatherForecastTheNextDays(numOfDays) - sortArrayOfDates(date1, date2) - roundDownMinutes(data) + SP.Utils.getActivityList(events, comfortTemp, limitTemp) + SP.Utils.parseMatlabString(dataString)

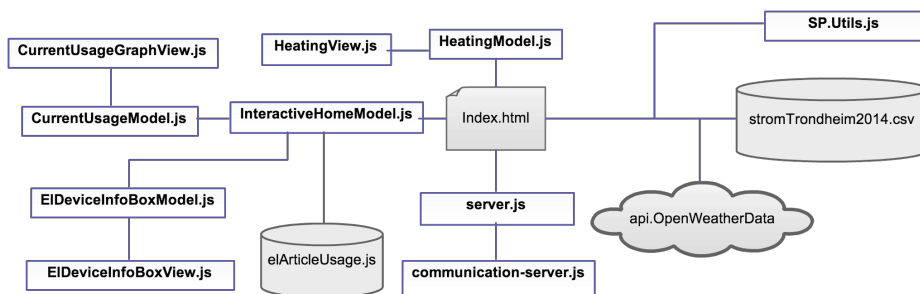
Figur 5.9: Utils

elArticleUsage.js

Da systemet i denne oppgaven ikke har tilgang på faktisk strømforbruk per enhet, ble det nødvendig å benytte seg av diverse kilder [18, 19, 20] for å anta forbruket. Disse antagelsene ble lagt i *elArticleUsage.js* der hver enhet fikk et navn, en id, en effekt og lister for antatt aktivt timesforbruk per dag. Dette var et grovt estimat, men nyttig for å kunne se estimert dags-, ukes- og månedsforbruk for flere artikler.

Klassediagram

Det har frem til nå blitt beskrevet noe om funksjonaliteten i hver klasse, men ikke så mye om koblingen mellom disse. I figur 5.10 forklares noe av dette. For et mer detaljert klassediagram se vedlegg A.



Figur 5.10: Overordnet klassediagram av websystemet

Som nevnt tidligere benytter systemet rammeverket Simple.js for å opprettholde en Model-View-arkitektur, hvor viewet lytter til endringer i en modell, og modellen sender dataoppdateringer til viewet. Måten dette foregår på er med såkalte *Simple.Events*. For å sende ute en event benyttes en trigger, som vist i kode 5.2.

```
// Data som sendes med et passende event-navn
Simple.Events.trigger("event-navn", data);
```

Kode 5.2: Simple.Event-trigger

For å lytte på et event benyttes *on*, som vist i kode 5.3

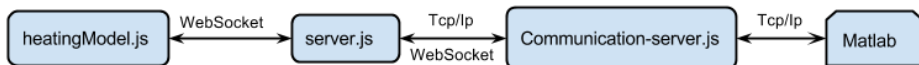
```
// Funksjon som kalles ved data til event-navn
Simple.Events.on("event-navn", function(name) {
  console.log(name)
});
```

Kode 5.3: Simple.Event-lytter

Denne mekanismen gjør at det er enkelt og svært modulerbart å velge om et view skal lytte til ulike hendelser, som for eksempel om en knapp trykkes på eller ikke.

5.3 Kommunikasjon i systemet

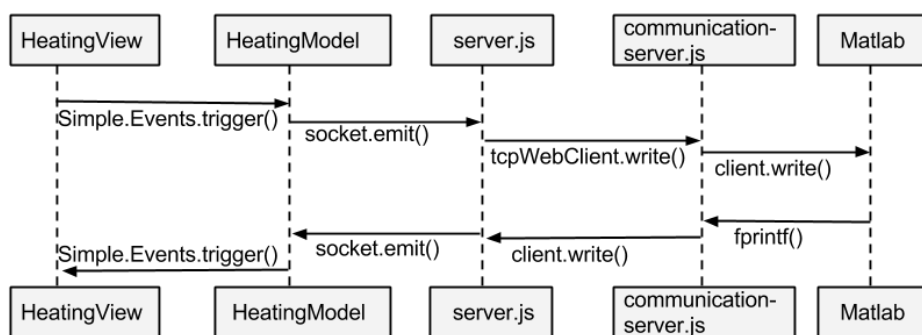
Til regulering av oppvarmingen og estimering av fremtidige strømutgifter for oppvarming ble det valgt å bruke MATLAB. Ønskede innstillinger fra brukeren skal kunne justeres fra et webgrensesnitt, samt at resultater fra simuleringen skal vises på samme sted. For å få til dette ville det være nødvendig å sette opp en kommunikasjonsforbindelse mellom websystemet og MATLAB.



Figur 5.11: Kommunikasjonsprotokoller i systemet

Som figur 5.11 viser er to servere koblet sammen. Grunnen til dette er for å få til kommunikasjon mellom Node.js Net (*TCP/IP*) og Socket.IO (*WebSocket*). Som nevnt i avsnittet om Socket.IO er det ikke mulig å kommunisere direkte med TCP/IP til en nettside, grunnet sikkerhetsmomenter.

MATLAB på sin side tilbyr kun kommunikasjon via TCP/IP, og ikke WebSocket. Systemet ble derfor konstruert slik at server.js oppretter en Node.js-Net forbindelse med Communication-server.js, i tillegg til at det opprettes en Socket.IO-kobling inne i Node.js-Net-koblingen. Dette gjør det mulig at tekststreng-data kan sendes og mottas mellom MATLAB og webklienten.



Figur 5.12: Meldingsflyt i systemet

I figur 5.12 vises det hvordan meldingsflyten går mellom websystemet og MATLAB ved nye brukerinnstillinger. Først sendes det en intern melding fra view til modell, før dette *emittes* videre til server.js over WebSocket. Dette blir så sendt til communication-server.js over TCP/IP-koblingen som er pakket inn i WebSocket-koblingen i server.js. Videre blir dataen sendt over TCP/IP til MATLAB, som tar dataen i mot, gjør beregninger og sender tilbake på tilsvarende måte til HeatingView der dataen blir presentert.

Håndtering i MATLAB

Instrument Control Toolbox er en del av MATLAB som gjør det mulig å koble seg opp mot instrumenter og elektroniske enheter, via ulike protokoller, som for eksempel TCP/IP [30]. Med dette verktøyet kan MATLAB motta og sende datasignaler i form av tekststrenger.

Som kode 5.4 viser, så opprettes det først en kobling opp mot ønsket ip-adresse og portnummer. Deretter åpnes koblingen. Videre kan en while-løkke benyttes for å lytte til om det er *BytesAvailable*, og dersom det er det leses dette inn og lagres i *DataReceived*-objektet. Til slutt lukkes og slettes koblingen [31].

```
t = tcpip('localhost', 8000);
fopen(t);

while (get(t, 'BytesAvailable') > 0)
    t.BytesAvailable
    DataReceived = fscanf(t)
end

fclose(t);
delete(t);
clear t
```

Kode 5.4: Oppsett for å starte TCP/IP kommunikasjon med Matlab

For denne oppgaven er det ønskelig å sende et sett av ulike data til MATLAB som gjør beregninger, og så sende resultater tilbake. Instrument Control Toolbox er beregnet for å sende og motta enkle beskjeder, gjerne kanskje bare i form av et tall eller en kort tekstkommando. Det ble derfor nødvendig å se på andre muligheter for å lytte til meldinger, samt formatere disse fra en tekststreng til et MATLAB-objekt.

BytesAvailableFcn er en funksjon som blir kjørt etter at *BytesAvailableFcnMode* har blitt aktivert. Dette kan enten skje dersom *BytesAvailableFcnMode* er en *terminator*, altså et gitt tegn som avslutter meldingen, eller *byte* som trigger funksjonen dersom det har kommet inn en gitt mengde data til koblingen. For denne oppgaven er det ikke gitt hvor store data som ville bli sendt for hver gang, og derfor ble $\backslash n$ (ny linje) som *terminator* valgt å bruke.

```
t.BytesAvailableFcnMode = 'terminator';  
t.BytesAvailableFcn = @runSimulation;
```

Kode 5.5: Funksjoner på transportobjektet

Siden TCP/IP-koblingen til MATLAB kun kan lese tekststrenger, blir JSON (JavaScript Object Notation) som sendes fra websystemet omformet til en tekststreng før det sendes. Det er ønskelig å sette sammen strengen til slik den opprinnelig var, og det blir gjort ved å bruke en *JSON Parser*[32]. Variablene blir da lagt inn riktig med tilhørende data i MATLAB-workspacet.

Når beregningene er gjennomført, blir de omgjort til en tekststreng før de blir sendt tilbake til websystemet. Det er ingen garanti for at hele strengen blir sendt tilbake i en melding. Derfor er det viktig at dette blir håndtert av websystem-viewet, og at det er et element på slutten av tekststrengen som forteller at strengen har nådd slutten. For denne oppgaven ble dette symbolet valgt til å være en 'X', da dette ikke vil komme i konflikt med noe annet symbol i resultatstrengen.

Del III

Regulering

Kapittel 6

Modellering

En modell er et sett med ligninger som matematisk beskriver oppførselen til et system. Et system kan ha flere ulike egenskaper som kan modelleres, men ofte lages det en modell kun for de egenskapene som er av interesse i det formålet modellen skal brukes. I denne oppgaven er det temperaturen i en bolig som er interessant å kjenne oppførselen til, og det skal derfor utvikles en termisk modell av en bolig, samt for oppvarmingssystemet til boligen.

Modellen brukes for å kunne simulere hvordan temperaturen vil oppføre seg når reguleringsystemet som utvikles, blir benyttet. Dermed vil det være mulig å forutse hvordan reguleringsystemet i praksis vil kunne fungere, uten å måtte teste det ut på et fysisk hus. Det er viktig å være klar over at en modell alltid bygger på antagelser og forenklinger, og vil dermed alltid avvike noe fra virkeligheten. Dette diskuteres videre i delkapittel 6.5.

I tillegg til å bruke modellen for simulering, er det også nødvendig med en modell av systemet til MPC-regulatoren. Hvordan modellen brukes i regulatoren står forklart nærmere i kapittel 7.

6.1 Modelleringssteknikker

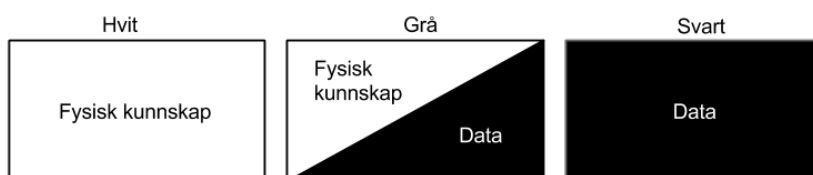
Målet med modelleringen er å finne relasjonen mellom inngang- og utgangsverdiene til et system, og det finnes ulike tilnæringsmetoder for å finne denne relasjonen [33].

En modell kan bygges opp basert på kunnskap om fysikken i systemet. Denne tilnæringsmetoden går ofte under navnet hvit- boks modellering, og en slik teknikk krever gjerne god innsikt i systemet som skal modelleres. Fordelene er at resultatet blir en modell med ligninger som har et stort gyldighetsområde, samt at det kreves få data for å validere disse ligningene.

I motsetning til hvit- boks modellering finnes det en tilnæringsmetode som blir kalt svart- boks modellering. Denne teknikken går ut på å anse systemet som skal modelleres som en svart boks, uten å ha innsikt i hva den består av. Modellen bygges opp ved å sende inn et variert sett med inngangsverdier, og deretter analysere de resulterende utgangsverdiene. Denne metoden krever ofte mindre utviklingskostnader i forhold til hvit- boks modellering. Ulempen er at det gjerne krever et stort datagrunnlag, og modellen ikke nødvendigvis blir gyldig utover området som disse dataene dekker.

Dersom både fysikk og dataanalyser blir benyttet for å utvikle modellen kan man si at en grå- boks modellerings teknikk blir benyttet. Denne kategorien dekker mange ulike nyanser.

De tre hovedkategoriene av modelleringsteknikkene er illustrert i figur 6.1, og i denne oppgaven er det valgt å benytte en hvit- boks tilnærming i modelleringprosessen. Modellen bygges dermed opp basert på kunnskap om de termiske egenskapene til et hus, som beskrevet i neste delkapittel.



Figur 6.1: Ulike modelleringsteknikker

6.2 Teori

Teorien som er nødvendig for å sette opp en termisk modell baserer seg på enkle prinsipper fra termodynamikk [34]. I tillegg ble det hentet inspirasjon fra et innebygd eksempel i MATLAB[35].

Varmestømmen \dot{Q} i en bolig vil være bestemt av oppvarmingen fra varmelementene Q_i og varmetapet Q_e til omgivelsene:

$$\dot{Q} = Q_i + Q_e \quad (6.1)$$

Varmeenergien Q er avhengig av den spesifikke varmekapasiteten c , massen m og temperaturen T til det oppvarmede stoffet (i dette tilfellet luft):

$$Q = cmT \quad (6.2)$$

Varmetapet til omgivelsene er beskrevet av Newtons avkjølingslov. Denne

loven viser at varmetapet er proporsjonalt med forskjellen mellom innendørstemperaturen T og temperaturen til omgivelsene T_{out} :

$$Q_e = -K_e(T - T_{out}) \quad (6.3)$$

Varmetapstallet K_e er en positiv konstant som beskriver energitapet per celsius grad i temperaturforskjell til omgivelsene, og er dermed et mål på effekten av isolasjonen i huset. Varmetapstallet for en konstruksjon (for eksempel en vegg eller et vindu) kan finnes ved formelen:

$$K_{ei} = \frac{k_i A_i}{l_i} \quad (6.4)$$

hvor A_i er overflateareal, l_i er materialets tykkelse og k_i er materialets konduktans. Konduktanstallet beskriver materialets ledeegenskaper, der en lav verdi for k_i indikerer god isoleringsevne. For enkelhets skyld vil det i denne modellen kun skilles mellom vinduer og øvrige konstruksjoner, som vegger, tak og gulv.

Husets totale varmetapstall K_e kan sees på som en parallellkobling av konduktanser, og finnes derfor ved å summere varmetapstallene for vinduer K_{e1} og for øvrige konstruksjoner K_{e2} :

$$K_e = K_{e1} + K_{e2} \quad (6.5)$$

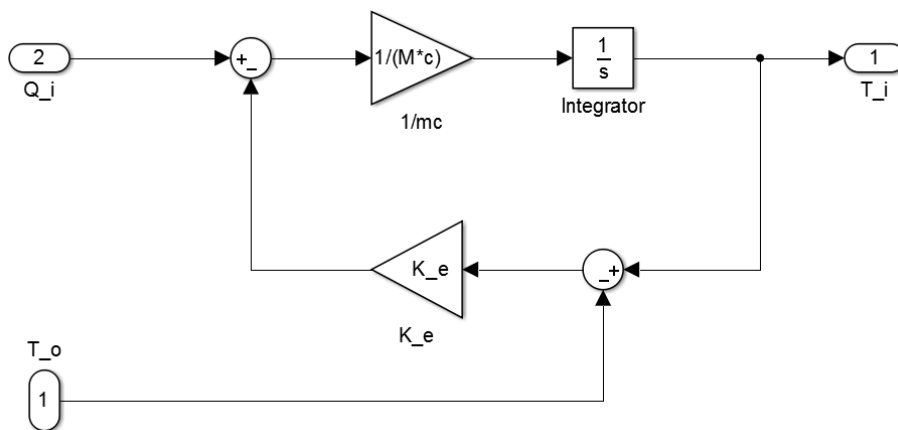
6.3 Termisk modell av en bolig

Ved å sette sammen ligning 6.1, 6.2 og 6.3 får vi en differensialligning som beskriver temperaturendringen i en bolig:

$$\begin{aligned} \dot{Q} &= Q_i + Q_e \\ cm\dot{T} &= Q_i - K_e(T - T_{out}) \end{aligned} \quad (6.6)$$

hvor K_e er gitt av ligning 6.4 og 6.5. Massen til den oppvarmede luften m er avhengig av boligens geometri.

Ligning 6.6 implementeres i Simulink, og resulterer i følgende blokkdiagram:



Figur 6.2: Husmodell

6.4 Modell av oppvarmingssystem

En bolig kan varmes opp med ett eller flere ulike oppvarmingssystemer. I Norge sier nesten halvparten, 47%, at det viktigste oppvarmingssystemet i boligen er elektrisitet. 30% sier at de har vannbåren varme eller varmepumpe som viktigste oppvarmingskilde, og 21% har ved, pellets eller åpen peis som viktigste oppvarmingskilde [36].

Ulike oppvarmingssystemer har forskjellige egenskaper, og bør derfor modelleres forskjellig. Felles for de ulike oppvarmingssystemer er at målet er å varme opp luften i boligen. Hvor effektiv et oppvarmingssystem varmer opp en bolig er avhengig av hvor mye luft som varmes opp per tidsenhet \dot{M} , hvor mange grader temperaturen til denne luften har økt C_{heat} . Virkningen av oppvarmingssystemet kan da beskrives generelt med ligningen:

$$Q_i = C_{heat} \dot{M} c u \quad (6.7)$$

c er her varmekapasiteten til luft. I denne oppgaven antas det at oppvarmingssystemet kun kan operere i to tilstander; av eller på. Et elektrisk oppvarmingssystem som for eksempel panelovner kan passe godt til denne antagelsen. u er dermed en binær variabel som indikerer om varmesystemet er på ($u = 1$) eller av ($u = 0$).

I modellen for varmeelementet er det gjort en forenkling ved å anta at panelovnen varmer opp en konstan mengde luft per tidsenhet, og at denne luften får en konstant økning i temperatur. Det betyr at \dot{M} og C_{heat} i ligning 6.7 er konstanter.

6.5 Antagelser og forenklinger

Når et fysisk system modelleres matematisk vil det alltid foreligge antagelser og forenklinger. Av den grunn vil en matematisk modell alltid avvike noe fra virkeligheten.

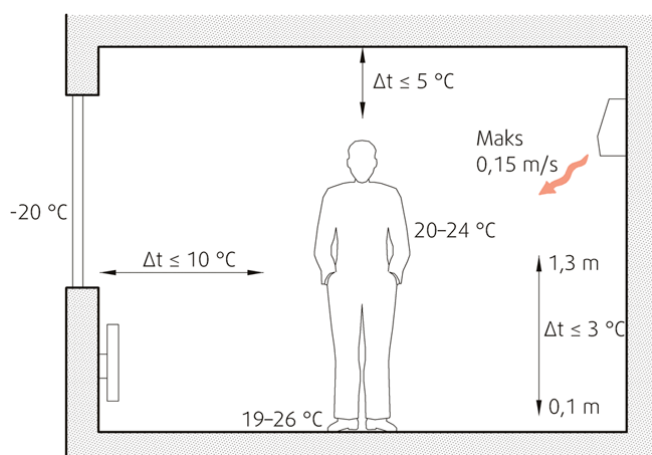
En god modell vil kunne oppføre seg relativt likt virkeligheten, men for å oppnå dette kreves det ofte at det utvikles en kompleks modell som baserer seg på detaljert kunnskap om systemet. Jo flere og grovere antagelser og forenklinger som blir gjort i modelleringen, desto større blir forskjellen mellom modellen og virkeligheten. Hvor god modell som er nødvendig er avhengig av formålet modellen skal brukes til.

I denne oppgaven er det valgt å lage en enkel modell for temperaturendringene i en bolig som bygger på grunnleggende termiske prinsipper. Formålet med modellen er å illustrere hvordan SmartHjem resultatene kan se ut, og i denne oppgaven er det er ikke kritisk at modellen avviker fra virkeligheten. Av den grunn er det ikke lagt ned mer arbeid enn nødvendig for å utvikle modellen.

I modellen som brukes i denne oppgaven er det gjort flere forenklinger. Det må derfor forventes at resultatet av simuleringene vil kunne avvike fra hvordan temperaturen i en fysisk bolig vil oppføre seg. Noe justering av modellen er blitt gjort ved prøving og feiling for at modellen skal oppføre seg mer likt til hva en kan forvente. Noen av forenklingene er diskutert nærmere i de etterfølgende delkapitlene.

Standard til termisk komfort

International Organization for Standardization (ISO) har laget en standard for termisk komfort - NS-EN ISO 7730. Den forteller “at et oppvarmings-system skal opprettholde ønsket romtemperatur i alle rom og kompensere for eventuelle kalde bygningsflater og kalde luftstrømmer (trekk, kaldras), som i stor grad er påvirket av bygningskonstruksjonens varmeisolasjon, tetthet og utforming.” se figur 6.3 [37].



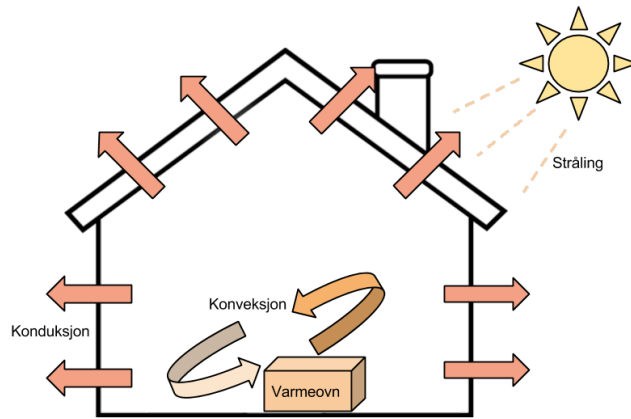
Figur 6.3: Minstekrav til termisk komfort i et rom, gitt ved overflatetemperatur, romtemperatur, luftshastighet og temperaturforskjeller

Termisk komfort i et rom kan ikke bare bestemmes av temperaturen i rommet, men som en samling av ulike faktorer som bygningsflater, luftstrømmer, temperaturdifferanse og lignende. I denne oppgaven blir kun temperaturen i boligen studert, men det er viktig å huske på at i virkeligheten er ikke dette nok for å sikre en god termisk komfort.

Varmetransport

I virkeligheten kan varmeenergi transporteres ved tre forskjellige prosesser [34]:

- **Varmeledning** (*også termisk konduksjon*): I et fast materiale vil varmeenergi kunne forflytte seg fra områder av materialet med høy temperatur til områder med lavere temperatur. Dette skjer fordi atomene i delen av materialet med høyest temperatur også har høyest gjennomsnittlig kinetisk energi, da varmeenergi i realiteten er kinetisk energi til materialets atomer. Atomene vil dytte på naboatomene, og dermed overføre noe av sin kinetisk energi. På denne måten transporteres den kinetiske energien i atomene, og dermed også varmeenergien, til områder med lavere temperatur.
- **Konveksjon** er varmetransport grunnet forflytning av fluider. Konveksjon kan skje naturlig ved forskjeller i massetetthet på grunn av oppvarming. Et eksempel på naturlig konveksjon er at varm luft stiger, da den har lavere tetthet enn kald luft, og varmeenergien transporteres oppover. Konveksjon kan også være kunstig ved at fluider forflyttes ved hjelp av en pumpe. Et eksempel på dette er at blodet i kroppen pumpes rundt i kroppen av hjerte, og dermed transporterer varme rundt slik at kroppen kan holde en jevn temperatur. Konveksjon er en kompleks prosess som er vanskelig å beskrive matematisk.
- **Stråling**: Varmeoverføring kan også skje via elektromagnetisk stråling. Det er denne typen varmeoverføring som overfører varme fra solen til jorden. Stråling kan overføre energi selv i vakuum, i motsetning til konduksjon og konveksjon som kun kan overføre varme ved hjelp av stoffer. Et hvert objekt som har temperatur over det absolutte nullpunkt avgir energi i form av elektromagnetisk stråling.



Figur 6.4: Illustrasjon av varmeoverføringsprosesser

Kun varmetransport via konduksjon er inkludert i den termiske modellen i denne oppgaven. I realiteten vil det være en kombinasjon av de tre prosessene som utgjør den totale varmetransporten, og denne forenklingen vil innføre unøyaktigheter i modellen. Grunnen til at varmetransporten er modellert til å kun bestå av konduksjon er for at det er den enkleste prosessen å modellere. Til tross for at dette vil resultere i en mer unøyaktig modell vil hovedprinsippet med at varmeenergi overføres til omgivelsene bli ivaretatt, og det anses for å være godt nok til dette formålet.

Påvirkning fra omgivelsene

Omgivelsene vil påvirke den termiske oppførselen til boligen. Det er for eksempel forskjell på varmetapet om boligen er en frittstående villa, eller om det er en leilighet i en blokk med andre leiligheter over og under. Modellen i denne oppgaven forenkler omgivelsene til å være like rundt hele huset. Dette er ikke realistisk, men en forenkling som gjør modelleringen lettere.

Været er en del av omgivelsene, og været er ikke statisk. Både temperaturer,

vind, luftfuktighet og skydekke er med på å bestemme hvordan omgivelsene påvirker boligen. Vind og sol vil også virke forskjellig på ulike deler av en bygning, avhengig av hvilken vegg som ligger i le, eller på skyggesiden. I modellen er kun utendørstemperaturer inkludert, men som diskutert er været i realiteten langt mer komplekst til å kun bli beskrevet av en temperatur.

Forenklingen som ble gjort med å kun se på konduksjon for varmeoverføring påvirker både hvordan varme fra huset blir avgitt til omgivelsene, men også omvendt. Selv om utetemperaturen er lavere enn temperaturen inne vil elektromagnetisk stråling fra solen kunne bidra til oppvarmingen. Drivhus er et eksempel der solstrålingen blir brukt til oppvarming ved at solstrålene slipper inn gjennom glasset. Varmestrålingen vil ikke slippes i samme grad ut på grunn av at denne strålingen har en betydelig lengre bølgelengde. Tilsvarende vil skje gjennom vinduer på en bolig, men denne effekten blir sett bort i fra i modellen i denne oppgaven. Denne forenklingen vil føre til størst avvik i tilfeller med solfylte dager og boliger med stort vindusareal.

Oppvarming

Ved oppvarming av en bolig skal luften i boligen varmes opp, og mengden luft som skal varmes opp bestemmes av boligens geometri. I tillegg til å varme opp luften kreves det også energi til å varme opp boligens vegger, gulv og interiør. Hvor mye energi som kreves for å varme opp en bolig er avhengig av varmekapasiteten og volum til massen som skal varmes opp. Å finne massen og varmekapasiteten til luften i boligen er enkelt, men det å ta med vegger, gulv og interiør vil bli komplekst dersom det skal gjøres nøyaktig. Det er av den grunn valgt å kun se på oppvarming av luften, og utelate alt annet.

Oppvarmingsmodellen antar at en konstant luftstrøm \dot{M} blir varmet opp slik

at temperaturen på luftstrømmen øker med en konstant temperaturøkning C_{heat} . Det vil være mer rimelig å tro at temperaturøkningen egentlig er avhengig av luftens temperatur før oppvarmingen, men en konstant oppvarming er lettere å implementere.

I modellen for oppvarmingssystemet er det fokusert på å lage en enkel og generell modell. På grunn av at det finnes ulike oppvarmingssystemer med ulike egenskaper må oppvarmingsmodellen uansett tilpasses det gjeldende oppvarmingssystemet.

Varmetap

Varmetapet i en bolig avhengig mye av isoleringen. Isolering leder varme dårlig, og det gjør at varmeenergien forblir i boligen i større grad. Hovedsaklig forsvinner mesteparten av varmen i en bolig gjennom tak og vinduer. I modellen skilles det mellom varmetap gjennom vinduer og øvrige flater (tak, vegger, gulv), da det gjerne er her den største forskjellen ligger. Differansen på varmetapet gjennom tak, vegger og gulv er satt til å være lik, selv om det i realiteten ville utgjøre en forskjell.

Varmetapet er avhengig av temperaturforskjellen mellom inne og ute, som beskrevet i ligning 6.3. I denne ligningen blir innetemperaturen beskrevet med kun en verdi. Som kjent er varm luft lettere enn kald luft, og vil derfor stige. Temperaturen i en bolig er altså ikke homogen slik som modellen antar, og varmetapet vil derfor være mer kompleks enn ligning 6.3 beskriver.

Ventilasjon i boliger medfører også varmetap, og det anslås at varmetap gjennom ventilasjon utgjør omlag 5% av all energibruk i Norge. Med et ventilasjonsanlegg som gjenvinner varmen fra luften, er det mulig å redusere varmetapet fra ventilasjon med ca. 85% [38]. Varmetap i forbindelse med ventilasjon er ikke tatt med i den termiske modellen av boligen i denne oppgaven.

Kapittel 7

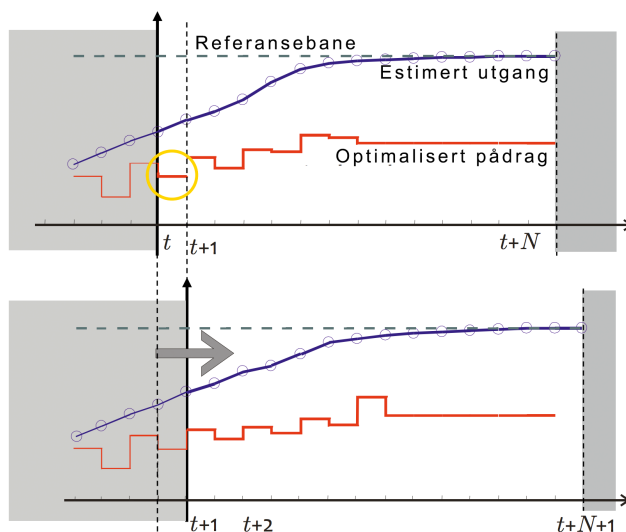
MPC

MPC (Model Predictive Control) er en regulator som beregner det optimale pådraget ved å løse et optimaliseringsproblem. Regulatoren har en modell av systemet som skal reguleres for å kunne estimere den fremtidige oppførselen til systemet, og dette blir brukt til å optimere pådraget [33].

For hver tidsinstans blir optimaliseringsproblemet som er definert i MPC-regulatoren løst. Resultatet blir en sekvens av pådrag som er optimalisert for tiden N , der N refereres til som tidshorisonten til optimaliseringsproblemet. Kun det første pådraget i sekvensen blir benyttet på systemet. Ved neste iterasjon gjentas prosedyren; Optimaliseringsproblemet løses igjen og en ny pådragsekvens beregnes.

MPC-regulatoren tar hensyn til fremtidige hendelser når pådraget optimaliseres. Tidshorisonten N angir hvor langt frem i tid MPC-regulatoren tar med i beregningene. Optimaliseringsproblemet løses på nytt for hver iterasjon, og det gjør at MPC har en flytende tidshorisont. For tiden t vil det beregnes pådrag frem til $t + N$. Ved neste iterasjon $t + 1$ løses det samme optimaliseringsproblemet frem til tiden $t + N + 1$.

Prinsippet bak MPC er illustrert i figur 7.1. Figuren viser hvordan tidshorisonten (området i hvitt) forflytter seg fra en iterasjon til neste. Pådraget optimaliseres for hele tidshorisonten, men kun det første pådraget blir benyttet (ringet ut i gult).



Figur 7.1: Illustrasjon av MPC prinsippet

Initialverdiene til optimaliseringsproblemet er gitt av tilstandene til systemet som skal reguleres. Tilstandene estimeres eller måles direkte fra systemet, og utgjør en tilbakekobling tilbake til MPC-regulatoren. Denne tilbakekoblingen gjør at MPC-regulatoren kombinerer fordelene fra både optimalisering og regulering basert på tilbakekoblingsprinsippet. Tilbakekoblingen gjør det mulig å redusere effekten av forstyrrelser i systemet fordi nye målinger (eventuelt estimer) av systemets tilstander blir brukt som initialverdien for hver iterasjon.

Optimaliseringsproblemet er sentralt i en MPC. Neste delkapittel vil redigjøre for formuleringen av optimaliseringsproblemet som er brukt i denne oppgaven.

7.1 Optimaliseringsproblemet

Et optimaliseringsproblem har tre hovedkomponenter; en objektivfunksjon, beslutningsvariable og begrensninger. Generelt kan et optimaliseringsproblem skrives på formen:

$$\min_z f(z) \quad (7.1a)$$

$$\text{slik at } c_i(z) = 0, \quad i \in \mathcal{E} \quad (7.1b)$$

$$c_i(z) \leq 0, \quad i \in \mathcal{I} \quad (7.1c)$$

Ligning 7.1a er objektivfunksjonen som beskriver det som skal minimeres. z er beslutningsvariabelen som kan manipuleres for å oppnå en minimal verdi på objektivfunksjonen, og dermed finne den optimale løsningen. Begrensningene c_i til optimaliseringsproblemet definerer det gyldige området til z , og er gitt som ligninger og ulikheter i henholdsvis 7.1b og 7.1c.

Beslutningsvariable og slakkvariable

Beslutningsvariable er de variablene som optimaliseringsproblemet kan manipulere for å minimere kostnadsfunksjonen. I denne oppgaven kan innendørstemperaturen T og tilstanden til oppvarmingssystemet u endres, og disse variablene betraktes dermed som beslutningsvariabel.

I tillegg til beslutningsvariablene innføres en slakkvariabel T_k som beskriver en ønsket komforttemperatur. Komforttemperaturen beskrives best som et intervall, og ikke en enkelt verdi. Intervallet for komforttemperaturen blir inkludert i begrensningene. Et argument for dette valget er at en forbruker vil synes det er uvesentlig at temperaturen varierer innenfor et satt intervall dersom regulatoren kan finne den mest kostnadseffektive oppvarmingen.

Det er først dersom temperaturen beveger seg utenfor dette området at det vil gå på bekostning av komforten. Ved å sette T_k som en slakkvariabel får optimaliseringsproblemet mer frihet til å finne en optimal løsning.

Optimaliseringsproblemet skal beregne en sekvens av verdier for hver av de tre beslutningsvariablene, der sekvensen består av verdier som tilhører hver tidsinstans i tidshorisonten. De tre beslutningsvariablene er dermed i realiteten tre vektorer med beslutningsvariable, hvor n er antall tidsinstanser i tidshorisonten N :

$$\begin{aligned} u &= [u_0 \ u_1 \ \cdots \ u_{n-1}]' \\ T &= [T_1 \ T_2 \ \cdots \ T_n]' \\ T_k &= [T_{k_1} \ T_{k_2} \ \cdots \ T_{k_n}]' \end{aligned} \tag{7.2}$$

Den generelle beslutningsvariabelen z i ligning 7.1 er i denne oppgaven en sammenslåing av de tre beslutningsvariabelsekvensene u , T og T_k :

$$z = \begin{bmatrix} u \\ T \\ T_k \end{bmatrix} \tag{7.3}$$

Objektivfunksjon

Objektivfunksjonen beskriver hva som skal minimeres i optimaliseringsproblemet. Hva dette er vil være avhengig av det aktuelle problemet. Det kan være flere ulike valg av objektivfunksjon som kan gi akseptable resultater, og målet er å finne en objektivfunksjon som er tilpasset formålet på en god måte. I denne oppgaven må objektivfunksjonen i hovedsak inneholde to elementer som skal minimaliseres:

- Kostanden av strøm som blir brukt til oppvarming.

- Differansen mellom den målte temperaturen i boligen og en ønsket komforttemperatur.

Kostnaden av strømforbruket til oppvarming frem til tidshorizonten N vil være gitt av:

$$\int_{t=0}^N E u(t) p(t) dt \quad (7.4)$$

hvor E er effekten av oppvarmingssystemet, $u(t)$ er en binær variabel som indikerer om oppvarmingssystemet er av eller på, og $p(t)$ er prisen per kilowatttime (kWh). Prisen for strøm og strømforbruket på oppvarming kan antas å være konstant mellom hvert samplingsintervall, og dermed kan integrasjonstegnet erstattes med et sumtegn:

$$\sum_{t=0}^N E u(t) p(t) \Delta t \quad (7.5)$$

Det andre elementet som må være med i objektivfunksjonen er differansen mellom temperaturen i boligen T og en ønsket komforttemperatur T_k . Hva som er satt som komforttemperatur vil kunne være avhengig av om beboerne er hjemme eller ikke, eller om det er natt- eller dagtid. Summen av denne temperaturdifferansen for hver tidsinstans i tidshorizonten bør minimaliseres. Dette kan formuleres på følgende måte:

$$\sum_{t=0}^N (T(t) - T_k(t)) \quad (7.6)$$

De to elementene i objektivfunksjonen kvadreres slik at det er absoluttverdien som minimeres, og for å forsterke viktigheten av å holde differansen

liten. Ved å slå sammen ligningene 7.5 og 7.6 får vi kostfunksjonen:

$$f = \sum_{t=0}^N \left(H(T - T_k)^2 + R(E u p \Delta t)^2 \right) \quad (7.7)$$

hvor H og R er vektingsmatriser for temperaturdifferansen av henholdsvis komforttemperatur og kostnad. Ved å velge en stor H i forhold til R vil dermed optimaliseringsproblemet fokusere mer på å holde temperaturen så nærme den satte komforttemperaturen, enn å holde kostnadene lave, og vice versa.

Det er mulig å endre eller utvide objektivfunksjonen slik at den passer bedre til det faktiske systemet som styres. Et eksempel på en mulig utvidelse er å legge til et ledd som sier at det er dyrere å varme opp raskt, enn å ha en jevn oppvarming. En slik situasjon kan for eksempel være forenlig med oppvarming av varmekabler i gulv. I et slikt oppvarmingssystem vil det sjelden lønne seg å skru helt av når ingen er hjemme, for så å skru på igjen. Forholdet mellom effekt ut, i form av varme, og tilført effekt, er lavere når temperaturen må økes enn når temperaturen holdes jevn. Dette kan inkluderes i objektivfunksjonen ved å legge til et ledd som bidrar til at temperaturforandringene minimeres. Diskretisert kan temperaturforandring skrives:

$$\sum_{t=1}^N \frac{T(t) - T(t-1)}{\Delta t} \quad (7.8)$$

Også dette elementet kvadreres og får en tilhørende vektingsmatrise S . Ved å inkludere dette i ligning 7.7 kan objektivfunksjonen skrives som:

$$f = \sum_{t=0}^N \left(H(T - T_k)^2 + R(E u p \Delta t)^2 + S \left(\frac{T(t) - T(t-1)}{\Delta t} \right)^2 \right) \quad (7.9)$$

Vektingsmatrisene H , R og S må tunes i forhold til hverandre, slik at hvert ledd blir godt balansert. Ved å sette sammen disse matrisene til en matrise Q kan objektivfunksjonen skrives som følger:

$$f = z'Qz \quad (7.10)$$

Begrensninger

Likhetsbegrensningene 7.1b til optimaliseringsproblemet blir funnet ved å ta utgangspunkt i den termiske modellen beskrevet i kapittel 6. Fra ligning 6.6 og 6.7 har vi:

$$\dot{T}(t) = \frac{C_{heat}\dot{M}}{m}u(t) - \frac{K_e}{mc}(T(t) - T_{out}(t)) \quad (7.11)$$

Modellen blir diskretisert ved å bruke Eulers diskretiseringsmetode:

$$\begin{aligned} \frac{T_{i+1} - T_i}{\Delta t} &= \frac{C_{heat}\dot{M}}{m}u_i - \frac{K_e}{mc}(T_i - T_{out_i}) \\ T_{i+1} + \underbrace{\left(\frac{K_e\Delta t}{mc} - 1\right)}_{K_1}T_i - \underbrace{\frac{C_{heat}\dot{M}\Delta t}{m}}_{K_2}u_i &= \underbrace{\frac{K_e\Delta t}{mc}}_{K_3}T_{out_i} \end{aligned} \quad (7.12)$$

hvor i representerer en tidsinstans, og $i \in [0, n]$.

Ligning 7.12 representerer et ligningssett, en ligning for hver tidsinstans i tidshorisonten N . Dette ligningssettet kan skrives på matriseform:

$$A_{eq}z = B_{eq} \quad (7.13)$$

hvor z er beslutningsvariabelen definert i ligning 7.3, og matrisene A_{eq} og B_{eq} er definert som:

$$A_{eq} = \left[\begin{array}{ccccc|cccc|ccc} -K_2 & 0 & \cdots & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots & K_1 & 1 & \ddots & & \vdots & & & \\ \vdots & \ddots & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & \ddots & 0 & & & \\ 0 & \cdots & \cdots & 0 & -K_2 & 0 & \cdots & 0 & K_1 & 1 & 0 & \cdots & 0 \end{array} \right]$$

$$B_{eq} = \begin{bmatrix} K_3 T_{out_0} - K_1 T_0 \\ K_3 T_{out_1} \\ K_3 T_{out_2} \\ \vdots \\ K_3 T_{out_{n-1}} \end{bmatrix}$$

T_{out_0} og T_0 er henholdsvis ute- og innetemperaturen i startpunktet. Disse initialverdiene vil i en praktisk sammenheng bli hentet fra termometre som er plassert inne i og utenfor boligen. $T_{out_1} \cdots T_{out_{n-1}}$ er fremtidige forventede utetemperaturer som blir hentet fra OpenWeatherAPI, forklart i del 3.4.

Ulikhetsbegrensningene 7.1c defineres ved å sette et gyldig intervall for slakkvariabelen T_k . T_k beskriver et ønsket intervall for komforttemperaturen, og er begrenset av en satt øvre og nedre grense, $\overline{T_k}$ og $\underline{T_k}$. Siden T_k er en vektor med n elementer, må også $\overline{T_k}$ og $\underline{T_k}$ være vektorer med tilsvarende lengde. Hvert element i vektorene representerer en øvre og nedre grense for komforttemperaturen i en tidsinstans. Det er dermed mulig at komforttemperaturintervallet varierer i løpet av dagen, for eksempel avhengig

om beboerene er hjemme eller ikke. Intervallet for T_k settes i ulikheten:

$$\underline{T}_k \leq T_k \leq \overline{T}_k \quad (7.14)$$

Pådraget u er som nevnt tidligere en binær variabel, så en blandet heltall algoritme har blitt valgt til å løse dette. Mer om det i neste delkapittel. Blandt heltall problemer krever en lang utregningstid, og det var derfor også behov for å kunne løse optimaliseringsproblemet som et kvadratisk problem, og det er da nødvendig å sette en begrensning på u :

$$0 \leq u_i \leq 1 \quad \text{for } i = 0, 1, \dots, n - 1 \quad (7.15)$$

Ulikhetsbegrensningene 7.14 og 7.15 kan sammen skrives på matriseform:

$$Az \leq B \quad (7.16)$$

hvor z er beslutningsvariabelen definert i ligning 7.3, og matrisene A og B er definert som:

$$A = \begin{bmatrix} I & 0 & 0 \\ -I & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & -I \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \\ \overline{T}_k \\ -\underline{T}_k \end{bmatrix}$$

I er i A - matrisen en $[n \times n]$ identitetsmatrise, mens i B - matrisen representerer 1 en vektor med lengde n der alle elementene har verdien 1 , og tilsvarende for 0 - elementet.

Med begrensningene kommer fordelen ved å innføre slakkvariabelen T_k til syne. Kun slakkvariabelen blir direkte begrenset, og ikke T som er tilstanden

til systemet som skal reguleres. Dette er en fordel fordi optimaliseringsproblemet ikke vil gi noen løsning dersom tilstanden av systemet befinner seg utenfor det område som er definert som gyldig.

I praksis er det fullt mulig at temperaturen vil kunne bevege seg utenfor komforttemperaturen uten at oppvarmingssystemet vil kunne ha kontroll over det. Eksempler på dette er dersom for eksempel en dør blir stående åpen en kald vinterdag, eller at det brenner i en peis over lengre tid. Ved å innføre en slakkvariabel slik som det er gjort her, er det kun nødvendig å sette en begrensning på T_k , og ikke på T , og dermed unngå at optimaliseringsproblemet går ut av det gyldige området i slike situasjoner.

Integralvirkning

Modellen som MPC- regulatoren benytter seg av vil i realiteten alltid inkludere unøyaktigheter i en eller annen grad. Dette avviket mellom modellen som MPC- regulatoren bruker og det fysiske systemet som skal reguleres vil resultere i at regulatoren får en dårligere evne til å regulere systemet. Denne modellfeilen bør det kompenseres for, og i denne oppgaven er det valgt gjøres ved å inkludere integralvirkning i regulatoren.

Integralvirkning går ut på å integrere opp et avvik, og benytte dette til å korrigere. Ved å integrere opp avviket er det mulig å fjerne stasjonære avvik, fordi integralet av slike avvik vil øke etterhvert som tiden går, og dermed er det tydelig dersom avviket ikke kanselleres.

Integralvirkningen blir i denne oppgaven inkludert ved å se på avviket mellom estimert temperatur \hat{T} og målt temperatur i huset T .

$$\epsilon = T - \hat{T} \tag{7.17}$$

For å finne \hat{T} blir modellen i MPC- regulatoren benyttet. Regulatoren

beregner en sekvens av optimale pådrag, men også en sekvens av temperaturer som er et estimat av temperaturforløpet. Det første estimatet lagres til neste måling av temperaturen blir tilgjengelig. ϵ vil da være differansen mellom måling og estimat.

Integraleffekten oppnås ved å addere avviket fra hver tidsinstans:

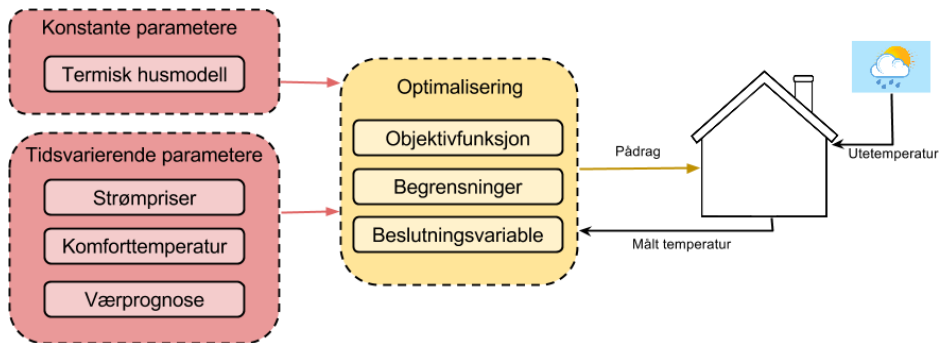
$$\zeta_t = \epsilon_t + \zeta_{t-1} \quad (7.18)$$

For å korrigere for modellfeilen blir ζ lagt til som et konstantledd i likhetsbegrensningene. Avviket som kommer etter ett samplingsintervall blir antatt å være konstant for hele tidshorisonen. Ved å inkludere integraleffekten blir dermed likhetsbegrensningene i ligning 7.13 endret:

$$A_{eq}z = B_{eq} + \zeta \quad (7.19)$$

Parametere

Parameterene i optimaliseringsproblemet blir gitt ut i fra estimerte strømpriser, værvarsel, komforttemperaturer og den termiske husmodellen. I denne oppgaven anses den termiske husmodellen å ha konstante parametre, og de øvrige parametrene er tidsvarierende. Figur 7.2 illustrerer overordnet hvordan optimaliseringsproblemet er bygget opp, og hvilke parametre som inngår. Interaksjonen mellom optimaliseringen og boligen illustrerer hvordan MPC- algoritmen opererer.



Figur 7.2: Oppbygging og bruk av optimaliseringen

7.2 Kvadratisk programmering

Objektivfunksjonen i optimaliseringsproblemet er kvadratisk, og alle likhetsbegrensningene og ulikhetsbegrensningene er lineære. Med dette er betingelsene for at optimaliseringsproblemet kan defineres som et kvadratisk problem oppfylt. I tillegg er Q i ligning 7.10 definert til å være positiv semidefinit og symmetrisk, og det medfører at optimaliseringsproblemet er konvekst. I et konvekst optimaliseringsproblem vil en løsning på problemet alltid være en global optimal løsning [33].

Optimaliseringsproblemet som er formulert i delkapittel 7.1 kan løses ved å bruke MATLAB- funksjonen `quadprog()`:

```
opts = optimset('Algorithm', 'interior-point-convex');
[q exFlag] = quadprog(Q, [], A, B, AeqQP, Beq, [], [], [], opts);
```

Kode 7.1: Løsning av optimaliseringsproblemet ved “`quadprog()`”

MATLAB beregner med dette en optimal løsning for z ved å benytte en algoritme som heter “Interior-point”. Det første elementet i z er det pådraget

som skal påtrykkes oppvarmingssystemet. Ved å se på verdiene av *exFlag* kan vi kontrollere at løsningene MATLAB finner på optimaliseringsproblemet er globale løsninger, slik som teorien tilsier.

Ved å løse optimaliseringsproblemet med “quadprog()” slik som beskrevet vil u kunne ta alle verdier i intervallet $[0, 1]$. I utgangspunktet er det ønskelig at u skal være en binær variabel, slik at det som styres er om oppvarmingssystemet er av eller på. Likevel er det interessant å se på resultatene fra den kvadratiske programmeringen. Ved å ikke legge til krav om at u må være binær får MPC-regulatoren langt større fleksibilitet og handlingsrom. Resultatene vil dermed bli mer nyanserte, og det blir lettere å forstå hvordan MPC-regulatoren fungerer.

I simuleringen som gjøres i SmartHjem systemet er det valgt å ikke legge til krav om at u skal være binær, til tross for at det ville vært mer reelt. I SmartHjem er det ønskelig at simuleringen skal være så rask som mulig for å ta hensyn til brukervennlighet, og en binær variabel vil øke beregningstiden. Simuleringen i SmartHjem har til hensikt å gi en indikasjon på temperaturforløpet og de resulterende kostnadene den nærmeste fremtiden. Simuleringer vil uansett avvike noe fra virkeligheten, blant annet fordi den baserer seg på usikre estimater av utetemperaturer og den ikke kan ta høyde for eventuelle fremtidige forstyrrelser. Det anses dermed at det er viktigere at responstiden på simuleringen er kort enn at resultatet er basert på om pådraget er binært eller ikke.

7.3 Blandet heltall programmering

I simuleringsdelen av denne oppgaven vil det være best å benytte seg av en løsningsalgoritme som krever lite beregningstid. I den faktiske reguleringen av temperaturen er ikke beregningstiden like kritisk, så lenge beregningstiden er kortere enn samplingstiden. De termiske egenskapene

til en bolig endrer seg relativt sakte, og det er valgt et samplingsintervall i systemet på 5 minutter. Det medfører at optimaliseringsproblemet kun trenger å beregne nytt pådrag hvert 5. minutt, og systemet har god tid på beregningen.

Ved den faktiske reguleringen av temperaturen kreves det i denne oppgaven at pådraget av oppvarmingssystemet u er binært, som indikerer om oppvarmingen er av eller på. Dette kravet implementeres ved å benytte MATLAB-funksjonen “`solvesdp()`” som finnes i verktøykassen *Yalmip*. Koden for hvordan dette blir implementert er vist i kode 7.2.

```
Tk = sdpvar(n,1);
T = sdpvar(n,1);
u = binvar(n,1); %Krever at u er binær

objective = [u' T' Tk']*Q*[u; T; Tk;];

con1 = [Tk_low' <= Tk <= Tk_high']; %Ulikhetsbegrensninger
con2 = [Aeq1 * T + Aeq2*u == Beq]; %Likhetsbegrensninger

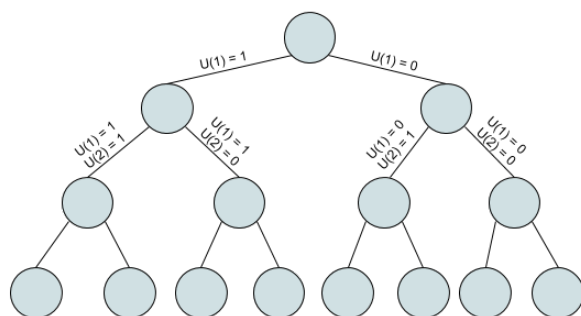
solvesdp([con1,con2],objective);
```

Kode 7.2: Løsning av optimaliseringsproblemet ved `solvesdp()`

Yalmip er et gratis verktøy som gjør det mulig å løse heltallsproblemer av ulike konvekse optimaliseringsklasser. “`solvesdp()`” benytter både “`quadprog()`” og “Branch- and Bound”-algoritmen til å løse optimaliseringsproblemet [39]. Ulempen er at denne algoritmen krever lang beregningstid. Ved store og komplekse optimaliseringsproblemer vil ikke Yalmip være tilstrekkelig, og programvarer som for eksempel CPLEX som er dedikert til heltallsproblemer bør vurderes. Yalmip velges i denne oppgaven fordi på en enkel måte illustrer behovet for heltalls-optimalisering.

Prinsippet bak “Branch and Bound” - algoritmen er å først løse problemet

uten krav om binære variabler. Deretter deles problemet opp i flere delproblemer. For hvert delproblem settes en av beslutningsreglene til å være enten 0 eller 1, og verdien på objektivfunksjonen blir beregnet. Denne algoritmen resulterer dermed i et tre av delproblemer, se figur 7.3. Algoritmen har metoder for å kunne utelukke deler av treet hvor den vet at løsningen ikke ligger, men med mange binære variabler vokser treet fort. Beregningstiden vokser eksponentielt med antall binære variabler, og algoritmen krever dermed lang beregningstid.



Figur 7.3: Illustrasjon av Branch and Bound-algoritmen

7.4 Beregningstid

Som diskutert vil beregningstiden for å løse et optimaliseringsproblem være avhengig av algoritmen som velges. “Branch and Bound” - algoritmen, som Yalmip benytter seg av, krever svært lang beregningstid. For å kunne benytte denne metoden i reguleringen til tross for den lange beregningstiden er det derfor helt nødvendig å gjøre tiltak for å redusere beregningstiden. I simuleringen i SmartHjem systemet er det også av interesse å redusere beregningstiden for økt brukeropplevelse.

Samplingsintervallet forteller hvor lang tid det er mellom hver måling. For

hver sampling løses optimaliseringsproblemet på nytt. Den totale simuleringstiden vil øke med et kort samplingsintervall fordi det vil kreve at optimaliseringsproblemet løses flere ganger. I praksis kan ikke samplingsintervallet være kortere en tiden det tar å løse optimaliseringsproblemet en gang. Et langt samplingsintervall vil også gjøre avviket grunnet modelleringsfeil større, og det blir vanskeligere å korrigere for denne feilen. Samplingsintervallet avgjør også hvor ofte pådraget kan endres, så et langt intervall gir MPC-regulatoren mindre frihet til å justere temperaturen.

Antall beslutningsvariable til optimaliseringsproblemet er også avgjørende for beregningstiden, og dette antallet kan reduseres på flere måter. I denne oppgaven er det hovedsaklig valgt å se på tre tiltak for å redusere antall variable; lengden på tidshorizonten, diskretiseringsintervallet og inngangsblokkering.

Tidshorizont

En MPC-regulator bruker modellen av systemet for å beregne en optimal pådragsekvens for den nærmeste fremtid. Tidshorizonten til MPC-regulatoren sier hvor langt fremover i tid denne sekvensen skal beregnes. MPC-regulatoren estimerer også oppførselen til systemet for hele tidshorizonten basert på informasjonen den har. Tidshorizonten forteller hvor langt inn i fremtiden MPC-regulatoren har informasjon som blir tatt høyde for, og hendelser utover tidshorizonten blir ikke med i beregningen. Ved å gi MPC-regulatoren en lang tidshorizont vil den ha mulighet til å ta høyde for endringer i temperaturkrav, prisforandringer og utetemperaturer fremover i tid, og dermed kunne reagere i tide på forandringene.

En lang tidshorizont vil også gjøre sekvensene u , T og T_k lengre, dermed vil optimaliseringsproblemet få flere beslutningsvariable. Med det vil også beregningstiden øke, så det er derfor viktig å avgjøre hvor lang tidshorizont

som faktisk er nødvendig.

Diskretiseringsintervall

I optimaliseringsproblemet er modellen til MPC - regulatoren diskretisert og inkludert i likhetsbegrensningene. Lengden på diskretiseringsintervallet avgjør hvor mange tidsinstanser det blir i tidshorisonten. Hver av sekvensene u , T og T_k er like lange som antall tidsinstanser, så på den måten er diskretiseringsintervallet avgjørende for antall beslutningsvariable.

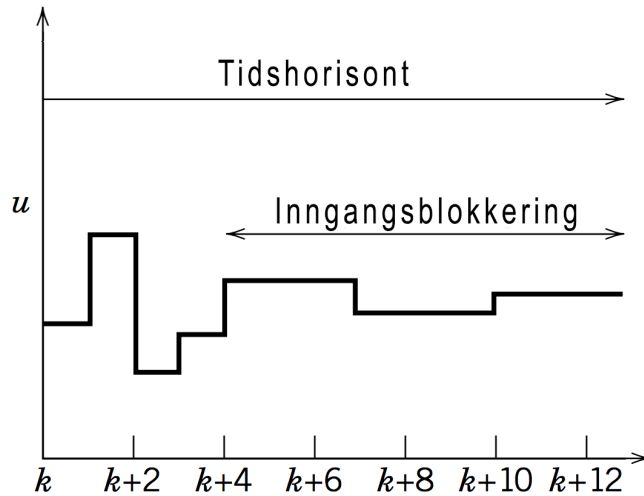
Ved å velge et kort diskretiseringsintervall vil modellen i MPC - regulatoren bli mer lik den opprinnelige kontinuerlige modellen som har blitt satt opp. Et langt diskretiseringsintervall vil derimot gjøre modellen mer unøyaktig, men samtidig medføre færre beslutningsvariable og dermed kortere beregningstid. Valget av diskretiseringsintervall er dermed en avveining mellom nøyaktighet og beregningstid.

Inngangsblokkering

Inngangsblokkering er en teknikk for å redusere antall beslutningsvariable betydelig, og dermed gi en redusert beregningstid. Denne teknikken går ut på at optimaliseringsproblemet ikke får muligheten til å endre pådraget for hver samplingsinstans i hele tidshorisonten, men definere blokker der pådraget må være konstant.

Teknikken er illustrert i figur 7.4, som har tegnet en sekvens av pådrag u som er løsningen på et optimaliseringsproblem. Pådraget endres for hver instans i starten av tidshorisonten, men fra $t = k + 4$ endres pådraget kun for hvert tredje instans. Dette gjøres ved å endre modellen til MPC- regulatoren til å ha et større diskretiseringsintervall etter en gitt tid. Begrensningen ligger kun i den beregnede sekvensen av pådrag, men som kjent vil kun

det første pådraget i sekvensen bli benyttet og MPC- regulatoren har en flytende tidshorisont. Pådraget vil dermed fremdeles kunne endres for hver samplingsinstans.



Figur 7.4: Illustrasjon av inngangsblokkering

Del IV

Avslutning

Kapittel 8

Resultater

I utviklingen av denne oppgaven har det blitt sett på flere ulike løsninger av smarte hjem, i tillegg til at det har blitt laget en prototype for det som kan være starten på utviklingen av det optimale smarte hjem for forbrukeren. Systemet som har blitt kalt SmartHjem består av muligheter for å styre elektriske artikler i hjemmet, få oversikt over forbruk og kostnad over tid, og sammenligne kostnad for innstillinger for oppvarming basert på faktisk værdata og strømpriser over tid.

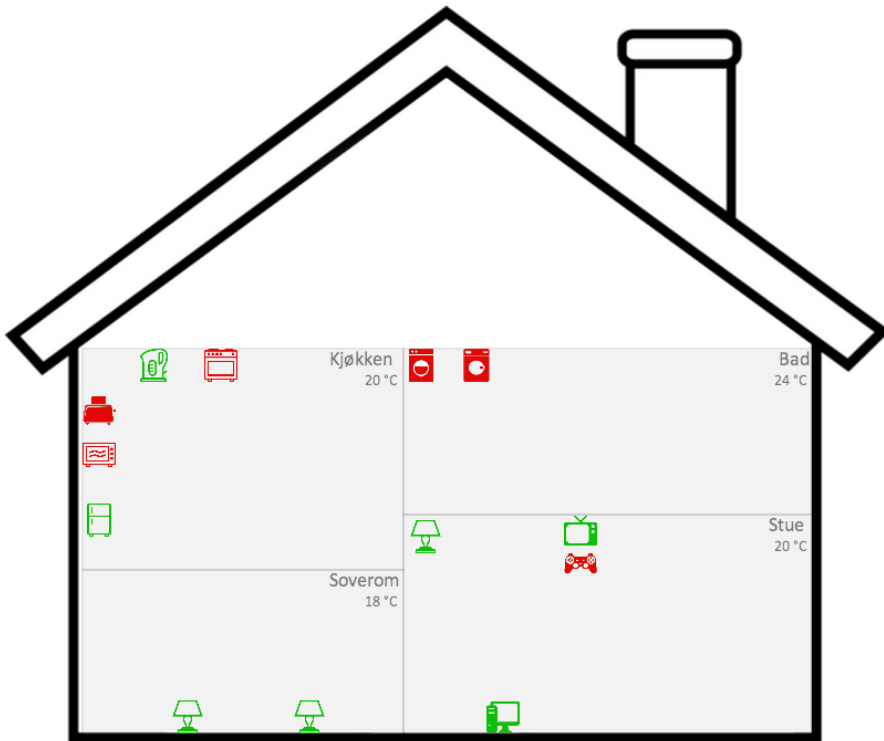
I dette kapittelet vil resultatene fra interaksjon med SmartHjem vises, samt flere resultater fra MPC regulatoren. I tillegg er det laget en film som introduserer systemet på <http://youtu.be/uROMFZXKmkI>.

8.1 Mitt hjem

Det første som møter brukeren av SmartHjem er en illustrasjon av hans eller hennes hjem, med tilhørende elektriske artikler som finnes i hjemmets forskjellige rom, vist i figur 8.1. En indikasjon på om disse artiklene er

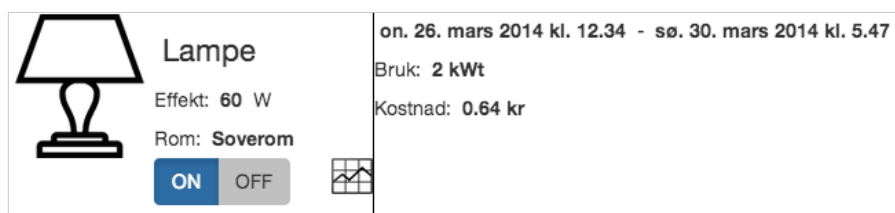
skrudd av eller på, vises i form av om enheten er henholdsvis grønn eller rød. Med dette får brukeren en oversikt over hva som finnes av elektriske enheter i hjemmet, og om hva som står på eller er skrudd av.

Temperaturen som står indikert under navnet på hvert rom er kun et statisk tall, og et forslag til hvordan det kan se ut i virkeligheten med faktiske temperaturer i rommet.



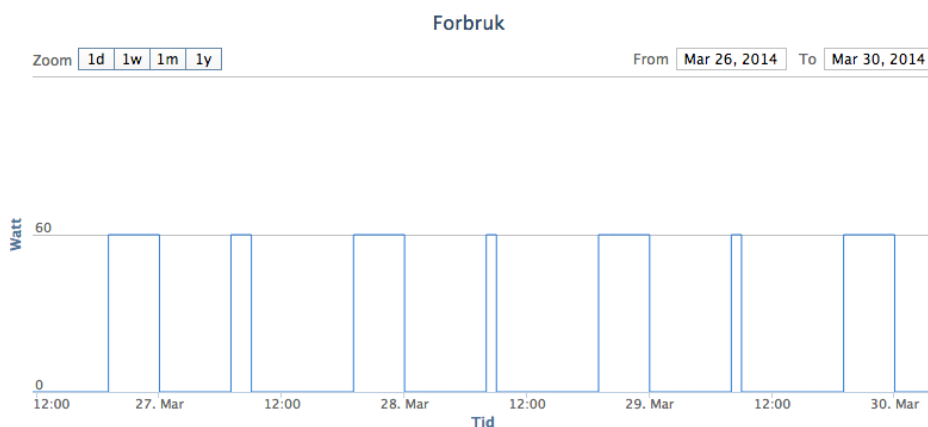
Figur 8.1: Smart Hjem-huset

For hver enhet er det mulig å få mer utdypende informasjon om historikk, lokasjon i hjemmet, effekt og kostnad over tid. Dette gjøres ved å klikke på ønsket enhet, og det vil da dukke opp en informasjonsboks tilsvarende som figur 8.2.



Figur 8.2: Lampe i Smart Hjem

Ved å så klikke videre på grafsymbolet, vil det være mulig å se og velge intervaller for historikk over når den elektriske enheten har vært i bruk, hvor mye strøm den har trukket, og hvilken kostnad dette har medført. Som grafen i figur 8.3 viser har lampen, i tidsintervallet 26.mars til 30.mars 2014 vært påslått på ettermiddagene og en liten periode hver morgen.



Figur 8.3: Forbruk av lampe i Smart Hjem

Tilsvarende for hver artikkel vil man i oversikten over *Mitt hjem* kunne se graf over det totale forbruket for et ønsket tidsintervall, samt den totale kostnaden. Tekstlig oppsummering av dette er vist i figur 8.4.

Forbruk: **325 kWt**
Kostnad: **110.54 kr**
Fra dato: **tirsdag 25. februar 2014 kl. 10.48**
Til dato: **tirsdag 25. mars 2014 kl. 9.48**

Figur 8.4: Total forbruk og kostnad i Smart Hjem

Tanken bak denne oversikten i SmartHjem er at brukeren skal kunne ha tilgang til all den informasjon han eller hun måtte ønske om sitt forbruk. Til hvilke tider av døgnet forbruket er størst og hva dette har kostet i ulike perioder.

8.2 Oppvarming i SmartHjem

En av hoveddelene i SmartHjem er knyttet til oppvarming, da dette gjerne regnes som den største delen av strømregningen. I SmartHjem er tanken at forbrukeren skal kunne simulere ulike innstillinger knyttet til oppvarming for best mulig å kunne tilfredsstille den ønskede komforten til den billigste prisen. Samtidig er hensikten å gjøre brukeren bedre informert om hvilken forskjell det gir å skru ned komforttemperaturen i hjemmet noen grader.

Måten dette er blitt gjort på er at brukeren setter opp de tidspunktene i kalenderen han eller hun regner med å være hjemme, og dette synkroniseres med Google- kalender. Deretter settes en ønsket komforttemperatur og en nedre grense for temperaturen i hjemmet når brukeren ikke er tilstede. Det kan også velges en vektning mellom pris og komfort, avhengig av om det viktigste for forbrukeren er en lavest mulig kostnad eller at komforttemperaturen blir overholdt til en hver tid når brukeren er hjemme. Antall dager det ønskes å simulere for settes, så er det klart til å gjøre simuleringen med de ønskede innstillingene. Resultatet av simuleringen vises i en graf

av temperaturforløpet i hjemmet, og de tilhørende kostnadene presenteres i en tabell.

Det skal sies at ved simulering av flere dager tar simuleringen lengre tid, og returgrafene regnes for å være mindre nøyaktig da simuleringen tar i betraktning den faktiske værprognosen for de neste dagene.

Det kan gjøres flere simuleringer med ulike innstillinger. Tabellen som viser kostnadene tilknyttet de ulike simuleringene kan brukes til å sammenligne forskjellige innstillinger og hvilken innsparing eller tap dette gir. Dataene blir også sammenlignet med de innstillingene som er aktive i huset per nå og det regnes ut en prosentvis forskjell av kostnaden. Se illustrasjon i vedlegg C.

8.3 MATLAB resultater

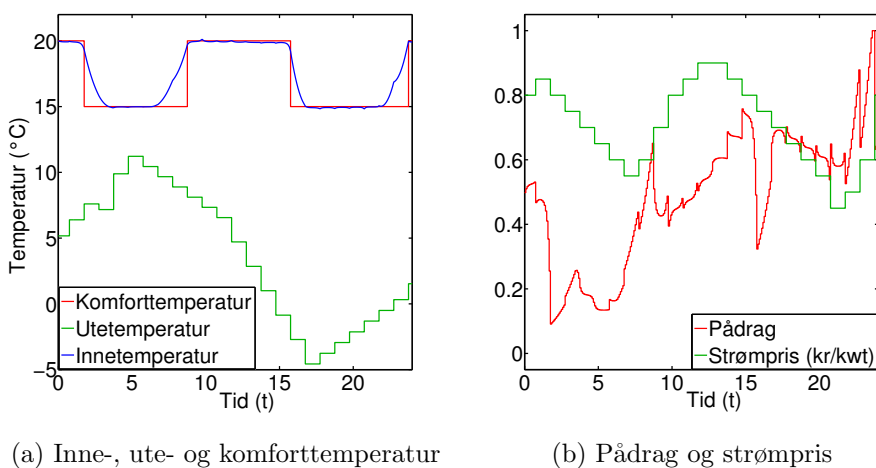
MPC-regulatoren ble implementert i MATLAB, og i dette delkapitlet vil resultatene fra MPC-regulatoren presenteres. I simuleringen i SmartHjem-systemet ble kvadratisk programmering benyttet for å løse optimaliseringen, hovedsakelig for å redusere beregningstiden og dermed øke brukervennligheten. Resultatene fra den kvadratiske programmeringen vil bli prioritert. Grunnen til det er at det er kvadratisk programmering blir benyttet i SmartHjem-systemet, og fordi de resultatene inneholder mer detaljer som kan diskuteres nærmere. Resultatene fra blandet heltall programmeringen krever svært lang tid, noe som gjør det vanskelig å få de ønskede resultatene innen rimelig tid.

Noen av resultatene vil sammenlignes mot et definert referanseresultat. Dette referanseresultatet består av pådraget som MPC-regulatoren beregner og det resulterende temperaturforløpet. For å belyse ulike aspekter vil utvalgte justeringer gjøres, og sammenlignes opp mot hverandre. Det

vil være det samme datasettet av utetemperaturer, strømpriser og brukerinnstillinger som brukes i alle simuleringene som blir presentert i dette kapittelet.

Referanseresultat

Referanseresultatet er presentert i figur 8.5. Her er det gjort en simulering av temperaturforløpet i boligen over 24 timer der temperaturen er styrt av MPC- regulatoren. Tidshorizonten i referanseresultatet er 8 timer og samplingsintervallet er satt til 5 minutter.



Figur 8.5: Referanseresultat

Figur 8.5a illustrerer temperaturforløpet i boligen (blå), komforttemperaturen satt av brukeren (rød) og utetemperaturen (grønn).

Figur 8.5b viser pådraget som er benyttet (rød), der pådraget har en verdi i intervallet $[0, 1]$. Verdien på dette pådraget forteller hvor stor andel av den maksimale effekten til oppvarmingssystemet som benyttes. Det gis altså her regulatoren muligheten å justere effekten på ovnene, slik at de

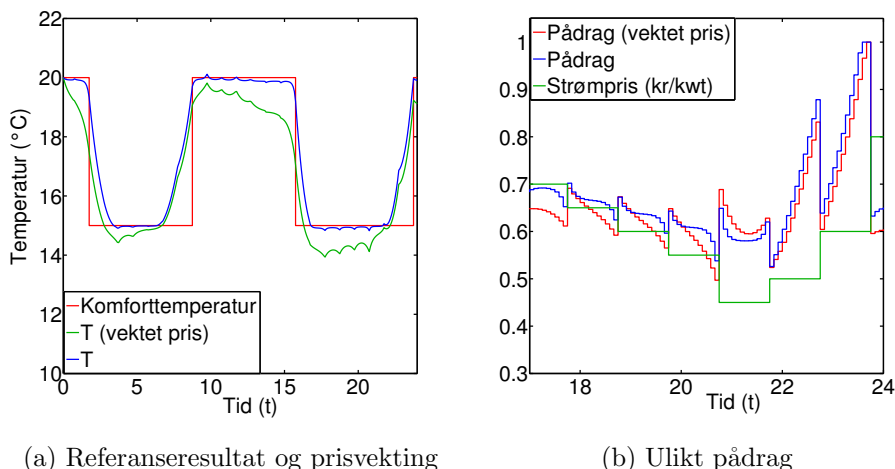
ikke kun må være enten av eller på. Prisen på strøm i kWt er tegnet i grønt.

Referanseresultatet viser at MPC- regulatoren klarer å beregne pådraget slik at den ønskede komforttemperaturen hovedsakelig blir overholdt, og regulatoren klarer å beregne hvor lang tid i forkant den trenger å starte oppvarmingen. Figur 8.5a viser et godt eksempel på at selv om utetemperaturen er på cirka 10°C lavere ved oppvarmingen 24 timer ut i simuleringen enn ved oppvarmingen etter cirka 8 timer, så klarer regulatoren å beregne pådraget som er nødvendig for å nå komfortkravet i tide.

Vekting av strømkostnad

Brukeren av SmartHjem har mulighet til å justere vektingen mellom strømkostnad og komfortkrav. Ved å vekte kostnaden vil MPC- regulatoren finne den mest økonomiske måten å varme opp boligen på, og la det gå noe ut over komfortkravet der det lønner seg mest. En høy vekt på komfort vil resultere i at temperaturen i større grad oppfyller temperaturkravene, og tillate at dette medfører noe høyere kostnader. Det er verdt å notere seg at MPC- regulatorer i begge tilfeller vil optimalisere med hensyn på kostnad, forskjellen ligger i hvilken grad pris vektlegges i forhold til komfort.

I figur 8.6 er det illustrert hva som blir resultatet av en høy vektning på reduisering av strømkostnaden. Dette resultatet blir sammenlignet med referanseresultatet, hvor vektingen er satt høyere på komfort. Utetemperaturer, strømpriser og komfortkrav er uendret.



Figur 8.6: Vekting av strømkostnad

Figur 8.6a viser temperaturforløpet som resultat av høy vekting på pris (grønn), sammenlignet med referanseresultatet (blå). Her kommer det tydelig frem at vektingen på pris medfører at det angitte komforttemperaturkravet (rød) ikke blir overholdt i like stor grad. Spesielt er forskjellen mellom temperaturene stor 12 - 15 timer ut i simuleringen. Grunnen til dette kan være at det i dette tidsrommet er utetemperatur relativt lav (se figur 8.5a), og i tillegg er strømprisen relativt høy (se figur 8.5b). Det er altså i dette området det kan lønne seg å senke temperaturen noen grader.

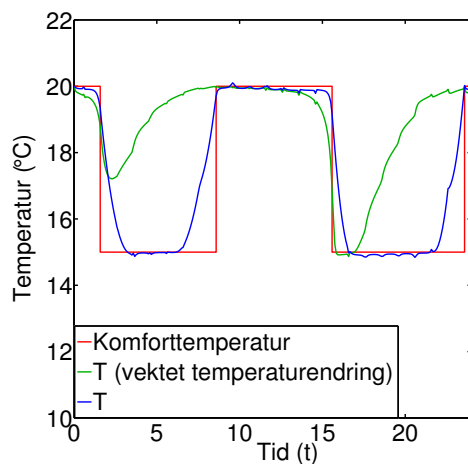
For å studere resultatet nærmere er det valgt å se på et utsnitt av pådraget, og figur 8.6b viser pådraget de siste syv timene av simuleringen. Også her sammenlignes pådraget (rød) med referansepådraget (blå). Som forventet ligger den røde grafen hovedsakelig under den blå grafen, fordi den totale kostnaden er en funksjon av den integrerte av pådraget. Mellom den 21. og 22. timen ligger derimot den røde grafen over den blå. Dette er fornuftig for det er nettopp i dette tidsintervallet at strømprisen er på sitt laveste (grønn), og dette utnytter MPC-regulatoren, spesielt når strømkostnadene

er høyt vektet.

Den totale strømkostnaden for simuleringen er også beregnet for de to simuleringene. For referanseresultatet er prisen 32,88 kr, mens for vektet pris er prisen 30,96 kr. Her ser vi at ved å sette stor vekt på å redusere strømkostnaden har MPC- regulatoren klart å redusere kostnadene med 5,8%.

Vekting av temperaturendring

Objektivfunksjonen har et ledd som kan straffe raske temperaturendringer. Ved å inkludere dette leddet vil MPC- regulatoren prøve å redusere raske temperaturendringer. Dette kan være ønskelig dersom oppvarmingssystemet fungerer slik at det koster mer å varme opp boligen rask enn en jevnere oppvarming. Figur 8.7 illustrerer effekten av en slik vekting.



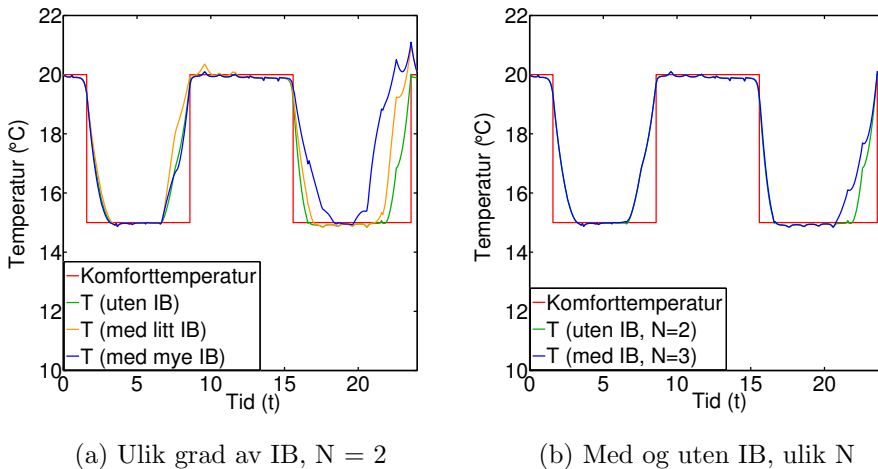
Figur 8.7: Vekting av temperaturendring

Figuren viser at ved første fraværperiode velger MPC- regulatoren å opprettholde temperaturen i boligen i større grad dersom temperaturendring

er vektet (grønn) enn om dette leddet ikke er inkludert som i referanseresultatet (blå). Ved fraværsperiode nummer to går temperaturen ned til 15°C i begge tilfellene, men ved å straffe temperaturendring vil oppvarmingen starte tidligere slik at temperaturøkningen er noe jevnere. Utetemperaturen er lavere ved andre fraværsperiode (se figur 8.5a), og dette vil være grunnen til at MPC-regulatoren velger å la temperaturen falle, siden det ville kostet mer å opprettholde temperaturen her.

Inngangsblokkering

Ved å innføre inngangsblokkering (IB) vil beregningstiden reduseres, men det vil også påvirke resultatene. I figur 8.8 vises simuleringresultatet med og uten inngangsblokkering.



Figur 8.8: Inngangsblokkering

Figur 8.8a viser hvordan resultatet påvirkes av at inngangsblokkering innføres. Ved å sette tidshorizonten $N = 2$ timer vil resultatet i dette eksempelet bli svært likt referanseresultatet når inngangsblokkering ikke er inkludert

(grønn). Inngangsblokkering innføres den siste timen i tidshorisonten, og det gjøres ved å definere blokker der inngangen ikke kan endres. Størrelsene på disse blokkene avgjør effekten av inngangsblokkering. Den oransje grafen viser resultatet av å doble blokkene i forhold til utgangspunktet, mens den blå grafen har 12 ganger så lange blokker.

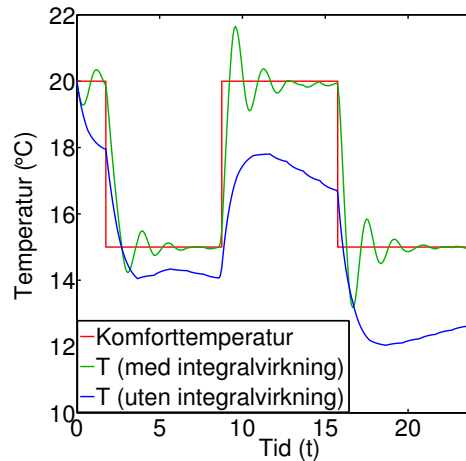
Det at inngangsblokkeringen påvirker resultatet er forventet, men figur 8.8b viser et eksempel på at inngangsblokkeringen påvirker resultatet mer enn først antatt. I denne figuren er resultatet av to simuleringer presentert; den ene simuleringen har en tidshorisont på 2 timer og ingen inngangsblokkering (grønn), den andre har en tidshorisont på 3 timer og inngangsblokkering den siste timen. Det var forventet at disse simuleringene skulle gi tilnærmet like resultater, men figuren viser at de er noe forskjell mot slutten av simuleringen. Inngangsblokkering påvirker her resultatet mer enn vi kan forklare, men det kan også være en indikasjon på at det kan finnes andre måter å implementere dette på.

Integralvirkning

I resultatene som er presentert frem til nå er den samme termiske modellen av boligen blitt brukt i MPC-regulatoren og i simuleringen, den eneste forskjellen er diskretiseringen. Det betyr at MPC-regulatoren har hatt en veldig god modell for systemet som skal reguleres, og ingen forstyrrelser tilstede. I praksis må det antas at det vil være både forstyrrelser og modelleringsfeil, og dette må regulatoren kompensere for. I denne oppgaven har det blitt gjort et forsøk på å ta høyde for dette ved å inkludere integralvirkning i MPC.

For å studere effekten av modelleringsfeil og integralvirkningen ble det lagt inn en stor modelleringsfeil i modellen til MPC-regulatoren. Arealet på boligen ble satt til 1/10 av modellen i simuleringen. Det resulterende

temperaturforløpet med og uten integralvirkning er presentert i figur 8.9.



Figur 8.9: Temperaturforløp med og uten integralvirkning

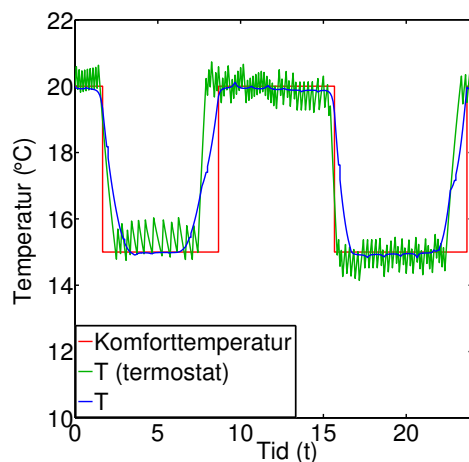
Temperaturforløpet uten integralvirkning (blå) klarer i liten grad å overholde temperaturkravene (rød), til tross for at komfort er vektet relativt høyt. Årsaken til dette er den store modelleringsfeilen som er lagt inn. Ved å inkludere integralvirkning (grønn) klarer MPC - regulatoren i stor grad å kompensere for denne feilen.

Termostat

MPC- regulatoren som er utviklet i denne oppgaven er et forslag til en oppvarmingsform som kan overta funksjonen til en klassisk termostat som finnes idag.

I figur 8.10 sammenlignes resultatet av et termostat med tidsur og referanse-resultatet. Tidsuret til termostaten er satt slik at referansetemperaturen settes til 20°C en time før brukeren har satt ønskelig komfortkrav til 20°C. Da termostaten kun har mulighet til å skru oppvarmingen av eller på, vil

det resultere i at temperaturen oscillerer mer enn referanseresultatet.



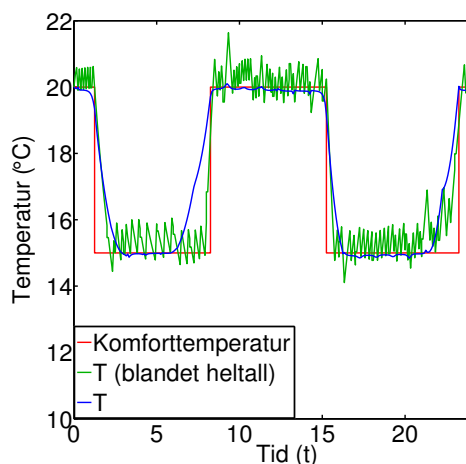
Figur 8.10: Termostat

Både referanseresultatet (blå) og resultatet av termostatstyring med tidsur (grønn), oppfyller i dette tilfellet temperaturkravet (rød) på en god måte. Den tydeligste forskjellen i dette eksempelet er temperaturforløpet ved oppvarming. Termostaten vil skru seg på en time før temperaturkravet endres fra 15°C til 20°C, uavhengig av pris på strøm, utetemperaturer eller andre faktorer, slik som MPC- regulatoren tar i betraktning. Ved den første oppvarmingen etter cirka 9 timer, vil dette medføre at temperaturen når 20°C litt før enn nødvendig. Derimot når oppvarmingen etter 23 timer så vidt komfortkravet, som følge av at det da er en lavere utetemperatur, se figur 8.5a.

Blandet heltall programmering

For å oppfylle kravet om at MPC- regulatoren kun skal kunne skrues av eller på, blir optimaliseringsproblemet formulert som et blandet heltall problem.

Algoritmen som blir brukt for å løse optimaliseringsproblemet viste seg å være svært langsom, så for å kunne få ut resultater i rimelig tid måtte en kort tidshorisont anvendes. Inngangsblokkeringen er utelatt på grunn av at det vil påvirke resultatene mer enn ønskelig når tidshorisonten er så kort ($N = 1$ time). Figur 8.11 viser resultatet av temperaturregulering med MPC basert på blandet heltall programmering, sammen med referanseresultatet.



Figur 8.11: Blandet heltall programmering

På tilsvarende måte som for termostaten blir temperaturen mer oscillerende, igjen på grunn av at oppvarmingssystemet kun kan skrues av eller på. Kravet om binært pådrag vil redusere MPC- regulatorens frihet, men likevel viser resultatet at MPC- regulatoren med blandet heltall problem og kvadratisk problem (referanseresultatet) i hovedsak gir samme resultat. Tilsynelatende ser resultatet av optimalisering av blandet heltall problem og termostat relativt like ut, men det er stor forskjell i evnen de har til å håndtere ulike hendelser.

Kapittel 9

Diskusjon

Dette kapitlet vil diskutere nærmere rundt resultatene i oppgaven. I tillegg vil det diskuteres rundt spørsmål som stadig dukket opp underveis i utviklingen om hvorfor det er aktuelt med såkalte smarte hjem. Denne delen vil derfor også forsøke å svare på spørsmål som: Hvorfor er det slik at de siste årene har det blitt mer fokus på smartere hjem? Hvordan vil dette påvirke hvordan vi lever til daglig? Og trenger vi egentlig å gjøre hjemmet smartere?



9.1 Oppvarmingregulering med MPC

Ved å lage et temperaturreguleringssystem som benytter seg av teorien om MPC- regulering blir det mulig å ta hensyn til mange elementer når en bolig skal varmes opp. I denne oppgaven er det vist hvordan MPC- regulatoren kan benytte strømpriser, utetemperaturer og komfortkrav fra brukeren for å finne den optimale måten å varme opp boligen.

En termostat kan i prinsippet kun forholde seg til en referansetemperatur, men det er også mulig å sette på et tidsur på termostaten slik at referansetemperaturen kan variere i løpet av døgnet. MPC- regulatoren har en avgjørende fordel ved at den kan ta hensyn til fremtidige hendelser, og beregne den beste oppførselen ut fra hva den vet om fremtiden. Dette gjør at regulatoren kan finne ut hvor tidlig den bør begynne oppvarmingen ut fra de gitte forholdene for å kunne nå temperaturkravene i tide.

En termostat med tidsur vil starte oppvarmingen en gitt tid før brukeren antar å komme hjem, uavhengig av utetemperatur. Dette vil kunne medføre at dersom det er varmt ute vil oppvarmingen starte unødvendig tidlig, og dermed medføre noe høyere kostnad. Motsatt vil en lav utetemperatur kunne resultere i at oppvarmingen ikke starter i tide, og brukeren kommer hjem til et kaldt hus.

Optimaliseringen i MPC- regulatoren gjør det mulig å utnytte variasjonene i strømprisene. Figur 8.6b illustrerer et eksempel der pådraget blir beregnet til å øke når strømmen er billig, noe som over tid vil spare brukeren for unødvendige strømutfgifter. Regulatoren finner den smarteste måten å bruke strømmen på. I noen boliger vil det kanskje være mer lønnsomt å holde temperaturen mer jevn i løpet av dagen, og i så fall vil en MPC- regulator kunne finne ut av det.

Som diskutert har en MPC mange fordeler i forhold til en enkel termostat

når temperaturen i en bolig skal reguleres. Tabell 9.1 oppsummerer kort de største fordelene og ulempene et reguleringsystem som benytter MPC medfører.

Fordeler	Ulemper
<ul style="list-style-type: none"> • Kan nå temperaturkrav uavhengig av utetemperaturer • Utnytter perioder med billig strøm • Gir brukeren fleksibilitet til å vekte mellom pris eller komfort • Tar hensyn til fremtidige hendelser 	<ul style="list-style-type: none"> • Mer kompleks å utvikle • Trenger informasjon om fremtidige strømpriser, utetemperaturer • Krever mer utstyr i en bolig • Må tilpasses hver enkelt bolig

Tabell 9.1: Fordeler og ulemper med oppvarming styrt av en MPC-regulator

MPC gir tydelig muligheter for en smartere måte å styre temperaturen i en bolig. Samtidig er den mer kompleks enn en termostat, så dersom det skal være en lønnsom investering er det nødvendig at den gir brukeren merkbart reduserte strømkostnader. Resultatene i denne oppgaven baserer seg på mange forenklinger, og kan ikke gi nøyaktig svar på hvor mye det er mulig å spare. Det er også viktig å huske at smart temperaturstyring ikke utelukkende har til hensikt å spare strøm, men også overholde komfortkravene til brukeren. Resultatene viser at regulatoren har en god evne til å overholde komfortkrav, uavhengig av utetemperaturer. Det er gjort tidligere arbeid med MPC-regulering av oppvarmingen i bygninger, med gode resultater. I doktorgraden, kilde [40], kan forfatteren vise til opp til

5% reduksjon av strømforbruket i et kontorbygg i München, og opp til 20% lavere strømforbruk i et kontor i Belgia ved bruk av en lignende MPC regulator. Dette er med på å bekrefte at MPC regulator til oppvarming har et stort potensiale.

9.2 Redusering av beregningstiden

Under utviklingen av MPC- regulatorer viste det seg beregningstiden til optimaliseringsproblemet var en utfordring. I SmartHjem var det nødvendig å redusere beregningstiden på simuleringen for å opprettholde brukervennligheten, og flere tiltak ble gjort for å redusere beregningstiden. Optimaliseringsalgoritmen, samplingsintervall og antall beslutningsvariable er avgjørende for hvor lang tid simuleringen tok. Reduseringen av antall beslutningsvariable ble gjort ved å øke diskretiseringsintervallet, redusere tidshorisonten og innføre inngangsblokkering.

Tidshorisonten til MPC- regulatoren forteller hvor langt frem i tid regulatoren har informasjon om. Minstekravet til tidshorisonten må være den tiden oppvarmingssystemet trenger for å kunne varme opp boligen fra nedre temperaturgrense til komforttemperatur. Ved maksimalt pådrag tar det cirka en time å varme opp boligen fra 15°C til 20°C ved normale utetemperaturer, så kortere enn det kan ikke tidshorisonten være. Referanseresultatet viser likevel at MPC- regulatoren i oppvarmingsfasen velger å fordele oppvarmingen ut over cirka to timer. Det betyr at det kan være fornuftig å gi MPC- regulatoren en tidshorisont på minst t timer slik at den har mulighet å fordele oppvarmingen over en lengre periode dersom det gir en lavere kostnad.

Ved å sette en lang tidshorisont får MPC- regulatoren større mulighet til å gjøre de valgene som er økonomiske i et mer langsiktig perspektiv. Et eksempel er at dersom regulatoren skal finne ut om det lønner seg å

senke temperaturen om natten, må den vite at temperaturen skal opp på samme nivå igjen om for eksempel 8 timer. Det kan den ikke se dersom tidshorisonten kun er 2 timer. Resultatene i denne oppgaven viser at resultatet blir svært likt om tidshorisonten settes til 8 eller 2 timer, men det er likevel ikke riktig å konkludere med at 2 timer er tilstrekkelig i alle tilfeller. Endring av utetemperaturer, strømpriser, termisk husmodell og oppvarmingssystem vil kunne vise at det er behov for en lengre tidshorisont. En lengre tidshorisont vil gi MPC- regulatoren større frihet til å finne den beste løsningen, men vil ta mye lengre tid å kalkulere. Valget av tidshorisont kan være avgjørende for hvor vellykket anvendelsen av MPC- regulatoren blir.

Inngangsblokkeringen har til hensikt å redusere beregningstiden, men vil også påvirke resultatet. Valg av tidshorisont og inngangsblokkering handler derfor om å finne en god balanse mellom beregningstid og nøyaktighet. Inngangsblokkeringen påvirker resultatet noe mer enn antatt. Det gjør det interessant å stille spørsmål om hvor vellykket det faktisk var å bruke inngangsblokkering i denne oppgaven. I så fall hvorfor egnet det seg ikke, og finnes det bedre måter å implementere inngangsblokkering på?

For å imøtekomme kravet om binært pådrag til oppvarmingssystemet ble optimaliseringsproblemet definert som et blandet heltall problem. Algoritmen som ble valgt for å løse dette optimeringsproblemet krever svært lang beregningstid. Ved å sette tidshorisonten til to timer ble ikke simuleringen ferdig i løpet av ett døgn, og det var derfor nødvendig å sette tidshorisonten så lav som en time for å få et resultat. Med kun mulighet til å velge en kort tidshorisont, er det vanskelig å illustrere alle mulighetene regulatoren har. Resultatet stemmer likevel godt overens med referanseresultatet, og det gir grunn til å tro at regulatoren har tilsvarende egenskaper og muligheter som når optimaliseringsproblemet er kvadratisk.

9.3 Livet i et smart hjem

I samfunnet i dag øker tilgjengeligheten og bruken av elektroniske enheter mer og mer fra dag til dag. Alt er koblet opp til internett, og stadig flere enheter kommuniserer med hverandre. For folk flest er det elektriske anlegget noe som ikke har forandret seg så mye de siste tiårene, og den største forandringen har nok vært overgangen fra smeltesikringer til automatsikringer.

Som nevnt tidligere i oppgaven har det blitt bestemt at alle norske husstander må installere AMS innen 2019 (se punkt 1.5). Da vil alle norske husstander være digitalt oppkoblet mot strømselskapene, og mulighetene for å utvikle nye løsninger i folks hjem åpner seg. For forbrukerne betyr dette at en ikke behøver å lese av strømmåleren, da dette gjøres automatisk. Samtidig som det kan bli generert detaljerte data om forbruket og til hvilken tid av døgnet dette fordeler seg på.

De neste 10 årene vil nok måten vi håndterer elektronikken i hjemmet ha forandre seg i stor grad. Oppvarming, som regnes som den største delen av strømutgiftene per måned, vil nok bli styrt ut i fra av om en faktisk er hjemme eller ikke. Det er ofte ingen grunn til at det holdes varmt når ingen er hjemme, og hvis det lønner seg å skru av, for så på igjen senere, så bør dette gjøres automatisk. For de som glemmer å slå av kaffetrakteren når de drar fra huset bør dette kunne varsles til mobiltelefonen i det hjemmet forlates, og kunne håndteres direkte fra telefonen.

Det å bli minnet på konkrete sparetips for eget hjem direkte på telefonen vil kunne gjøre det lettere for folk å faktisk følge opp dette i hverdagen. Ved å i tillegg inkludere spillifisering [41], som for eksempel poengliste og medaljer, kan det for folk bli morsommere å spare strøm.

9.4 Fremtiden og miljøet

I løpet av det siste århundre har behovet for levering og distribusjon av elektrisk energi økt radikalt. Produksjonen har blitt mer effektiv og det brukes i dag mange ulike kilder for å generere energi.

Strøm er ferskvare og må brukes i det den blir produsert, så derfor har det blitt slik at flere land har gått sammen om å importere og eksportere strøm seg i mellom for å utnytte strømmen best mulig når den blir produsert. Dette har krevd utbygging av store ledninger og kabler mellom land, og en kraftbørs for kjøp og salg, NordPool. Prisene på børsen varierer fra time til time og er gjerne høyeste når behovet er høyest, om morgningen og på ettermiddagene. For å kunne tilby alle mye strøm samtidig, kreves det større strømmaster og koblinger mellom kraftstasjonene. Disse såkalte monstremastene vil mange mene ødelegger naturen, og dersom det fortsetter med like store ulikheter i strømforbruket fra dag til dag, vil disse mastene etterhvert ha behov for å bli enda større monstre.

Med andre ord så blir det stadig viktigere at hjemmet kan tenke selv og styre strømmen smartere og mer økonomisk. Med smartere løsninger og billigere strømregning kan fokuset rettes mer mot forskning på ny og grønnere energi, samt skåne miljøet for større utbygginger rettet mot elektriske anlegg.



9.5 Hvem vil tjene på smartere hjem?

Per i dag finnes det en rekke ulike leverandører av strøm i Norge, og det er stor konkurranse om å ha de fleste kundene og de beste tilbudene. For forbrukerne er det vanskelig å finne ut hva som faktisk er det beste tilbudet for sitt eget hjem, og det finnes mange som ikke bryr seg med å bytte til et bedre tilbud.

Med utviklingen av smartere hjem vil det trolig dukke opp nye strømleverandører som vil kunne tilby lavere priser og bedre tilbud som følge av systemer som utnytter strømmen smartere. For forbrukerne vil det muligens bli noen kostnader knyttet til oppgradering til smartkontakter, varmeovner med kommunikasjon, men dette vil nok være en investering som vil lønne seg over tid.

Det er hovedsaklig forbrukerne som over tid vil tjene på en omforming til smartere hjem, da dette vil føre til en mye riktigere strømregning sett i forhold til det nødvendige forbruket i hjemmet. De mest innovative og nyskapende strømleverandørene vil nok også kunne tjene på smartere hjem da disse vil få flere kunder og kan holde prisene lavere enn konkurrentene.

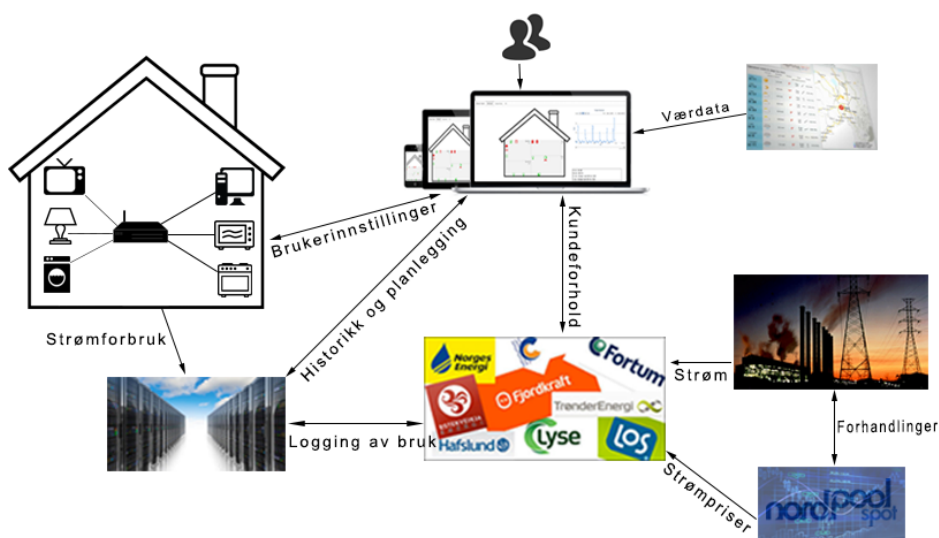
9.6 Systemet i praksis

I denne oppgaven har en prototype til et smart hjem system blitt laget. Dersom dette skulle blitt laget i praksis ville det nok vært nødvendig å gjort ting litt annerledes.

Først og fremst ville det vært viktig at en server i huset kunne ha kontroll på og kunne styre hver enkelt enhet. I tillegg til å kunne utføre beregninger knyttet til oppvarming, holde på modellen av huset, motta brukerinnstillinger og diverse informasjon som værddata og strømpriser. Serveren ville

måtte kunne sende forbruksdata tilbake til en database som også ville være tilknyttet strømleverandøren. Som vist i figur 9.1 ville det hovedsaklig være strømleverandøren, vist nederst i midten som en samling av flere selskaper, som burde ha ansvaret for organiseringen av den totale løsningen. Som for eksempel kontroll på databaser, vist ved svarte bokser i et skylandskap, håndtering av handel med NordPool og strøm levert fra kraftselskapene. I tillegg til å drifte nettsiden der forbrukerne ville kunne kontrollere hjemmet sitt.

Leverandøren ville måtte ha ansvar for å visualisere forbruksdataen til brukeren via en brukerportal á la SmartHjem, for å få oversikten over strømforbruk og kostnad. Det vil også være herfra at brukeren vil kunne få se loggdata, styre de ulike enhetene, simulere kostnad for temperaturer i hjemmet og motta værdata for optimal temperaturregulering i hjemmet.



Figur 9.1: Smart Hjem i praksis

Kapittel 10

Konklusjon

I denne oppgaven har det blitt laget et forslag til et smart hjem system som gir brukeren mulighet til å kunne styre elektriske artikler i hjemmet, samt se oversikt over forbruk og kostnad. I tillegg har det blitt laget en MPC- regulator som finner den beste måten å varme opp hjemmet på, basert på brukerens innstillinger i web systemet. Det forventede resultatet av brukerinnstillingene kan enkelt simuleres og sammenlignes i SmartHjem-systemet, og forbrukeren får se konkrete tilbakemeldinger på hva nettopp disse innstillingene vil medføre.

Denne oppgaven konkluderes med at fremtidige smartere hjem vil kunne tilby forbrukerne økt innsikt i eget strømforbruk og den tilhørende kostnaden. Det å gjøre informasjonen om strømforbruket tilgjengelig over nett vil gjøre det lettere for forbrukeren å følge med på forbruket. Et smartere hjem vil kunne gjøre valg for brukeren som reduserer strømforbruket, og i denne oppgaven blir disse valgene realisert av en MPC- regulator som styrer oppvarmingen. En kostnadseffektiv oppvarming, sammen med forbrukerens økte innsikt i strømforbruket vil kunne resultere i at det over tid vil være en positiv investering, både for økonomien og for miljøet.

Kapittel 11

Videre arbeid

I denne oppgaven har det blitt utarbeidet et forslag til et smart hjem system, der det har blitt fokusert på noen elementer slik som informasjon om forbruk, kostnad, oppvarming og styring. I dette kapittelet vil det bli sett nærmere på hva som kan gjøres videre for å gjøre systemet bedre for potensielle brukere.

11.1 SmartHjem websystemet

I dag består websystemet av muligheter for å simulere ulike innstillinger knyttet til oppvarming, skru av og på elektriske enheter, og få informasjon om forbruk per enhet og for alle enhetene. Potensielle utvidelser er mange. Et av hovedelementene i SmartHjem handler om å få bruksdata og kunne styre faktiske elektriske artikler i hjemmet. For å få til dette vil det kreve at artiklene er koblet opp mot en server, og forbruksdata lagret i en database. For å få til å styre den elektriske enheten må den programmeres til å lytte til kommandoer fra SmartHjem-systemet, se nærmere beskrivelse av dette i diskusjonsdelen 9.6.

For websystemet kunne det vært mulig å utvidet med en mer korrekt fremstilling av huset, for eksempel en 3D-modell laget på grunnlag av plantegninger for hjemmet eller ved bruk av program som RoomSketcher [42]. Dette vil gi brukeren en mer reel følelse av å styre sitt eget hjem. Dette kan videre utvides til at brukeren kan definere ulike komforttemperaturer i ulike rom etter eget ønske.

Videre kunne det vært en god idé å sørget for at systemet gav sparetips og forslag til intelligente strømvalg til forbrukeren basert på tilstedeværelse, forbruk og kostnad. Å få systemet til å logge følgende scenarier og gi passende tilbakemeldinger kunne vært interessante å sett nærmere på:

- “TV’en i kjellerstuen står på, men systemet har ikke registrert noen aktivitet i rommet den siste timen. Ønsker du å skru den av?”
- “Soverom i 1.etasje holder 20 grader, basert på utetemperatur og strømpris kan du spare 40 kroner i uken på å skru ned til 18 grader. Ønsker du å gjøre dette?”
- “Dagens sparetips: Dusjen som ble foretatt klokken 07:10 i dag tidlig varte i 5 minutter og kostet 3 kroner. Dersom du hadde dusjet i 3 minutter hadde det kostet 2 kroner, noe som utgjør 365 kroner spart i året.”

Systemet per nå er synkronisert med Google Calendar. Dette kunne også vært brukt mer, til å se på for eksempel andre hendelser og om brukeren vil være hjemme eller bortreist som følge av en annen avtale. Det skal sies at det også kan være flere som bor i husstanden og at deres kalenderoppføringer da også burde bli tatt hensyn til.

11.2 Oppvarmingsalternativer

I denne oppgaven har det først og fremst blitt sett på oppvarming av boligen med en varmeovn. En varmeovn er kanskje ikke det mest økonomiske over lengre tid, så det kunne vært mulig å sammenligne ulike typer oppvarming av boligen og vist dette for brukeren. Et eksempel kunne vært å sett nærmere på varmepumpe, hvordan den ville oppført seg til modellen av huset, til forskjell fra varmeovn. Det kunne gått an å sett på varmekabler, effekten av slike og sammenlignet kostnad.

For å kunne sammenligne ulike varmekilder på en realistisk måte må systemet ha en god modell av boligen. Da kreves det faktiske målinger og justeringer for å kunne jobbe med noe som kan gi riktigere resultater. Etter å ha opparbeidet seg en god modell for boligen vil det også kunne være aktuelt å se nærmere på varmeeffekt fra andre påskrudde elektriske artikler. En 60 watts lyspære gir også ut varme til rommet der den er skrudd på, og i tillegg avgir også mennesker i huset varme.

11.3 Smartere varmtvannsbereder

Oppvarming av vann i en bolig gjøres i varmtvannsberederen. Denne oppvarmingen skjer ved at kaldt vann fylles på tanken etter hvert som varmt vann brukes. Deretter sørger en termostat i tanken for at vannet blir varmet opp til rundt 75-80 grader. Ikke lavere, da det kan føre til utvikling av legionellabakterier i tanken. Varmt vannet som så kommer ut av springen er kjølt ned til rundt 55 grader for å hindre skolding [43].

Grunnen til at temaet om varmtvannstank blir nevnt er fordi oppvarmingen av vannet i tanken skjer på en lite økonomisk måte for forbrukerne. Folk flest dusjer gjerne om morgningen, og da er strømmen ofte dyrere. Deretter

forlater de fleste hjemmet for å dra på jobb eller skole, og boligen står gjerne tom i flere timer. Det er med andre ord ingen grunn til at vannet i varmtvannsberederen trenger å varmes opp med en gang det har blitt brukt. Dersom dette hadde blitt gjort over en lengre periode i løpet av dagen, forutsatt at det ikke er behov for varmtvannet, kunne det medført en besparelse ved at oppvarmingen foregår på tidspunkt der strømmen antatt er billigere enn mellom klokken 7 og 9 om morgningen.

11.4 Termisk modell av boligen

Simuleringene som er gjort i denne oppgaven bruker en termisk modell av boligen, og som vi har sett baserer denne modellen seg på flere forenklinger. For å kunne få mer nøyaktige resultater er det derfor nødvendig å utvikle en modell som er mer lik virkeligheten. En bedre modell kan oppnås ved å blant annet inkludere de elementene som er utelatt i modellen i denne oppgaven. Estimering av parametrene i modellen kan også gjøres ved å benytte eksperimentelle data.

Modellen kan også utvides til å ha ulike rom i boligen, tilsvarende til plantegningen. Dersom plantegningen er tilgjengelig i websystemet kan informasjonen om romdelingen hentes derfra.

En bedre modell er nødvendig i MPC-regulatoren, for den bruker modellen til å beregne det mest optimale pådraget til oppvarmingssystemet. I denne oppgaven ble det inkludert en integralvirkning i MPC for å korrigere for modelleringsfeil, og det er mulig å se nærmere på om effekten av integralvirkningen kan forbedres ytterligere.

Når en realistisk modell er på plass vil det være mulig å kunne finne mer konkrete resultater av hvor mye det er mulig å spare, og om MPC-styring av temperaturen i hjemmet kan være en lønnsom investering.

11.5 Optimaliseringsproblemet

Optimaliseringsproblemet som ble formulert i MPC- regulatoren i denne oppgaven er et eksempel på hvordan det kan gjøres. Det er fullt mulig at det finnes andre formuleringer som vil gi bedre resultater.

Det er også mulig å inkludere flere elementer i optimaliseringen dersom det er ønskelig. Som nevnt er det for eksempel mulig å inkludere oppvarming av varmtvann i optimaliseringen, eller gi brukeren mulighet til å ha ulike temperatursoner i huset.

Når det gjelder selve løsning av optimaliseringen er det behov for å se på andre algoritmer som er mer effektive, spesielt med tanke på optimalisering av blandet heltall problemet. Dette er nødvendig for å kunne analysere forskjellige resultater nærmere.

Bibliografi

- [1] “Vannkraft og elektrisitet i historisk perspektiv.” <http://www.energifakta.no/documents/Miljo%20og%20velferd/Samfunn/Historie.htm>. Sist besøkt 13. februar 2014.
- [2] “Energiloven.” <http://lovdata.no/dokument/NL/lov/1990-06-29-50>. Sist besøkt 13. februar 2014.
- [3] “Produksjon av elektrisitet.” http://www.regjeringen.no/nb/dep/oed/tema/energi_og_vannsressurser/produksjon-av-elektrisitet.html?id=440487. Sist besøkt 20. februar 2014.
- [4] Globalis, “Strømproduksjon.” [http://www.globalis.no/Land/Norge/\(show\)/indicators/\(indicator\)/495](http://www.globalis.no/Land/Norge/(show)/indicators/(indicator)/495). Sist besøkt 4. april 2014.
- [5] “Forbrukerinformasjon om strøm.” <http://www.nve.no/no/kraftmarked/forbrukersider/>. Sist besøkt 17. februar 2014.
- [6] “Elspot prices.” <http://www.nordpoolspot.com/Market-data1/Elspot/Area-Prices/NO/Hourly/>. Sist besøkt 27. februar 2014.
- [7] “Ny kunnskap om fordeling av strømforbruket.” <http://www.sintef.no/SINTEF-Energi-AS/Xergi/Xergi-2008/Nr-2---september/>

- Ny-kunnskap-om-fordeling-av-stromforbruket/. Sist besøkt 25. april 2014.
- [8] “AMS.” <http://www.nve.no/no/kraftmarked/sluttbrukermarkedet/ams/>. Sist besøkt 19. februar 2014.
- [9] Kjell Sand, “AMS og Smart Grids,” Sist besøkt 4. mars 2014.
- [10] “Om Smart Grid.” <http://smartgrids.no/senteret/about-smartgrid/>. Sist besøkt 4. mars 2014.
- [11] “Gjør hjemmet ditt smartere.” <http://meetviva.com/no/>. Sist besøkt 14. januar 2014.
- [12] “Smartly gjør boligen din smartere. Og den gjør deg smartere.” <http://www.smartly.no/>. Sist besøkt 14. januar 2014.
- [13] “Løsninger for det smarte hjem.” <http://www.verisure.no/>. Sist besøkt 16. januar 2014.
- [14] “Meet the Nest thermostat.” <http://www.nest.com/>. Sist besøkt 16. januar 2014.
- [15] “Personal wireless lighting.” <http://www.meethue.com/>. Sist besøkt 16. januar 2014.
- [16] “WeMo Home Automation.” <http://www.belkin.com/us/Products/home-automation/c/wemo-home-automation/>. Sist besøkt 17. januar 2014.
- [17] “Smart Home awesome on your smart phone.” <http://revolv.com/>. Sist besøkt 20. januar 2014.
- [18] “Elektriske apparater.” <http://www.haugaland-kraft.no/spar-stroem/elektriske-apparater-article716-578.html>. Sist besøkt 31. mars 2014.

-
- [19] Hanne Sæle, “Strømforbruk i husholdningene.” <http://www.sintef.no/SINTEF-Energi-AS/Xergi/Xergi-2010/Nr-1---juni1/Engelskmann-for-ett-ar-2211/>. Sist besøkt 31. mars 2014.
- [20] Torgeir Ericson og Bente Halvorsen, “Hvordan varierer timeforbruket av strøm i ulike sektorer?,” *Økonomiske analyser*, 2008.
- [21] “Nord Pool Spot.” <http://www.nordpoolspot.com/>. Sist besøkt 3. februar 2014.
- [22] “OpenWeatherMap - free weather data and forecast API.” <http://openweathermap.org/>. Sist besøkt 3. mars 2014.
- [23] ISO/IEC, “Software engineering — software product quality requirements and evaluation (square) — requirements for quality of commercial off- the-shelf (cots) software product and instructions for testing,” *Computer*, 2012.
- [24] Maja Charlotte Otnes og Carl Joachim Rørvik, “Galaxy Battle: Et multiplayer HTML5 spill styrt med gyroskop i mobiltelefoner over en WebSockets kobling,” 2013.
- [25] “Sleek, intuitive, and powerful front-end framework for faster and easier web development.” <http://getbootstrap.com/2.3.2/>. Sist besøkt 19. mars 2014.
- [26] “Interactive JavaScript charts for your web projects.” <http://www.highcharts.com/>. Sist besøkt 20. mars 2014.
- [27] “Accelerating the pace of engineering and science.” <http://www.mathworks.se/products/matlab/>. Sist besøkt 15. mars 2014.
- [28] MathWorks, “Simulink.” <http://www.mathworks.se/products/simulink/>. Sist besøkt 5. april 2014.

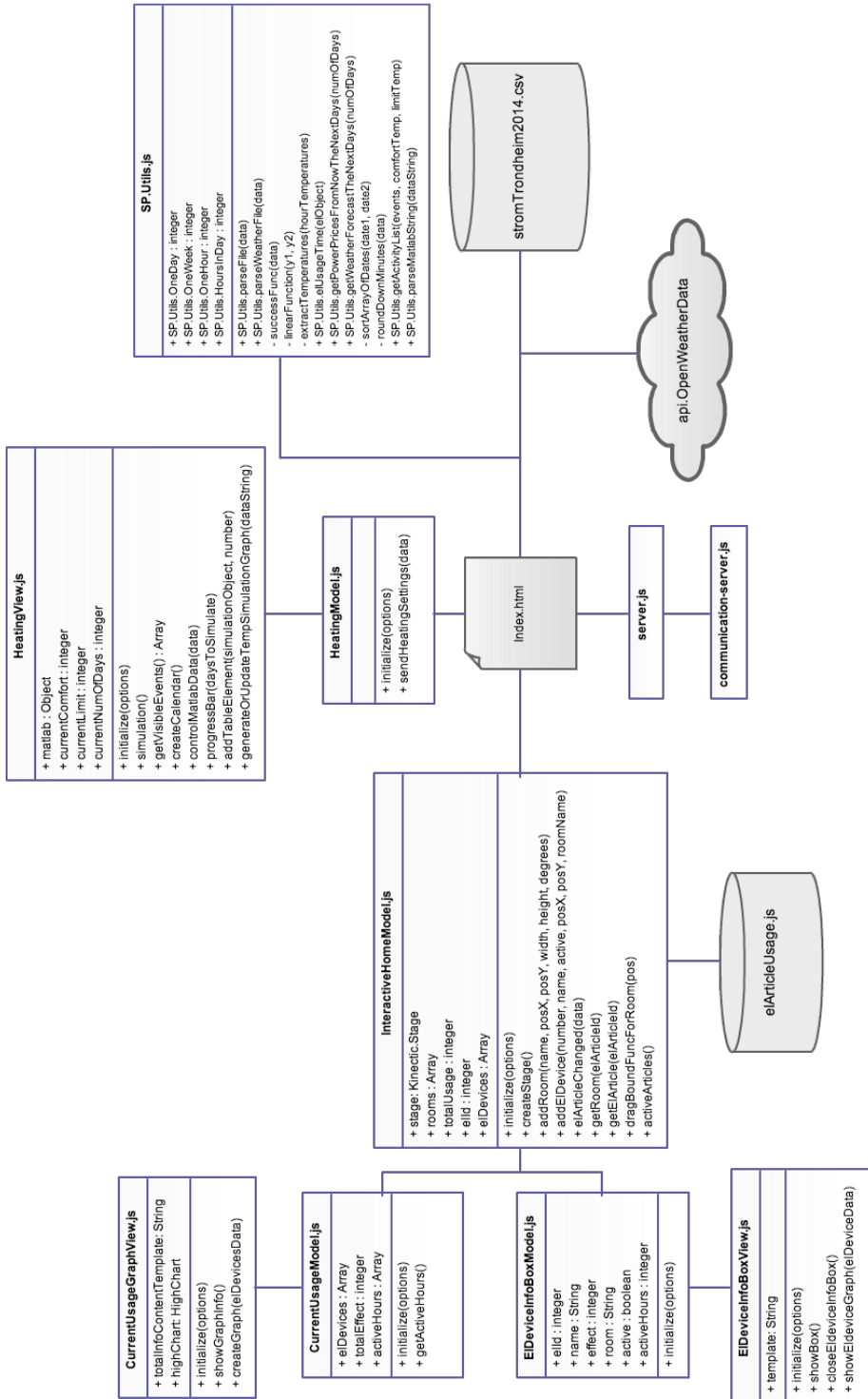
- [29] “Enterprise class interactive web graphics.” <http://www.kineticjs.com/>. Sist besøkt 20. mars 2014.
- [30] “Control and communicate with test and measurement instruments.” <http://www.mathworks.se/products/instrument/index.html>. Sist besøkt 1. april 2014.
- [31] Steven Kolak, “MATLAB TCP/IP - code example.” <http://www.mathworks.se/matlabcentral/fileexchange/11802-matlab-tcpip-code-example>. Sist besøkt 1. april 2014.
- [32] Joel Feenstra, “JSON Parser.” <http://www.mathworks.com/matlabcentral/fileexchange/20565-json-parser>. Sist besøkt 1. april 2014.
- [33] Bjarne Foss og Tor Aksel N. Heirung, “Merging Optimization and Control,” 18 Februar, 2014.
- [34] H. D. Y. og Roger A. Freedman, *University Physics with modern physics*. Pearson Education, 13 ed., 2012.
- [35] MathWorks, “Thermal Model of a House.” <http://www.mathworks.se/help/simulink/examples/thermal-model-of-a-house.html>. Sist besøkt 2. mars 2014.
- [36] SSB, “Boforhold, levekårsundersøkelsen, 2012.” <http://www.ssb.no/bygg-bolig-og-eiendom/statistikker/bo/hvert-3-aar/2013-02-05#content>. Sist besøkt 17. mars 2014.
- [37] COWI AS, “Romoppvarming av boliger. Prinsipper og systemer.” <http://bks.byggforsk.no/DocumentView.aspx?documentId=518§ionId=2>. Sist besøkt 29. mars 2014.
- [38] Bellona, Eniro, “Miljøfakta.” <http://miljofakta.no/privatpersoner/bolig/styringssystemer-text/?sourceFlash=1§ionFlash=4>. Sist besøkt 1. april 2014.

-
- [39] “What is Yalmip.” <http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.WhatIsYALMIP>. Sist besøkt 22. april 2014.
- [40] J. Cigler, *Model Predictive Control for Buildings*. PhD thesis, Czech Technical University in Prague, 2013.
- [41] “Spillifisering.” <http://no.wikipedia.org/wiki/Spillifisering>. Sist besøkt 4. april 2014.
- [42] “RoomSketcher.” <http://www.roomsketcher.no/funksjonalitet/oversikt/>. Sist besøkt 10. april 2014.
- [43] “Stor test av varmtvannsberedere.” <http://www.forbrukerradet.no/annet/tester-og-kj%C3%B8petips/produkter/stor-test-av-varmtvannsberedere>. Sist besøkt 10. april 2014.

Vedlegg A

Detaljert klassediagram

På neste side vises et detaljert klassediagram som er brukt i websystemet. Hver del er forklart nærmere i del 5.2, så dette er bare en oversikt. Det kunne vært flere koblinger mellom klassene, da de fleste kan kommunisere via Simple events (også forklart i del 5.2), men dette diagrammet viser mer en oversikt over koblingene til hvor hver klasse først blir initialisert.



Vedlegg B

Hvordan komme i gang med systemet

Dette systemet er programmert og testet på Mac OSX 10.9 og Windows 7. Det bør fungerer fint også på andre systemer, men det er vanskelig å gi en garanti for det. For å kunne kjøre systemet slik det fungerer nå, er det noen ting som først må lastes ned og installeres. Det gjøres på følgende måte:

1. Først og fremst må koden være hentet frem. Dette kan enten gjøres ved å kopiere fra den medfølgende CD'en, eller lastes ned fra GitHub ved å navigere seg frem til linken:
`http://github.com/cjrorvik/smarthjem`
2. Sørg deretter for å ha lastet ned og innstalert node.js. Dette kan gjøres via `http://nodejs.org/`
3. Etter å ha lastet ned Node, må Express og Socket.IO lastes ned. Dette gjøres på følgende måte:

- (a) **Windows:** Start Node.js kommandovindu, navigér til mappen der Smart Hjem koden ligger.
 - (b) **Mac:** Etter å ha lastet ned Node til Mac, kan Macs innebygde terminalvindu brukes. Navigér til mappen der Smart Hjem koden ligger.
4. Når kommandovinduet er åpnet og det er navigert til riktig plassering, kjøres kommandoen:

```
npm install express
```

for deretter å kjøre kommandoen

```
npm install socket.io
```

5. I et terminalvindu, naviger til filen *communication_server.js*, og kjør kommandoen:

```
node communication_server.js
```

6. Start så et nytt kommandovindu, naviger til filen *server.js*, og kjør kommandoen:

```
node server.js
```

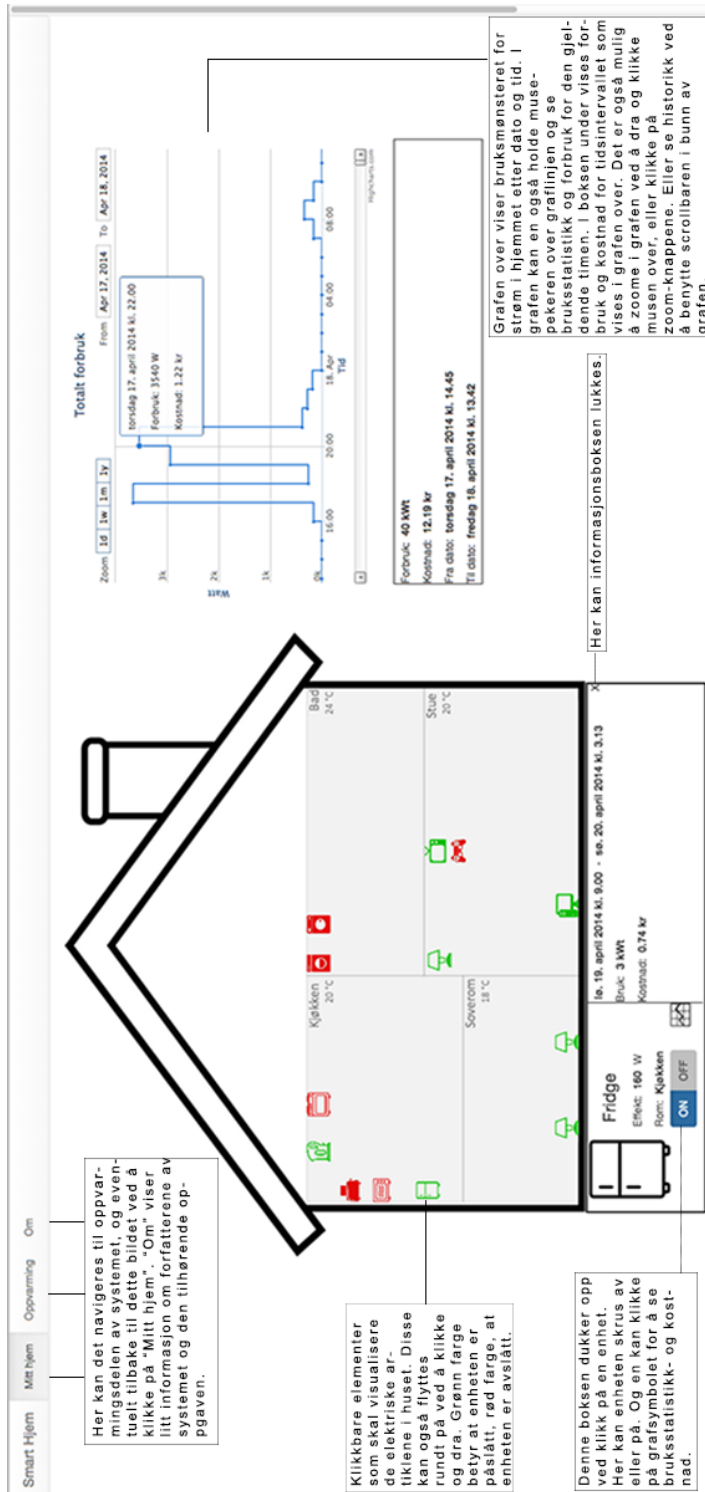
7. Start matlab og kjør filen *smart_hjem.m*
8. Åpne nettleseren og skriv inn adressen `http://localhost:8080/`. Smart Hjem skal da vises.
9. Klikk videre inn på oppvarming og prøv å simulér med ulike innstillinger. Dersom ingen data vises i grafen kan det være lurt å starte matlabfilen om igjen, samt relaste nettsiden og prøve innstillingene på nytt. Sørg også for at både *communication_server.js* og *server.js* fortsatt kjører og ikke viser noen feilmelding.

Vedlegg C

Hvordan bruke systemet

I systemet som har blitt laget i denne oppgaven finnes det flere funksjonaliteter. Disse vil bli forklart nærmere i dette vedlegget, som en slags brukermanual. Dette vil forhåpentligvis gjøre det enklere for å få en oversikt over hva som har blitt laget, samt gjøre det lettere å teste det ut.

Det er også laget en film av hvordan systemet fungerer. Den kan sees på <http://youtu.be/uROMFZXKkI>.



Smart Hjem
Mitt hjem
Oppvarming
Om

1 – 7 Mai 2014

< >
Idag
< >

	Tor 1/5	Fre 2/5	Lør 3/5	Søn 4/5	Man 5/5	Tir 6/5	Ons 7/5
01:00							
02:00							
03:00							
04:00							
05:00							
06:00							
07:00	Hjemme	Hjemme			Hjemme	Hjemme	Hjemme
08:00	Hjemme	Hjemme			Hjemme	Hjemme	Hjemme
09:00							
10:00							
11:00							
12:00							
13:00							
14:00							
15:00							
16:00	Hjemme	Hjemme			Hjemme	Hjemme	Hjemme
17:00	Hjemme	Hjemme			Hjemme	Hjemme	Hjemme
18:00							
19:00							
20:00							
21:00							
22:00							
23:00							

Komforttemperatur Her settes ønsket komforttemperatur det ønskes å simuleres for

Nedre grense Her settes nedre temperaturgrense

Antall dager Her settes antall dager det ønskes å simuleres for. Jo flere dager jo mere usikre værprognoser og strømpriser, og lengre simuleringstid.

Pris vs komfort Her settes vektningen mellom pris og komfort. Høyere verdi på komfort kan gjøre det noe dyrere da systemet vil jobbe hardere for å opprettholde nøyaktig ønsket komforttemperatur. Motsatt for prisvektningen.

Simuler Her startes simuleringen

I denne kalenderen settes det når brukeren av systemet har tenkt til å være hjemme, og våkne. Det er dette komforttemperaturen tar utgangspunkt i. Om natten antas det at temperaturen kan senkes til nedre grense. Kalenderen er også synkronisert opp mot Google Calendar, slik at kalenderen kan modifiseres fra mobil, nettbrett og skrivebordsmaskiner.

Simulér



Her er de ulike grafene for temperaturendringer i hjemmet basert på reelle værdata, strømpriser og innstillinger satt av brukeren.

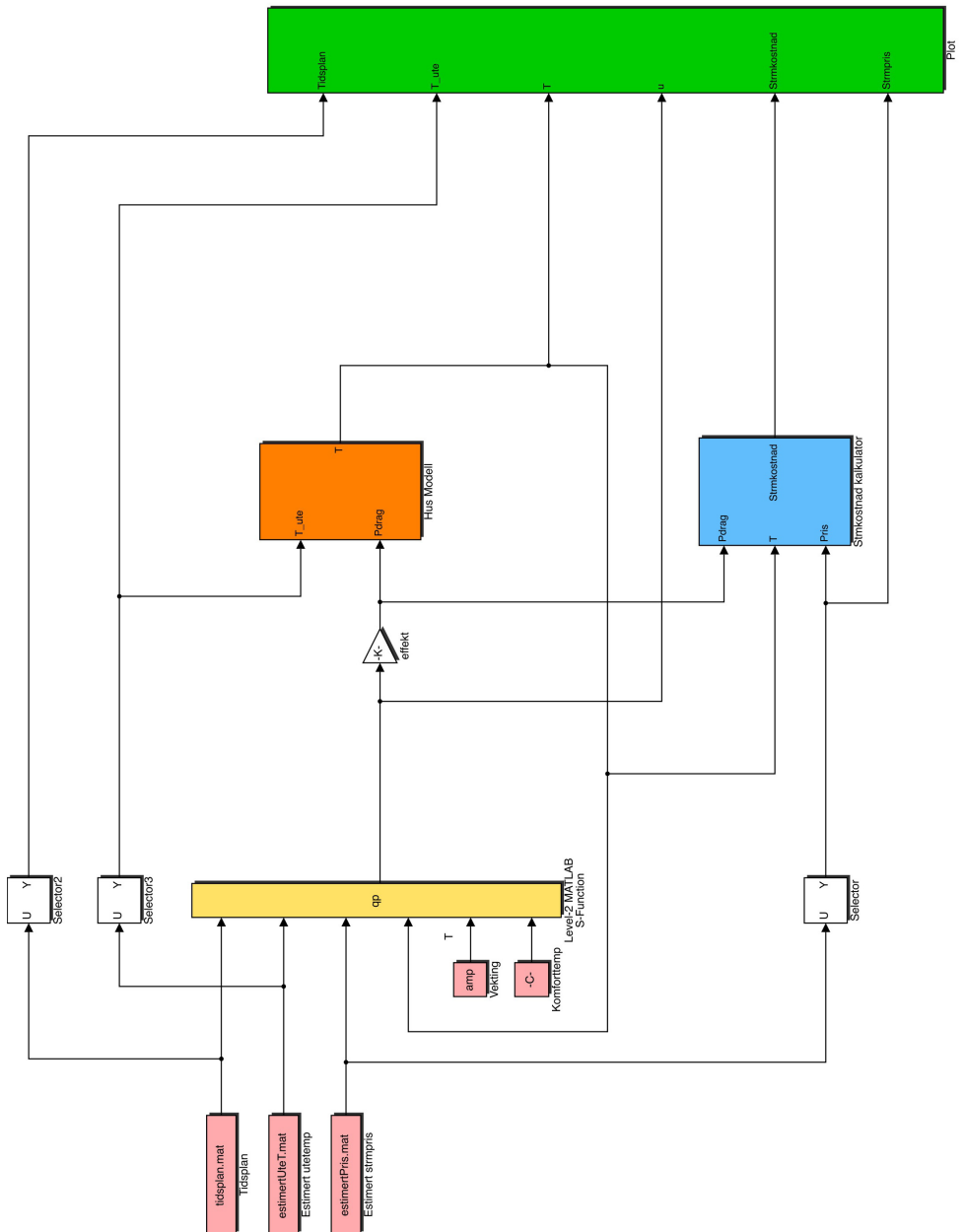
Nr	Komforttemperatur	Nedre grense	Dager	Nå	Gj. kostnad per dag	Besparelse	Aktiver innstillinger
-	20	15		Nå	13 kr	-	-
1	22	15	1	14,2 kr	-9.2 %	<input type="button" value="Ok"/>	
2	19	15	1	12,4 kr	4.6 %	<input type="button" value="Ok"/>	
3	18	16	1	12,3 kr	5.4 %	<input type="button" value="Ok"/>	
4	23	18	1	16,2 kr	-24.6 %	<input type="button" value="Ok"/>	

Her er liste over de ulike innstillingene med tilhørende resultater. Ved å klikke Ok. Settes ønsket temperaturinnstillinger i hjemmet.

Vedlegg D

Simulink

Vedlagt ligger oversikten over implementasjonen i Simulink. For å studere hver av blokkene nærmere kan filen, MPC.slx, åpnes fra den medfølgende CD'en.



Vedlegg E

CD

Det medfølger en CD til denne oppgaven. På denne vil følgende ligge:

1. Introduksjonsfilm av systemet.
2. Kildekode for både MATLAB og websystemet, SmartHjem.
3. Alle vedlegg.
4. Diverse kilder.
5. Digital versjon av denne oppgaven.