



TECHNISCHE UNIVERSITÄT BERLIN

Fachgebiet Energieversorgungsnetz und Integration
Erneuerbarer Energien

STUDIENARBEIT

Objektorientierte Implementierung und Simulation einer Kältelast mit Kältespeicher im Energieversorgungsnetz

vorgelegt von Juri STEBLAU
Matr.-Nr: 300244

8. August 2011

Korrektoren:

Advisor: Dipl.-Ing. Felix KLEIN - (TU-Berlin)
Examinator: Prof.Dr.-Ing. Kai STRUNZ - (TU-Berlin)

Realisation und Simulation einer Kältelast mit Kältespeicher im Energieversorgungsnetz

Abstract: Hier bitte am Ende der Arbeit auf english die kurze Inhaltsangabe schreiben.

Hier muss noch ein Abstract rein !!!AM ENDE!!!

Keywords: Möglicherweise in Englisch! Fuck!

Todo list

| | |
|---|----|
| Abstract nicht da | i |
| Einleitung mehrfach durchlesen! | 1 |
| Reicht so? Guck bitte nach oben. | 1 |
| Bewertung einfügen. | 1 |
| Der Satz ist scheiße, umschreiben. | 1 |
| Das am Ende zur Ende schreiben | 1 |
| wo ist die Quelle? | 2 |
| Eine Überleitung finden | 3 |
| ab hier noch nicht umformuliert | 11 |
| die nachfolgende Gleichung gehört nicht dahin! oder? | 13 |
| hier ein Überleitungssatz | 16 |
| stehen oder steht? | 17 |
| stehen oder steht? | 17 |
| stehen oder steht? | 19 |
| stehen oder steht? | 19 |
| Warum ist es nötig, die Funktion expliziert zu schreiben? | 19 |
| was ist das heute? | 22 |
| das ist scheiße! | 27 |
| freaky sentence | 28 |
| Hier auf Kapitel verweisen. | 28 |
| Kältemaschinen beschreiben? | 29 |
| $i \times 1$ eigentlich noch nicht ideal, Schwachstelle, umdenken | 29 |
| Beschränkungen hier? | 29 |
| das ist doch überflüssig, oder? | 30 |
| Wo wird das Verwendet? Hier schreiben | 30 |
| Klären was das ist. | 32 |
| Das ist noch nicht richtig im Programm implementiert. Das muss DU NOCH MACHEN! | 32 |
| Funktion mit für die graphische Ausgabe schreiben | 32 |
| Funktion für tabellarische Ausgabe schreiben | 32 |

Inhaltsverzeichnis

| | |
|---|-----------|
| Abbildungsverzeichnis | iv |
| Tabellenverzeichnis | v |
| MATLAB® -Code Verzeichnis | vi |
| Abkürzungsverzeichnes | vii |
| 1 Einleitung | 1 |
| 2 Theorie | 2 |
| 2.1 Erneuerbare Energien im Energieversorgungsnetz | 2 |
| 2.2 Kälteerzeugung im Supermarkt: Potentiale und Besonderheiten . . . | 3 |
| 2.3 Objektorientierte Programmierung mit MATLAB® | 4 |
| 2.4 Einführung in die Kältetechnik | 6 |
| 3 Aufgabenstellung | 14 |
| 4 Modellierung | 15 |
| 4.1 Erstellung des OOP-Modells | 15 |
| 4.2 Handhabung des Programms | 25 |
| A Anhang | 33 |
| A.1 Anhang Example section | 33 |
| Literaturverzeichnis | 34 |

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 2.1 | Prinzipdarstellung Kompressionskälteanlage | 8 |
| 4.1 | Vier Knoten Beispiel | 15 |
| 4.2 | Klassendiagramm Modellkonstrukt | 16 |
| 4.3 | Sequenzdiagramm Modellkonstrukt | 21 |
| 4.4 | Klassendiagramm Kooperationskonzept | 24 |
| 4.5 | Flussdiagramm | 26 |
| 4.6 | Sequenzdiagramm Kooperationsstrategie | 27 |

Tabellenverzeichnis

MATLAB[®] -Code Verzeichnis

| | | |
|-----|--------------------------------------|----|
| 2.1 | Beispiel Klassendefinition | 6 |
| 4.1 | config_grid.m | 28 |
| 4.2 | config_supermarkets.m | 29 |
| 4.3 | config_fridges.m | 30 |

Abkürzungsverzeichnis

| | |
|---------------|---|
| NK_KR_V..... | normalgekühltes Kühltregal an Verbundanlage |
| NK_KT_S..... | normalgekühlte Kühltruhe steckerfertig |
| OOP | objektorientierte Programmierung |
| TK_TKT_S..... | tiefgekühlte Tiefkühltruhe steckerfertig |

KAPITEL 1

Einleitung

Die Erforschung der Ursachen und der Folgen des Klimawandels, die wachsende Schwierigkeit bei der Bereitstellung der konventionellen Energien, die Neubewertung der Risiken und technischen Mitteln bei der Endlagerung von Abfällen der Atomindustrie, die besorgniserregende Erkenntnis der bisherigen Fehlbewertung der Atom-sicherheit **werden mit Gewissheit** die politisch beschlossene Förderung der erneuerbaren Energien zu einer grundlegenden Energieform in den kommenden Jahren in Deutschland und in Europa forcieren.

Einleitung
mehrfach
durchlesen!

Reicht so?
Guck bitte
nach oben.

Der Umstieg auf alternative Energien ist mit einigen Problemen verbunden. An einigen Orten ist der Einsatz dieser Technik aus politischer, technischer, ökonomischer oder ökologischer Sicht nicht möglich. Darum weichen oft die Stromerzeugung und der Strombedarf zeitlich und räumlich voneinander ab. Windkraft im Meer, Wasserkraft in den Bergen, Sonnenkraft in dem Süden, Geothermie in Island, Biomasse Land. Es ist auch eine Tatsache, dass nur ein Teil der erneuerbaren Energien direkt vom Menschen beeinflusst werden kann. Besonders die Menge der durch Sonne und Wind gewonnenen Energie schwankt abhängig von der Wetterlage. Die Integration dieser Energie in Netz führt zur erhöhten Bereitstellung an Regenergie. Hauptsächlich wird diese Herausforderung durch übermäßige Belastung der zur Ausregelung geeigneten konventionellen thermischen Kraftwerke gelöst. Durch den regelungsbedingten ineffizienten Teillastbetrieb und wiederholte An- und Ab-fahrtvorgänge sinkt der Wirkungsgrad und ein höherer Verschleiß der ist die Folge. Aus langfristiger Sicht wird jedoch

Bewertung ein-
fügen.

Der Satz ist
scheiße, um-
schreiben.

Das am Ende zur Ende schreiben

eine Investition in Lastmanagement und in Energiespeicher unentbehrlich sein.
Brückentechnologien nicht sicher. Politischer Druck wächst.

Inhaltsangabe

| | | |
|-----|---|---|
| 2.1 | Erneuerbare Energien im Energieversorgungsnetz | 2 |
| 2.2 | Kälteerzeugung im Supermarkt: Potentiale und Besonderheiten | 3 |
| 2.3 | Objektorientierte Programmierung mit MATLAB® | 4 |
| 2.4 | Einführung in die Kältetechnik | 6 |

2.1 Erneuerbare Energien im Energieversorgungsnetz

Die erneuerbaren Energien übernehmen in Deutschland mit jedem Jahr einen größeren Anteil an der Elektrizitätsversorgung. Mehr als die Hälfte der erneuerbaren Energie wird durch Wind und Photovoltaik erzeugt. Diese Energiequellen sind jedoch im hohen Grad witterungsabhängig und können stark fluktuieren. Dadurch kann die Menge der zur Verfügung stehenden Energie von der Nachfrage zeitlich enorm abweichen. Der sichere Betrieb des Stromnetzes bei einer konstanten Frequenz von 50 Hz kann dadurch gefährdet werden. Zusätzliche Bereitstellung an Regelleistung wird dadurch notwendig. Außerdem erfolgt die Erstellung der Fahrpläne für den Einsatz der konventionellen Kraftwerke zwangsweise auf den fehlerbehafteten Vortagsprognosen über die Menge an Leistung aus erneuerbaren Energien. Die Güte der Prognose bestimmt direkt den Regel- und den Reservebedarf.

wo ist die Quelle?

Wetter- und Windprognose

Eine exakte Prognose über die Leistung aus Wind- und Photovoltaikanlagen setzt eine nach Möglichkeit getreu Wettervorhersage voraus. In der Literatur wird eine Genauigkeit für Day-Ahead Windleistungsprognose von 7% [17] und für Leistungsprognose aus Photovoltaikanlagen im Bereich von $3.5\% - 4.4\%$ [8] angegeben. Im Verlaufe des Tages können zeitweise deutlich höhere Abweichungen auftreten. Durch den gedrosselten Betrieb der Anlagen ist eine Möglichkeit zur Kompensation der Prognosefehler gegeben. Der gedrosselte Anteil kann dadurch jedoch nicht genutzt werden. Die Bereitstellung der Regelleistung durch konventionelle oder andere Kraftwerke erscheint hier als sinnvoll. Eine bessere Lösung sind Energiespeicher,

die bei Überangebot Energie speichern und bei einem Defizit Energie an das Netz freigeben.

Eine weitere Möglichkeit zur Reduzierung des obengenannten Effektes ist eine variable Gestaltung der Netzlast. Kann sich ein Verbraucher nicht zu seinem Nachteil dem Angebot der Energieerzeugung folgen, so würden die Regelungsverluste und der Aufwand verringert und damit Kosten gespart werden. Kann ein Verbraucher seine Hauptlast zeitlich verschieben, besteht das Potential diese Fähigkeit zur Regelung einzusetzen.

Es ist vorstellbar, dass Anlagen zur Kälteerzeugung auf Grund der thermischen Trägheit zu günstigen Zeiten die Temperatur zusätzlich senken können und zu ungünstigen Zeiten die Kühltätigkeit auf das Minimum herunterfahren. Hiermit ist also das Interesse begründet, das Lastverlagerungspotential in der Kälteerzeugung zu untersuchen.

2.2 Kälteerzeugung im Supermarkt: Potentiale und Besonderheiten

Im Mittel entfallen 60 % [4,7] des Stromverbrauchs in einem Supermarkt auf Kühlen und Tiefkühlen von Produkten. Der durchschnittliche Verbrauch in einem Supermarkt liegt zwischen 300 *MWh* und 500 *MWh* im Jahr [7]. In den kommenden Jahren schätzt man den Wachstum des gesamten Kältebedarfs für Deutschland mit 100 % [13]. Der Anteil am Bedarf an elektrischen Energie vom Gesamtverbrauch eines Industrielandes für Kälteerzeugung in den Supermärkten wird in der Literatur für Australien mit 1 % [9] und in Schweden mit 2 % [2,4] Prozent angegeben. In Deutschland liegt der Verbrauch mit 6294 *GWh* für das Jahr 1999 bei rund 1.27 % [16]. Die Größenordnung des Verbrauchs für die Produktlagerung in Supermärkten und die Möglichkeit diesen zeitlich zu verschieben, macht die Supermärkte für die Regelung besonders interessant.

Eine Überleitung finden

Beschränkungen und Randbedingungen

Die Mindesthaltbarkeit für bestimmten Produkte kann nur durch Lagerung dieser Produkte in für sie festgelegtem Kühltemperaturbereichen garantiert werden. Dieser Temperaturbereich ist je nach Bedarf und Anwendung in Normalkühlung (NK) über 0 °C und in Tiefkühlung (TK) unter 0 °C unterteilt. Auf der nationalen und auch auf internationalen Ebene existieren Auflagen, die die maximale Temperaturen bei der Lagerung von Nahrungsmitteln vorschreiben. Die fundamentale Verordnung ist die EG 853/2004. Aus der geht hervor, dass normalgekühlte Produkte im Temperaturbereich 0 bis +8 °C je nach Nahrungsmittel und tiefgekühlte Produkte mindestens bei −18 °C gekühlt werden müssen.

Die Kälteerzeugung im Supermarkt kann räumlich generell in zwei Bereiche unterteilt werden, dem Verkaufsbereich und dem Warenlager. Außerdem kommen Kühleinheiten zum Einsatz, die einer Verbundkälteanlage angehören oder stecker-

fertig zur Verfügung stehen. Überwiegend werden im Verkaufsbereich Kühlregale, Kühltruhen und Kühlschränke im Tiefkühlbereich und im Niederkühlbereich ebenso wie Kühltheken im Niederkühlbereich installiert. Zur Ausstattung zählen Geräte in offener sowie durch Türen verschließbarer Ausführung. Großenteils werden, die die offenen Modelle außerhalb der Öffnungszeiten durch Decken und Rollos zwecks Energieeinsparung verschlossen.

2.3 Objektorientierte Programmierung mit MATLAB®

Seit Ende des letzten Jahrhunderts herrscht in der Fachliteratur für Informatik die Meinung, dass der Einsatz von objektorientierten Techniken Programme hervorbringt, die im Vergleich *einfacher erweiterbar*, *besser testbar* und *besser wartbar* sind. Dabei wird ein Verfahren angewendet, nachdem große Systeme in kleinere Teile des Ganzen zerlegt werden. Programme lassen sich dadurch im Allgemeinen mit weniger Aufwand und kleineren Fehlerwahrscheinlichkeit programmieren. Inspiriert durch die Vorgänge aus der realen Welt, werden die Abläufe durch operierende Objekte dargestellt, die Aufträge erledigen und vergeben können. Die wesentlichen Eigenschaften der objektorientierten Programmierung, kurz OOP, sind die Datenkapselung, die Polymorphie und die Vererbung.¹

Allgemeine Erläuterungen

Klasse Eine Klasse ist ein Instrument der Programmierung zur Erfassung von charakteristischen Eigenschaften zusammenhängender Objekte. Die Definition der Strukturen der Objekte erfolgt durch Klassen.

Objekt Ein Objekt ist ein konkretes Exemplar einer Klasse.

Eigenschaften Eigenschaften sind Variablen, die für jedes Objekt existieren.

Methoden Methoden sind objektlokale Funktionen.

Datenkapselung Man spricht von Kapselung, wenn Objekte den Zugriff auf ihre Daten kontrollieren.

Polymorphie Können unterschiedliche Objekte auf eine gleiche Nachricht unterschiedlich reagieren, spricht man von Polymorphie.

Vererbung Die Vererbung ermöglicht durch Veränderung der bestehenden Klassen neue Klassen zu erstellen. Die grundlegenden Programmteile der bestehenden Klasse werden zwangsläufig übernommen.

¹ Ausführliche Informationen dazu findet man z.B. in [3], [11], [12] oder [18].

Erläuterungen zur OOP-Syntax²

classdef Neue Klassen werden mit der Anweisung `classdef` eingeleitet. Der Name der Klasse wird unmittelbar nach der Anweisung eingetragen. Der Klassen-Block endet mit der Anweisung `end`.

properties Die Anweisung `properties` beginnt den Eigenschaften-Block. Abgeschlossen wird der Block mit `end`. Innerhalb einer Klasse können mehrere Blöcke existieren. Möchte man das Verhalten eines Eigenschaften-Blocks zusätzlich verändern, zum Beispiel neu Zugriffsarten oder Zugriffsrechte vergeben, ist der Einsatz von Attributen zweckmäßig.

```
properties (attribut1, attribut2, etc.)  
end
```

methods Die Anweisung `methods` beginnt den Methoden-Block und mit `end` wird er geschlossen. Analog zu dem Eigenschaften-Block können in einer Klasse mehrere Methoden-Blöcke existieren. Möchte man das Verhalten eines Methoden-Blocks zusätzlich verändern, ist der Einsatz von Attributen zweckmäßig.

```
methods (attribut1, attribut2, etc.)  
end
```

Klassen und Objekte mit MATLAB®

Klassen werden in MATLAB® mit Hilfe einer Datei mit der Endung `.m` definiert, die den selben Namen wie die Klasse hat. Im Quelltextbeispiel MATLAB®-Code 2.1 wird eine Beispielklasse `class_name` mit einem Eigenschaften-Block und einem Methoden-Block definiert. Die Datei muss demnach `class_name.m` heißen. In die erste Zeile der Datei kommt die Anweisung `classdef`, die die Klasse einleitet. Werden der Klasse zusätzliche Verhaltensmuster zugewiesen, folgt die Setzung der Attribute in Klammern. Anschließend kommt der Name der Klasse. Wird die neue Klasse eine durch Vererbung abgeleitete einer anderen Klasse, so folgt das Klein-erzeichen und der Name der Superklasse. In den anschließenden Zeilen werden der Eigenschaften-Block und der Methoden-Block definiert.

Objekte einer Klasse werden durch eine Konstruktor-Methode erzeugt. Diese Funktion wird im ersten Methoden-Block an erster Stelle definiert. Sie hat den selben Namen wie die Klasse. Über die Konstruktor-Funktion können bereits bei der Erzeugung der Objekte Werte an ausgewählte Attribute übergeben werden. Erfolgt der Aufruf einer Konstruktor-Funktion einer Klasse fehlerfrei, so ist das Ergebnis

² Es werden nur Schlüsselwörter vorgestellt, die essentiell sind. Eine tiefgreifende Darstellung würde den Rahmen einer Studienarbeit bei Weitem übersteigen. Ausführliche Informationen dazu findet man z.B. in [15] oder <http://www.mathworks.com/help/>.

MATLAB®-Code 2.1: Beispiel Klassendefinition

```

classdef (Attributes) class_name < super_class % class definition
    properties (Attributes) % first property block
        PropertyName = [];
    end % end of properties block
    % additionally here can be another methods block with
    % specifying by another Attributes
    methods (Attributes)
        function obj = class_name(obj,a) % consturctor
            obj.PropertyName = a;
        end
        function [X] = second_function(obj)
            X = obj.PropertyName + 1;
        end
    end
    % additionally here can be another methods block with
    % specifying by another Attributes
end % end of classdef block

```

des Aufrufes ein Objekt dieser Klasse. Der Aufruf eines Konstruktors unterscheidet sich nicht von dem Aufruf gewöhnlicher MATLAB®-Funktionen. Im folgenden Quelltextbeispiel wird im Command-Window durch den Aufruf der Konstruktor-Funktion ein Objekt **object** der im MATLAB®-Code 2.1 vorgestellten Klasse erzeugt.

```
>> object = class_name(input_data);
```

Methoden, die applikationsspezifische Operationen mit einem Datensatz durchführen sollen, werden nach der Konstruktor-Funktion definiert. Die meisten dieser Funktionen nutzen das Objekt als Eingabeargument, zum Beispiel: `second_function(obj)`. Der Zugriff auf die Variablen der Objekteigenschaften erfolgt durch Referenzierung auf das Objekt. Im Quelltextbeispiel MATLAB®-Code 2.1 wird das wie folgt vorgestellt: `obj.PropertyName`. Wenn ein Zugriff auf Objektfunktionen erlaubt ist, kann darauf wie folgt zugegriffen werden:

```
>> function_return = object.second_function;
```

Die Referenz auf den Namen des Objektes muss stets erfolgen.

2.4 Einführung in die Kältetechnik

An dieser Stelle wird ein knapper Überblick über die mathematischen und physikalischen Zusammenhänge gegeben, die bei der Entwicklung eines Modells einer Kälteanlage in einem Modellsupermarkt zwingend beachtet werden müssen.

In den Supermärkten werden Kühlgeräte zur Lagerung der Ware bis zum Verkauf an den Endkunden eingesetzt. Um die Haltbarkeit dieser Ware für den Mindestzeitraum zu gewährleisten, wird sie bei niedrigen Temperaturen gehalten. Körper mit unterschiedlicher Temperatur sind bestrebt, wenn sie thermisch von einander nicht vollkommen isoliert sind, durch gegenseitige Wechselwirkung ihre Temperaturen anzugleichen. Infolge dessen wird ein Wärmegleichgewicht erreicht. Der natürliche Wärmefluss findet selbstständig immer vom Körper mit höheren Temperatur in die Richtung des Körpers mit der kleineren Temperatur statt. Um eine negative Temperaturänderung herzustellen und diese auch zu halten, muss die eindringende Wärmeenergie ständig in der selben Höhe abgeführt werden, damit die Temperatur konstant bleibt. Diese Energiemenge pro Zeiteinheit wird als Kälteleistung bezeichnet. Eine Abweichung von dieser Menge führt zum Steigen der Temperatur, wenn weniger und zum sinken der Temperatur wenn mehr abgeführt wird. Um diesen Kühlkreislauf aufrecht zu erhalten, muss Leistung aufgewendet werden³.

In der Abbildung 2.1 auf der 8 ist die Prinzipdarstellung der weitverbreiteten Kompressionskältemaschine zu sehen. Die Ware, die symbolisch durch einen Kohlkopf dargestellt ist, wird mit einer Temperatur zwischen ± 0 und $+8^\circ\text{C}$ gelagert. Der Temperaturunterschied zwischen dem Innen,- und Außenbereich folgt der Wärmeausgleich (Verlust an Kälte). Die Kälteenergie, die in der Substanzmasse (der Ware) gespeichert wird, geht dabei für den Betreiber verloren.

Die Aufnahme der Wärmeenergie und der spezifischen Wärmekapazität dieser Substanzmasse ergibt die Temperaturdifferenz Δt .

$$\Delta t = \frac{Q}{m \cdot c} \quad (2.1)$$

| | |
|------------|---|
| Δt | Temperaturdifferenz in Kelvin K |
| Q | Eindringende Wärmeenergie in kJ |
| m | Substanzmasse zur Aufnahme der Wärmeenergie in kg |
| c | Spezifische Wärmekapazität der Substanzmasse in $\frac{\text{kJ}}{\text{kg} \cdot K}$ |

Die installierte Kälteleistung multipliziert mit der täglichen Betriebszeit muss zwangsweise größer oder gleich dem stündlichen Kältebedarf multipliziert mit der Tagesstundenzahl sein.

$$\dot{Q}_0 = \frac{24 h}{\tau_B} \cdot \dot{Q}_0 \quad (2.2)$$

| | |
|-------------|---|
| \dot{Q}_0 | Installierte Kälteleistung in kW |
|-------------|---|

³ Eine detaillierte Beschreibung dieser Prozesse in einer Kompressionskälteanlage und Spezifikation ist nicht Gegenstand dieser Arbeit. Ausführliche Informationen dazu findet man z.B. in [2,10,14].

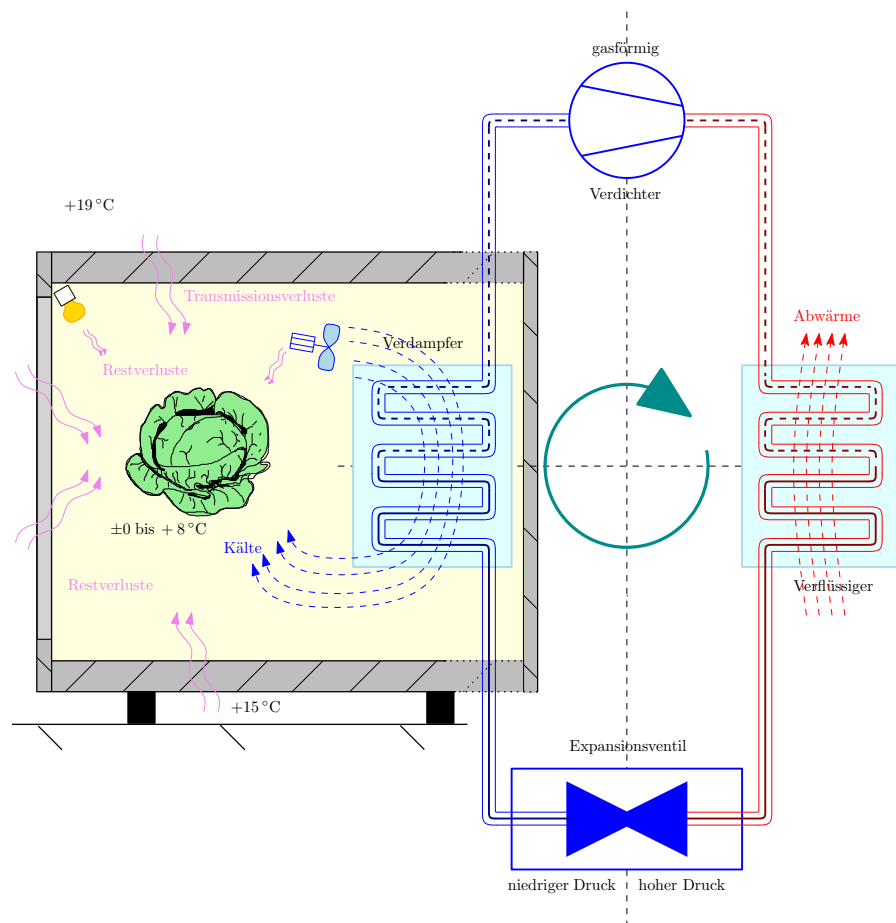


Abbildung 2.1: Prinzipdarstellung Kompressionskälteanlage

τ_B Tägliche Betriebszeit in h

\dot{Q}_0 Kältebedarf in kW

Die Transmissionswärmeleistung wird aus der Multiplikation der Fläche der wärmeübertragenden Wände mit ihrem spezifischen Wärmedurchgangskoeffizient und der Temperaturdifferenz zwischen der Kühlraumtemperatur und der Umgebungstemperatur berechnet.

$$\dot{Q}_{Tr} = A \cdot k \cdot \Delta t \quad (2.3)$$

\dot{Q}_{Tr} Transmissionswärmeleistung in kW

A Fläche in m^2

k Wärmedurchgangskoeffiziente in $\frac{W}{m^2 \cdot K}$

Δt Temperaturdifferenz in K

Der Zusammenhang zwischen der aufgewendeten elektrischen Antriebsleistung P eines Verdichters in einer Kompressionskälteanlage und der genutzten Kälteleistung \dot{Q}_0 wird durch die Kältezah ε wiedergegeben. Die Leistungszahl wird in die zweite Spalte im Array (vergl. Zeile 4 im MATLAB®-Code 4.3) eingetragen.

$$\varepsilon = \frac{\dot{Q}_0}{P} \quad (2.4)$$

| | |
|---------------|---|
| ε | Leistungszahl (einheitenlos) |
| \dot{Q}_0 | Kälteleistungsbedarf in kW |
| P | elektrische Verdichterantriebsleistung in einer Kompressionskälteanlage in kW |

Die Öffnungszeit des Supermarkts hat einen spürbaren Einfluss auf die Größe der Wärmeverluste. In der Literatur wird der Nachtverbrauch mit 10% bis 20% des Tagesverbrauchs angegeben [5]. In der Nacht fallen keine zusätzlichen Verluste zum Beispiel durch Licht, Körperwärme oder Türöffnungszeiten, sodass nur Transmissionsverluste bei der Berechnung beachtet werden.

$$\dot{Q}_{Nacht} = \dot{Q}_{Tr} \quad (2.5)$$

| | |
|-------------------|--------------------------------------|
| \dot{Q}_{Nacht} | Leistungsbedarf in der Nacht in kW |
| \dot{Q}_{Tr} | Transmissionswärmeleistung in kW |

Der Leistungsbedarf am Tag ergibt sich aus der Summe des Tagesmehrbedarfs und der Transmissionswärmeleistung. Aus Gründen der Vereinfachung wird Tagesmehrbedarf als weitgehend konstant angenommen.

$$\dot{Q}_{Tag} = \dot{Q}_{mehr} + \dot{Q}_{Tr} \quad (2.6)$$

| | |
|------------------|------------------------------------|
| \dot{Q}_{Tag} | Leistungsbedarf am Tag in kW |
| \dot{Q}_{mehr} | Mehrbedarf am Tag in kW |
| \dot{Q}_{Tr} | Transmissionswärmeleistung in kW |

Die in den Kühlbereich eindringende Verlustwärmemenge ist aufgrund der oben genannten Schlussfolgerungen je nach Tageszeit unterschiedlich.

$$Q_v = \begin{cases} 1h \cdot \dot{Q}_{Nacht}, & \text{außerhalb der Öffnungszeiten} \\ 1h \cdot \dot{Q}_{Tag}, & \text{innerhalb der Öffnungszeiten.} \end{cases} \quad (2.7)$$

| | |
|-------------------|--|
| Q_v | eindringende Verlustwärmemenge in kJ |
| \dot{Q}_{Tag} | Leistungsbedarf am Tag in kW |
| \dot{Q}_{Nacht} | Mehrbedarf am Tag in kW |
| h | Zeiteinheit für Stunden |

Die Berechnung der mittleren Transmissionswärmeleistung erfolgt durch die Multiplikation der Differenz zwischen der Umgebungstemperatur und der mittleren Kühlraumtemperatur mit dem Wärmedurchgangskoeffizienten und der Fläche der wärmeübertragenden Wände.

$$\bar{\dot{Q}}_{Tr} = A \cdot k \cdot (t_{amb} - \bar{t}_{KR}) \quad (2.8)$$

| | |
|----------------------|--|
| $\bar{\dot{Q}}_{Tr}$ | mittlere Transmissionswärmeleistung in kW |
| A | Fläche in m^2 |
| k | Wärmedurchgangskoeffiziente in $\frac{W}{m^2 \cdot K}$ |
| t_{amb} | Umgebungstemperatur in $^{\circ}C$ |
| \bar{t}_{KR} | mittlere Kühlraumtemperatur in $^{\circ}C$ |

Ist für eine Kälteanlage der Kälteleistungsbedarf bekannt, so wird der Tagesmehrbedarf an Kälteleistung ermittelt, indem vom Produkt des Kälteleistungsbedarfes für 24 h mit dem Faktor für Kältebedarfsabsenkung die mittlere Transmissionswärmeleistung abgezogen wird.

$$\dot{Q}_{mehr} = \dot{Q}_0 \cdot K - \bar{\dot{Q}}_{Tr} \quad (2.9)$$

| | |
|----------------------|---|
| \dot{Q}_{mehr} | Mehrbedarf am Tag in kW |
| \dot{Q}_0 | Kälteleistungsbedarf in kW |
| K | Faktor für Kältebedarfsabsenkung (einheitenlos) |
| $\bar{\dot{Q}}_{Tr}$ | mittlere Transmissionswärmeleistung in kW |

Ist der Kälteleistungsbedarf nicht bekannt, wie zum Beispiel bei steckerfertigen Geräten, kann der Mehrbedarf am Tag über den Wert Verdichterarbeit pro 24 Stunden W_{Verd24} für den gesamten Kälteverbraucher ermittelt werden.

Das Produkt aus dem spezifischen Energieverbrauch mit dem Faktor für Kältebedarfsabsenkung, dem Verdichteranteil ergibt die Verdichterarbeit.

$$v_{mod} = K \cdot v \quad (2.10)$$

$$W_{Verd24} = W_{spez24} \cdot v_{mod} \quad (2.11)$$

| | |
|--------------|---|
| W_{Verd24} | Verdichterarbeit pro 24 h in $\frac{kWh}{24h}$ |
| W_{spez24} | spezifischer Energieverbrauch pro 24 h in $\frac{kWh}{24h}$ |
| K | Faktor für Kältebedarfsabsenkung (einheitenlos) |
| v | Verdichteranteil (einheitenlos) |
| v_{mod} | modifizierte Verdichteranteil (einheitenlos) |

Im Folgenden muss die Verdichterarbeit zum Abführen von zusätzlich eindringenden Wärmeenergie in der Öffnungszeit berechnet werden, die einem weiteren Schritt in Wärmeleistung \dot{Q}_{mehr} umgewandelt wird.

$$W_{mehr} = W_{Verd24} - \frac{\bar{\dot{Q}}_{Tr}}{\varepsilon} \cdot 24h \quad (2.12)$$

| | |
|----------------------|--|
| W_{mehr} | Verdichterarbeit zum Abführen von eindringenden Wärmeenergie Q_{mehr} in kWh in der Öffnungszeit |
| W_{Verd24} | Verdichterarbeit pro 24 h in $\frac{kWh}{24h}$ |
| $\bar{\dot{Q}}_{Tr}$ | mittlere Transmissionswärmeleistung in kW |
| ε | Leistungszahl (einheitenlos) |

Die Umrechnung von W_{mehr} in \dot{Q}_{mehr} erfolgt mit Hilfe der Leistungszahl ε . Die Verdichterarbeit W_{mehr} wird mit ε multipliziert. Es wird angenommen, dass der Mehrbedarf in den rund zwölf Stunden der Öffnungszeit entsteht.

$$\dot{Q}_{mehr} = \frac{W_{mehr}}{12h} \cdot \varepsilon \quad (2.13)$$

| | |
|------------------|--|
| \dot{Q}_{mehr} | Mehrbedarf am Tag in kW |
| W_{mehr} | Verdichterarbeit zum Abführen von eindringenden Wärmeenergie Q_{mehr} in kWh in der Öffnungszeit |
| ε | Leistungszahl (einheitenlos) |

Dabei entspricht \dot{Q}_{Nacht} der mittleren Transmissionswärmeleistung $\bar{\dot{Q}}_{Tr}$. \dot{Q}_{Tag} ist der Kältebedarf \dot{Q}_0 , multipliziert mit dem Faktor für die Kältebedarfsabsenkung K bei Geräten, bei denen der Kältebedarf gegeben ist. Bei steckerfertigen Geräten ist \dot{Q}_{Tag} die Summe aus dem Mehrbedarf an Leistung am Tag \dot{Q}_{mehr} und der mittleren Transmissionswärmeleistung $\bar{\dot{Q}}_{Tr}$.

ab hier noch
nicht umfor-
mulierte

Mit den Wärmeverlusten, die am Tag und in der Nacht in unterschiedlicher Größe auftreten, wird für jeden Zeitschritt die Zeit bis zum kritischen Temperaturmaximum bestimmt. Diese Zeit braucht das Programm, um den Einsatz der Supermarktkälteanlagen als Speicher zu planen. Mit der Gleichung

$$\bar{\dot{Q}}_{Tr_{ln}}(i) = \frac{\dot{Q}_{Tr_{max}}(i) - \dot{Q}_{Tr}(i)}{\log\left(\frac{\dot{Q}_{Tr_{max}}(i)}{\dot{Q}_{Tr}(i)}\right)} \quad (2.14)$$

$$\tau_{krit}(i) = \frac{m \cdot c \cdot (t_{max} - t(i))}{\bar{\dot{Q}}_{Tr_{ln}}(i)} \quad (2.15)$$

| | |
|---------------------------|---|
| τ_{krit} | Zeit bis zur kritischen Temperatur in h |
| m | Substanzmasse zur Aufnahme der Wärmeenergie in kg |
| c | Spezifische Wärmekapazität der Substanzmasse in $\frac{kJ}{kg \cdot K}$ |
| t_{max} | obere Temperaturgrenze in $^{\circ}C$ |
| $t(i)$ | Temperatur zur Stunde i in $^{\circ}C$ |
| $\bar{\dot{Q}}_{Tr_{ln}}$ | logarithmierte Mittelwert der Transmissionswärmeleistung in kW |

wird die Zeit τ_{krit} für jeden Zeitpunkt i berechnet, wobei $\bar{\dot{Q}}_{Tr_{ln}}$ der logarithmische Mittelwert ist, der sich aus den Transmissionswärmeleistungen zum jeweils aktuellen Zeitpunkt i mit der Temperatur $t(i)$ und den Transmissionswärmeleistungen zum Zeitpunkt, an dem der Kühlinnenraum die maximale Temperatur t_{max} erreicht hätte, berechnet. Am Tag müssen die restlichen Verluste \dot{Q}_{mehr} zusätzlich zu den Transmissionwärmeverlusten für die Berechnung der Zeit bis zur kritischen Temperatur berücksichtigt werden, wodurch sich folgende Gleichung ergibt:

$$\tau_{krit}(i) = \frac{m \cdot c \cdot (t_{max} - t(i))}{\bar{\dot{Q}}_{Tr_{ln}}(i) + \dot{Q}_{mehr}} \quad (2.16)$$

| | |
|---------------------------|---|
| τ_{krit} | Zeit bis zur kritischen Temperatur in h |
| m | Substanzmasse zur Aufnahme der Wärmeenergie in kg |
| c | Spezifische Wärmekapazität der Substanzmasse in $\frac{kJ}{kg \cdot K}$ |
| t_{max} | obere Temperaturgrenze in $^{\circ}C$ |
| $t(i)$ | Temperatur zur Stunde i in $^{\circ}C$ |
| $\bar{\dot{Q}}_{Tr_{ln}}$ | logarithmierte Mittelwert der Transmissionswärmeleistung in kW |
| \dot{Q}_{mehr} | Mehrbedarf am Tag in kW |

Die Zeit τ_{krit} ist abhängig vom Anstieg der Temperaturen und dieser wiederum von den eindringenden Wärmelasten.

Um den Temperatúrausgleich in den Lebensmitteln im Algorithmus zu berücksichtigen, werden deshalb die zu- und abgeführten Wärmeenergiemengen bei der Berechnung der Temperatur für jeden Zeitschritt stets mit dem Faktor 0,8 multipliziert. Die Gleichung zur stündlichen Berechnung der aktuellen Temperatur ist damit:

$$t(i+1) = 0.8 \cdot \frac{Q_v - Q_{ab}}{m \cdot c} h + t(i) \quad (2.17)$$

t Temperatur in $^{\circ}C$

Q_v eindringende Verlustwärmemenge in kJ

Q_{ab} abführende Wärmemenge in kJ

wobei Q_v die aktuell eindringende Wärmeenergie und Q_{ab} die abgeführte Wärmeenergie ist.

die nachfolgende Gleichung gehört nicht dahin! oder?

$$Q_{max} = m \cdot c \cdot (t(i) - t_{min}) + Q_v \quad (2.18)$$

Q_{max} maximal abzuführende Wärmeenergiemenge in kJ

m Substanzmasse zur Aufnahme der Wärmeenergie in kg

c Spezifische Wärmekapazität der Substanzmasse in $\frac{kJ}{kg \cdot K}$

Q_v eindringende Verlustwärmemenge in kJ

t_{min} obere Temperaturgrenze in $^{\circ}C$

$t(i)$ Temperatur zur Stunde i in $^{\circ}C$

Aufgabenstellung

Die Intention der Arbeit, ist der objektorientierte Entwurf eines MATLAB® - Programms zur Simulation des variablen Lastverhaltens von Kältelasten mit Kältespeichern im Energieversorgungsnetz.

Es sollen ausschließlich Kälteanlagen modelliert werden, die in den Supermärkten zum Einsatz kommen. Das Ergebnis der Simulation soll der Verbrauch einer variabel geführten Supermarktkette auf der Basis der einzelnen in den Supermärkten eingesetzten Kälteanlagen sein.

Verbrauchswerte in einem Lastflussoptimierungsprogramm zu verwenden.

Explizite Anforderungen

Folgende explizite Anforderungen sind an das Programm gestellt:

- Die Speicherung der für das Beschreiben der Modelle erforderlichen Parameter findet in einer Konfigurationsdatei statt.
- Der elektrische Energieverbrauch wird auf Grund der Modellparameter der Kältelasten sowie des gefahrenen Einsatzes der Kältelasten berechnet.
- Die Topologie des elektrischen Netzwerkes werden berücksichtigt.
- Die Möglichkeit einer eindeutigen Zuweisung der Verbrauchswerte für folgende Verursacher wird realisiert:
 - Daten für jede Kälteanlage werden einzeln gespeichert, wenn der Verbrauch mehrerer Anlagen simuliert wird.
 - Daten für jede Supermarktkette werden einzeln gespeichert, wenn der Verbrauch mehrerer Supermarktketten simuliert wird.
 - Erfolgt die Simulation des Energieverbrauchs für ein Mehrknotennetz, so werden die Daten für jeden Knoten, einzeln gespeichert.

Modellierung

Inhaltsangabe

| | |
|---|-----------|
| 4.1 Erstellung des OOP-Modells | 15 |
| 4.2 Handhabung des Programms | 25 |

Der durchschnittliche Energieverbrauch der Kälteanlagen je Supermarktkette kann aufgrund der technischen Ausführung unterschiedlich sein. Der Energieverbrauch entsteht an definierten Punkten im Netz. An den einzelnen Knotenpunkten können mehrere Kältelasten angeschlossen sein.

4.1 Erstellung des OOP-Modells

In der Abbildung 4.1 wird ein einfaches Energieversorgungsnetz mit vier Knoten dargestellt. Am Knoten eins ist eine regenerative elektrische Energiequelle, in diesem Fall ein Windpark, angeschlossen. Weitere konventionelle elektrische Energiequellen befinden sich an den Knoten zwei und drei. Die passiven Lasten befinden sich am Knoten zwei und vier. Der Kältespeicher ist am Knoten zwei angeschlossen. Im Bild wird der Kältespeicher Supermarkt durch ein Gebäude mit Einkaufswagen dargestellt. Die Knoten sind untereinander durch Leitungen verbunden.

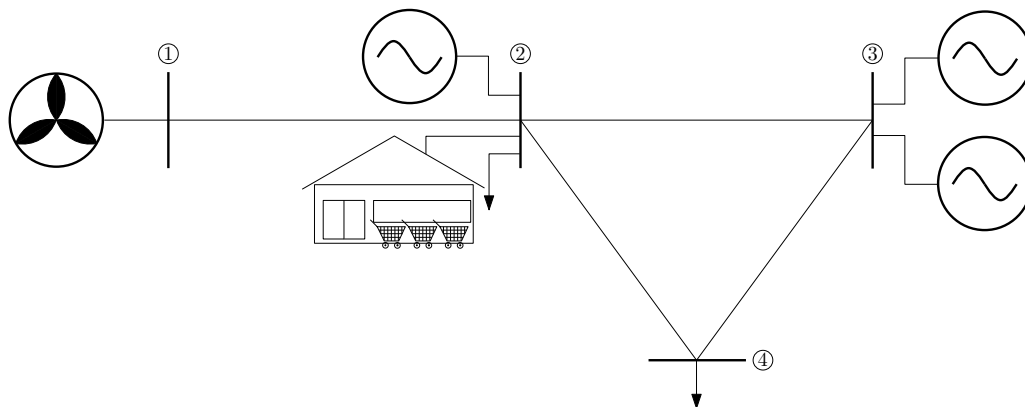


Abbildung 4.1: Vier Knoten Beispiel

In der Realität kann ein Energieversorgungsnetz durch die Variation der Knotenzahl und die Vermaschung beliebig komplizierte Form aufweisen.

In Anbetracht der im Abschnitt 2.3 vorgestellten Verfahren erscheint der Ansatz der OOP bei der Erstellung eines Programms zur Simulation einer Kälteslast mit Kältespeicher im Energieversorgungsnetz auf der Basis des vorangegangenen Beispiels (Abbildung 4.1 auf der Seite 15) als sinnvoll. In der Abbildung 4.2 auf der Seite 16 wird in Form eines Klassendiagramms das fertige Klassen-Modell dargestellt und die Abhängigkeit unter Klassen visualisiert.

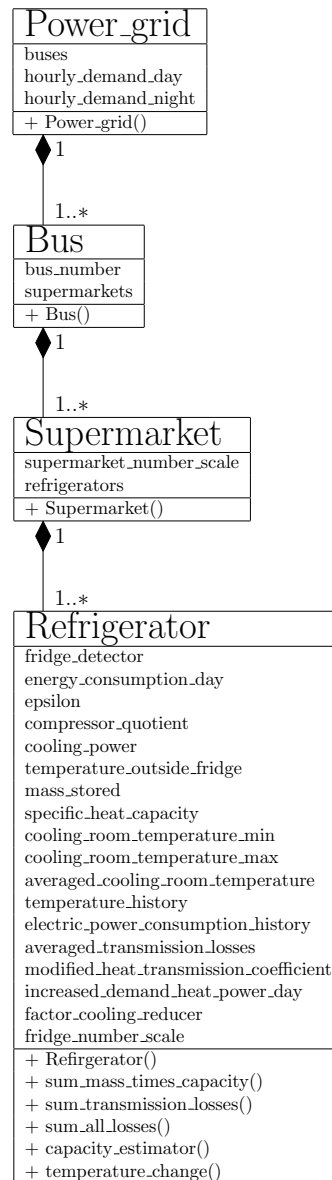


Abbildung 4.2: Klassendiagramm Modellkonstrukt

Das Ergebnis der Abstraktion sind vier Klassen. In der Klasse **Refrigerator** sind Eigenschaften und Methoden zusammengefasst, die das Modell einer Kälte- last beschreiben. Die explizite Erklärung der.

hier ein Über-
leitungssatz

Refrigerator

An dieser Stelle werden die Eigenschaften und die Methoden der Klasse **Refrigerator** explizit erklärt.

Attribute

fridge_detector Art der Anlage. Die Berechnung der Verluste der Verbundanlagen unterscheidet sich von der der steckerfertigen Geräten.

energy_consumption_day Höhe des durchschnittlichen elektrischen Energieverbrauches.

epsilon Leistungszahl.

compressor_quotient Anteil des Verdichters am Gesamtverbrauch der Kälteanlage.

cooling_power Kälteleistungsbedarf (Herstellerangaben).

temperatur_outside_fridge $1 \times M$ -Array mit Außentemperaturwertet für die Umgebung der Kälteanlage. M stimmt mit der Anzahl der Wände überein, für die ihre Fläche bekannt ist.

mass_stored $1 \times N$ -Array mit den Massewerten, der zur Kühlung vorhandenen Substanzmasse. N stimmt mit der Anzahl der Massen überein, für die die spezifische Wärmekapazität unterschiedlich ist.

specific_heat_capacity $1 \times N$ -Array mit den Werten für spezifische Wärmekapazitäten der Substanzmassen.

cooling_room_temperature_min Niedrigste Temperatur, die durch Kühlung erreicht werden darf.

cooling_room_temperature_max Höchste Temperatur, die durch Kühlung erreicht werden darf.

averaged_cooling_room_temperature Durchschnittliche Temperatur im Normalbetrieb.

temperature_history $n \times m$ -Array mit $(n, m) \in \mathbb{Z}_0^+$, wobei n für die Anzahl der Werten des berechneten Temperaturverlaufes und m für die Anzahl der simulierten Tage stehen. Die Werte sind das Ergebnis der Berechnung der Funktion **main_equation**. Eine Spalte des Arrays stellt den Temperaturverlauf an genau einem Tag dar.

stehen oder steht?

electric_power_consumption_history $n \times m$ -Array mit $(n, m) \in \mathbb{Z}_0^+$, wobei n für die Anzahl der Werten des berechneten Verbrauchs und m für die Anzahl der simulierten Tage stehen. Die Werte sind das Ergebnis der Berechnung

stehen oder steht?

der Funktion **main_equation**. Eine Spalte des Arrays stellt den Verlauf des berechneten Verbrauches an genau einem Tag dar.

averaged_transmission_losses Summe aller durchschnittlichen Transmissionsverluste im Normalbetrieb (vgl. Gleichung (2.8) auf der Seite 10).

modified_heat_transmission_coefficient $1 \times M$ -Array mit Werten aus der elementenweise Multiplikation des Vektors mit Werten der spezifischen Wärmedurchgangskoeffizienten mit dem Vektor mit Werten für die Durchgangsflächen (vgl. Gleichung (2.10) auf der Seite 10).

increased_demand_heat_power_day Mehrbedarf am Tag, aufgrund der zusätzlichen Verluste durch Licht, Menschenkörperwärme, Türöffnungszeiten und weiteren Einflüsse. Die Berechnung erfolgt nach einer Auswahl zwischen der Gleichung (2.9) und der Gleichung (2.11) bis Gleichung (2.13) wie auf der Seiten 10 und 11 beschrieben.

factor_cooling_reducer Faktor für Kältebedarfsabsenkung infolge der Luftfeuchtigkeit und der Umgebungstemperatur.

fridge_number_scale Anzahl der Anlagen mit der identischen technischen Ausführung, der Beladung und dem Fahrplan.

Methoden

Refrigerator(fridge_config,number_steps,days) Konstruktor der Klasse. Bei dem Aufruf dieser Funktion werden folgende Argumente übergeben:

fridge_config 1×2 -Cell Array mit Modellparametern¹.

number_steps Anzahl der Zeitschritte die einen Tag beschreiben.

days Anzahl der zu simulierenden Tage.

Bei der Initialisierung eines Objektes werden einem Teil der oben vorgestellten Attribute der Klasse **Refrigerator** Werte aus dem **fridge_config**-Array direkt übergeben. Andere Werte werden einmalig von der Konstruktor-Methode berechnet oder im Laufe der Simulation verändert.

Folgende Attribute werden einmalig bei der Initialisierung berechnet:

averaged_transmission_losses Summe aller durchschnittlichen Transmissionsverluste im Normalbetrieb (vgl. Gleichung (2.8) auf der Seite 10).

¹ Die Form des Arrays ist äquivalent zu genau einer Zeile der in MATLAB®-Code 4.2 auf der Seite 29 vorgestellten Arrays, die die Modelle der Supermärkte abbilden.

modified_heat_transmission_coefficient $1 \times M$ -Array mit Werten aus der elementenweise Multiplikation des Vektors mit Werten der spezifischen Wärmedurchgangskoeffizienten mit dem Vektor mit Werten für die Durchgangsflächen (vgl. Gleichung (2.10) auf der Seite 10).

increased_demand_heat_power_day Mehrbedarf am Tag, aufgrund der zusätzlichen Verluste durch Licht, Menschenkörperwärme, Türöffnungszeiten und weiteren Einflüsse. Die Berechnung erfolgt nach einer Auswahl zwischen der Gleichung (2.9) und der Gleichung (2.11) bis Gleichung (2.13) wie auf der Seiten 10 und 11 beschrieben.

Folgende Attribute werden im Laufe der Simulation verändert:

temperature_history $n \times m$ -Array mit $(n, m) \in \mathbb{Z}_0^+$, wobei n für die Anzahl der Werten des berechneten Temperaturverlaufes und m für die Anzahl der simulierten Tage stehen. Die Werte sind das Ergebnis der Berechnung der Funktion **main_equation**. Eine Spalte des Arrays stellt den Temperaturverlauf an genau einem Tag dar.

stehen oder steht?

electric_power_consumption_history $n \times m$ -Array mit $(n, m) \in \mathbb{Z}_0^+$, wobei n für die Anzahl der Werten des berechneten Verbrauchs und m für die Anzahl der simulierten Tage stehen. Die Werte sind das Ergebnis der Berechnung der Funktion **main_equation**. Die Werte sind das Ergebnis der Berechnung der Funktion **main_equation**. Eine Spalte des Arrays stellt den Verlauf des berechneten Verbrauches an genau einem Tag dar.

stehen oder steht?

Folgende Attribute sind auf Grund der Annahmen dieser Studie konstant, können aber bei der bei einer Modifizierung der Annahmen im Laufe der Simulation verändert werden:

temperature_outside_fridge Befindet sich der Kältespeicher außerhalb geschlossener Räume und im direkten Kontakt mit der Außentemperatur, muss die Temperaturänderung die im Tagesverlauf in Abhängigkeit von der Jahreszeit entsteht, beachtet werden.

mass_stored Wenn nicht mehr Angenommen wird, dass die Substanzmasse im Supermarkt im Laufe des Tages konstant ist, sondern mit dem Konsumverhalten der Bevölkerung in Abhängigkeit steht, muss eine Anpassung erfolgen.

sum_mass_times_capacity() Der Rückgabewert dieser Funktion ist die berechnete Summe aus elementenweise Multiplikation aller Substanzmassen mit der jeweiligen spezifischen Wärmekapazitäten. Der Funktion werden keine explizite Argumente übergeben. Die Funktion greift direkt auf die Attribute **specific_heat_capacity** und **mass_stored** des Objektes zu.

Warum ist es nötig, die Funktion explizit zu schreiben?

sum_transmission_losses() Es gibt zwei Rückgabewerte dieser Funktion. Der erste Wert ist die Summe der Transmissionsverluste (vgl. Gleichung (2.8) auf

der Seite 8) durch alle Durchgangsflächen für den aktuellen Zeitpunkt und die aktuelle Kühlraumtemperatur. Desweiteren wird die Summe der Transmissionsverluste berechnet, die bei der oberen Grenze des zugelassenen Temperaturbereiches entstehen. Die Kenntnis dieser Verluste ist wichtig um die Zeitspanne abschätzen zu können, bis obere Temperaturgrenze erreicht wird (vgl. Gleichung (2.15) auf der Seite 12).

Die Übergabe folgender Argumente an die Funktion ist erforderlich:

number_steps Laufvariable. Gesamtzahl der zu simulierenden Zeitschritte für ein Tag.

count_number_steps Der aktuelle Wert von **number_steps**.

count_number_day Der aktuelle Wert von der Laufvariable **number_days**.

sum_all_losses() Es gibt zwei Rückgabewerte dieser Funktion. Der erste Rückgabewert beinhaltet die Summe aller Verluste an Kälte (vgl. Gleichung (2.7) auf der Seite 9). Außerhalb der Öffnungszeiten entfällt der Anteil an zusätzlichen Verlusten, die durch Licht, offene Türen und etc. anfallen. Durch eine alternative Verzweigung, deren Bedingung die Zugehörigkeit des aktuellen Wertes der Laufvariable **count_number_steps** zu den Öffnungszeiten ist, wird die Berücksichtigung der zusätzlichen Verluste sichergestellt. Der zweite Rückgabewert ist die Zeitspanne bis zum Erreichen der kritischen maximalen Temperatur (vgl. Gleichung (2.15) und Gleichung (2.16) auf den Seiten 12 und 12).

Die Übergabe folgender Argumente an die Funktion ist erforderlich:

number_steps Laufvariable. Gesamtzahl der zu simulierenden Zeitschritte für ein Tag.

count_number_steps Der aktuelle Wert von **number_steps**.

count_number_day Der aktuelle Wert von der Laufvariable **number_days**.

number_days Laufvariable. Gesamtzahl der zu simulierenden Tage.

capacity_estimator() Die Funktion hat einen Rückgabewert. Berechnet wird die maximal abzuführende Wärmeenergiemenge Q_{max} (vgl. die Gleichung (2.18) auf der Seite 13).

Die Übergabe folgender Argumente an die Funktion ist erforderlich:

count_number_steps Der aktuelle Wert von **number_steps**.

count_number_day Der aktuelle Wert von der Laufvariable **number_days**.

Q_losses Wert der aktuellen eindringenden Wärmeenergie als Ergebnis der Funktion **sum_all_losses()**.

temperature_change() Die Funktion hat einen Rückgabewert. Berechnet wird die Temperatur der Substanzmasse, die durch Kühlung am Ende eines Zeitschrittes erreicht wird (vgl. die Gleichung (2.17) auf der Seite 13).

Die Übergabe folgender Argumente an die Funktion ist erforderlich:

number_steps Laufvariable. Gesamtzahl der zu simulierenden Zeitschritte für ein Tag.

count_number_steps Der aktuelle Wert von **number_steps**.

count_number_day Der aktuelle Wert von der Laufvariable **number_days**.

Q_losses Wert der aktuellen eindringenden Wärmeenergie als Ergebnis der Funktion **sum_all_losses()**.

Alle Klassen haben einen unterschiedlichen Status. Die Klasse **Power_grid** vertritt die Rolle des ganzen Systems, also des Energieversorgungsnetzes, welches aus charakteristischen Teilen, den Knoten (**Bus**) besteht. Die Existenz eines Knotens außerhalb eines Netzes ergibt für die Berechnung keinen Sinn. Dieser Zusammenhang zwischen Klassen wird im Klassendiagramm Abbildung 4.2 durch eine Linienverbindung verdeutlicht. Auf der Seite des Ganzen endet die Linie mit einem ausgefüllten Rhombus. Das komplette Modell ist nach diesem Prinzip aufgestellt. Die kleinste Einheit des Ganzen stellt die Kälteanlage (**Refrigerator**) dar.

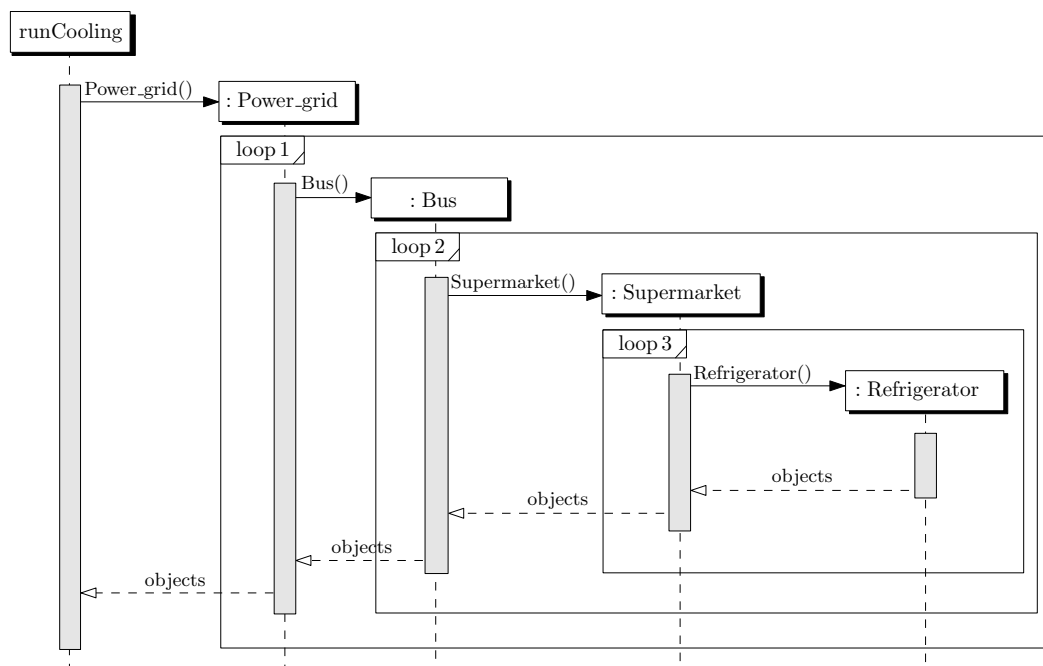


Abbildung 4.3: Sequenzdiagramm Modellkonstrukt

Auf Grund des Modells wird das elektrische Verhalten einer Supermarket-Kältelast im Energieversorgungsnetz durch miteinander kooperierende Objekte

dargestellt. Wird eine Funktion aufgerufen, die den Start der Simulation veranlasst, müssen, bevor auf die Methoden der einzelnen Objekte zugegriffen werden kann, die Objekte erzeugt werden. In der Abbildung 4.3 wird mit Hilfe eines Sequenzdiagrammes der Nachrichtenaustausch hervorgehoben, der bei der Erstellung der Objekte durchlaufen wird. In der Funktion **runCooling**, die den Start der Simulation veranlasst und steuert, wird der Konstruktor der Klasse **Power_grid** aufgerufen. Als Ergebnis des Aufrufes bekommt die Funktion **runCooling** ein Objekt der Klasse **Power_grid** mit allen Unterobjekten, die den zu simulierenden Fall beschreiben. Die Konstruktor-Funktionen der Klassen, die einen untergeordneten Status besitzen, werden bei der Erstellung der übergeordneten Objekte aufgerufen. Die Anzahl der erzeugten Objekte hängt davon ab, wie oft die Konstruktor-Funktion der jeweiligen Klasse aufgerufen wird. Diese Zahlen werden in Konfigurationsdateien abhängig von dem untersuchten Fall festgelegt. Die ausführliche Erklärung der Konfigurationsdateien wird in kommenden Abschnitten vorgenommen.

Bereitstellung der Leistung zur Kälteerzeugung

Der augenblickliche Gleichgewicht zwischen dem Kältebedarf und der zur Verfügung stehenden Kälteleistung sichert die Einhaltung der Soll-Temperatur der Kühlgüter. Werden die Kälteanlagen als Energiespeicher zur Regelung eingesetzt, so führt das auf Grund der fluktuierenden zur Verfügung stehenden Kälteleistung zu Temperaturschwankungen. Die engen Grenzen des zulässigen Temperaturbandes dürfen nicht überschritten werden.

Verschiedene Konzepte für den Einsatz der Kälteanlagen als Energiespeicher im Energieversorgungsnetz sind denkbar. Im folgenden Abschnitt wird die Implementierung eines möglicher Kooperationskonzept zwischen Supermarkt und Windpark vorgestellt.

Die prognostizierten elektrischen Windleistungsdaten und die tatsächlich vom Windpark produzierten elektrischen Windleistungsdaten der Vattenfall-Regelzone aus dem Jahr 2005 werden als Grundlage-Datensatz zur Implementierung des Kooperationskonzeptets verwendet. Im Datensatz sind Stundenwerte von Windleistung gespeichert weshalb in diesem Fall auch von Energie gesprochen werden kann.

was ist das heute?

Der Windpark ist bestrebt die Verpflichtung über die Bereitstellung von Leistung beim Netzbetreiber zu erfüllen. Abweichungen, die aufgrund der fehlerhaften Windprognosen entstehen, sollen durch die Kooperation mit Supermarktketten ausgeglichen werden.

Es wird angenommen, dass der Prognosefehler von der Windleistungsvorhersage für 24 h den 15% entspricht. Der Windparkbetreiber meldet beim Netzbetreiber die untere Grenze, also 85% von der Windleistungsvorhersage, an. Der Supermarkt besitzt einen stündlichen Bedarf, der unbedingt gedeckt werden muss. Sind die 15% von der Prognose kleiner als der stündliche Bedarf, wird die Differenz aus dem Netz gezogen. Diese Abweichung wird beim Netzbetreiber zur Erstellung des Lastfahrplanes mitgeteilt. Übersteigt der Prognosefehler die 15%, versucht der Supermarkt diesen Überschuss zu neutralisieren.

Die vom Windparkbetreiber beim Netzbetreiber anzumeldende Leistung wird berechnet, indem von der Windleistungsvorhersage die 85% bestimmt werden oder der stündliche Bedarf des Supermarktes abgezogen wird, wenn die 15% kleiner als der stündliche Bedarf sind.

$$b = 0.85 \cdot P_{DA}$$

oder

$$b = P_{DA} - P_{dem}$$

b Leistung, die vom Windparkbetreiber beim Netzbetreiber angemeldet wird in MW

P_{DA} Vorhersage der Windleistung (day ahead) in MW

P_{dem} stündlicher Leistungsbedarf des Supermarkts (demand) MW

Die vom Windparkbetreiber dem Supermarkt versprochene Leistung wird berechnet, indem von der Windleistungsvorhersage die 15% bestimmt werden oder nur der stündliche Bedarf, wenn die 15% kleiner als der stündliche Bedarf sind.

$$a = 0.15 \cdot P_{DA}$$

oder

$$a = P_{dem}$$

a Leistung, die vom Windparkbetreiber dem Supermarkt versprochen wird in MW

P_{DA} Vorhersage der Windleistung (day ahead) in MW

P_{dem} stündlicher Leistungsbedarf des Supermarkts (demand) MW

Die Leistung, die insgesamt den Kälteanlagen zur Verfügung steht, berechnet sich nach der Gleichung (4.1). Der Ausdruck $P_{RT} - b$ bedeutet die überschüssige Windleistung. Der Ausdruck $P_{dem} - a$ bedeutet die vom Supermarkts beim Netzbetreiber nicht abgemeldete Leistung.

$$P_{load} = P_{RT} - b + P_{dem} - a \quad (4.1)$$

P_{load} vorhandene Leistung zum Laden in MW

a Leistung, die vom Windparkbetreiber dem Supermarkt versprochen wird in MW

P_{dem} stündlicher Leistungsbedarf des Supermarkts (demand) MW

Implementierung

Die Implementierung des Kooperationskonzeptes erfolgt in einer neuen Klasse **Cooling_strategy**. Diese Klasse hat die Aufgabe jeder Kühlstelle in jedem Supermarkt im Netz die aufgrund des Kooperationskonzeptes und der Temperaturbeschränkungen optimale Menge an Kälteleistung zuzuweisen. Die zur Verfügung vorhandene Leistung wird auf alle Kühlstellen gleichmäßig aufgeteilt bis keine Leistung mehr zur Verfügung steht oder die Minimaltemperatur erreicht wird. Alle Kühlstellen sollen zu jeder Stunde mindestens soviel Leistung zugewiesen bekommen, dass genau diese eine Stunde überbrückt werden kann, ohne dass die Höchsttemperatur überschritten wird.

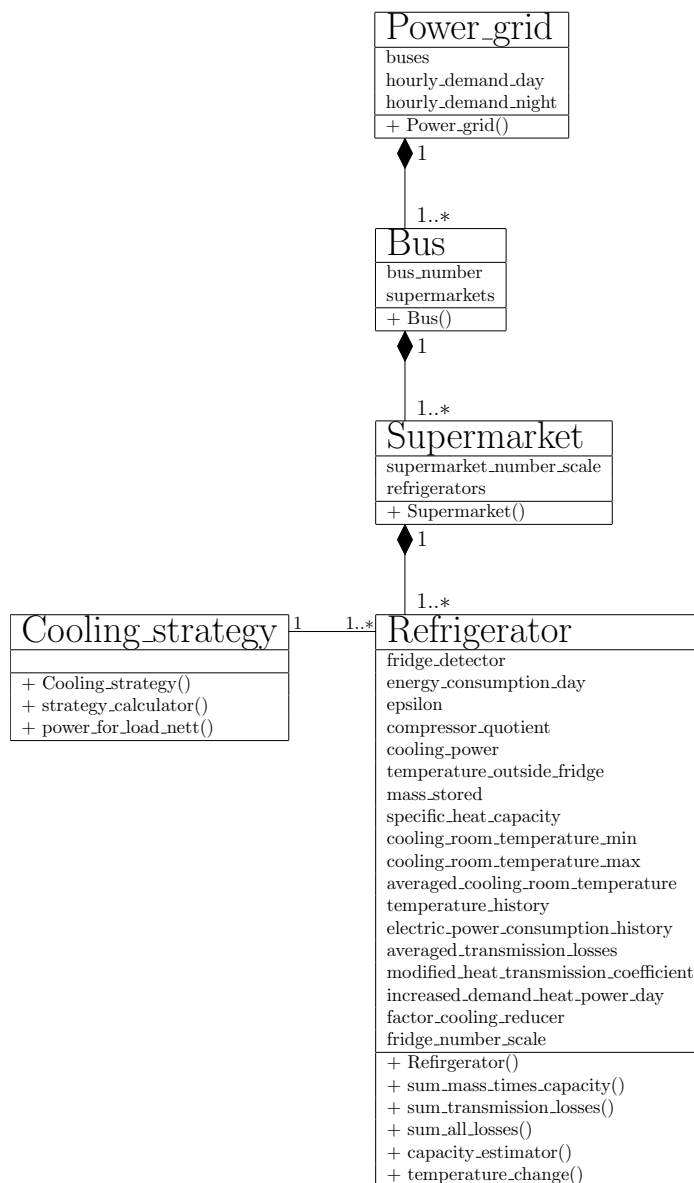


Abbildung 4.4: Klassendiagramm Kooperationskonzept

In der Abbildung 4.4 auf der Seite 24 wird in einem Klassendiagramm die Beziehung zwischen dem bestehenden Modell und der Klasse **Cooling_strategy** dargestellt. Eine Klasse **Cooling_strategy** steht in der direkten Verbindung mit der Klasse **Refrigerator**, für die sie den zur Kühlung erforderliche Leistung berechnet. Ein Objekt der Klasse **Cooling_strategy** kann mit einem bis mehreren Objekten der Klasse **Refrigerator** in Verbindung stehen.

Aus dem Diagramm wird ersichtlich, dass die Klasse **Cooling_strategy** über keine Attribute verfügt. Die Klasse besitzt drei Methoden. Das Ergebnis des Aufrufes der Konstruktor-Funktion ist ein konkretes Objekt der Klasse.

In der Abbildung 4.5 auf der Seite 26 ist der Programmstrukturplan der Kooperationskonzeptes dargestellt. Der Algorithmus zeigt den Ablauf beim “Auffühlen” der Kühlstellen für genau einen Zeitschritt der Simulation. Die Umsetzung erfolgt in der Funktion **strategy_calculator()**. Die Leistung, die zum Kühlen der Kälteanlagen zur Verfügung steht (vgl. Gleichung (4.1) auf der Seite 23), wird mit Hilfe der Funktion **power_for_load_nett()** berechnet.

Der genaue Ablauf des Informationsflusses, der zwischen den Objekten im Verlauf der Simulation stattfindet, wird im Sequenzdiagramm in der Abbildung 4.6 auf der Seite 27 grafisch verdeutlicht. Auf der Zeitachse, die senkrecht von oben nach unten verläuft, wird mit Hilfe der Pfeile der Zugriff auf die Methoden der Objekte dargestellt. Ist eine Methode des Objektes für einen Zeitraum aktiv, wird dieser Zustand durch einen grauen Balken auf dem Zeitstrahl des zugehörigen Objektes visualisiert.

4.2 Handhabung des Programms

Das Programm ist ein Packet aus MATLAB[®] M-files zur Simulation von Kälteanlagen mit Kältespeichern in einem Energieversorgungsnetz. Das Simulationsprogramm ist in der Lage das Verhalten einer bzw. mehrerer Supermarktkälteanlagen als flexibler Verbraucher in einem Energieversorgungsnetz zu simulieren. Das Programm kann über eine vordefinierte Schnittstelle in ein Leistungsflussberechnungs-Programm eingebunden werden.

4.2.0.1 Systemanforderungen

Zur Benutzung des Programms muss aufgrund der gewählten Programmiersprache und der objektorientierten Programmierweise folgendes gelten:

- Im Rechner muss die Software MATLAB[®] der Version 5.0 oder höher² eingerichtet sein.

Die Anforderungen an die Hardware sind durch die eingesetzte MATLAB[®]-Version bestimmt.

² MATLAB[®] verfügbar über The MathWorks, Inc. (<http://www.mathworks.com>).

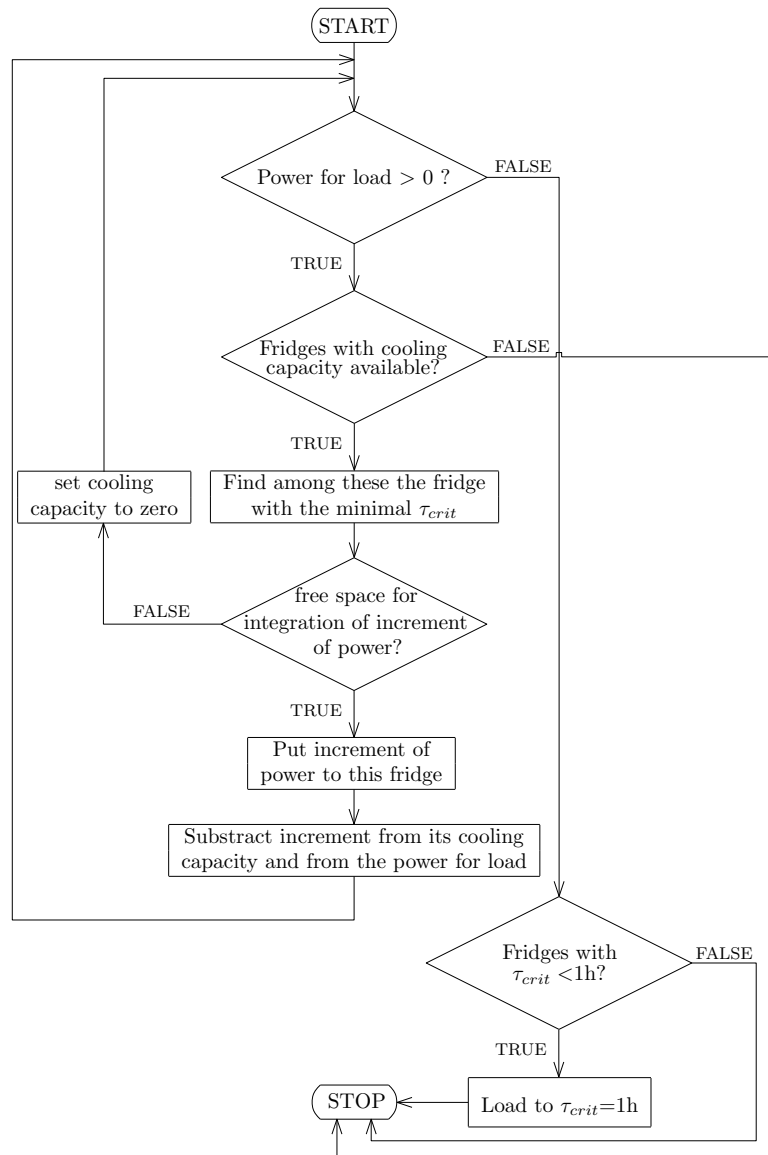


Abbildung 4.5: Flussdiagramm

4.2.0.2 Installation

Das Programm richtet sich in seiner Installation und der Benutzung nach dem allgemein gültigen Gebrauch der MATLAB[®] M-files³.

4.2.0.3 Aufruf der Simulation

Die Primärfunktion des Programms ist, wie oben schon erwähnt, die Simulation des Lastverhaltens eines Supermarkts oder mehreren Supermarktketten in einem Energieversorgungsnetz. Das setzt voraus, dass folgende grundlegende Prinzipien realisiert werden:

³ Ausführliche Information dazu findet man z.B. im [1].

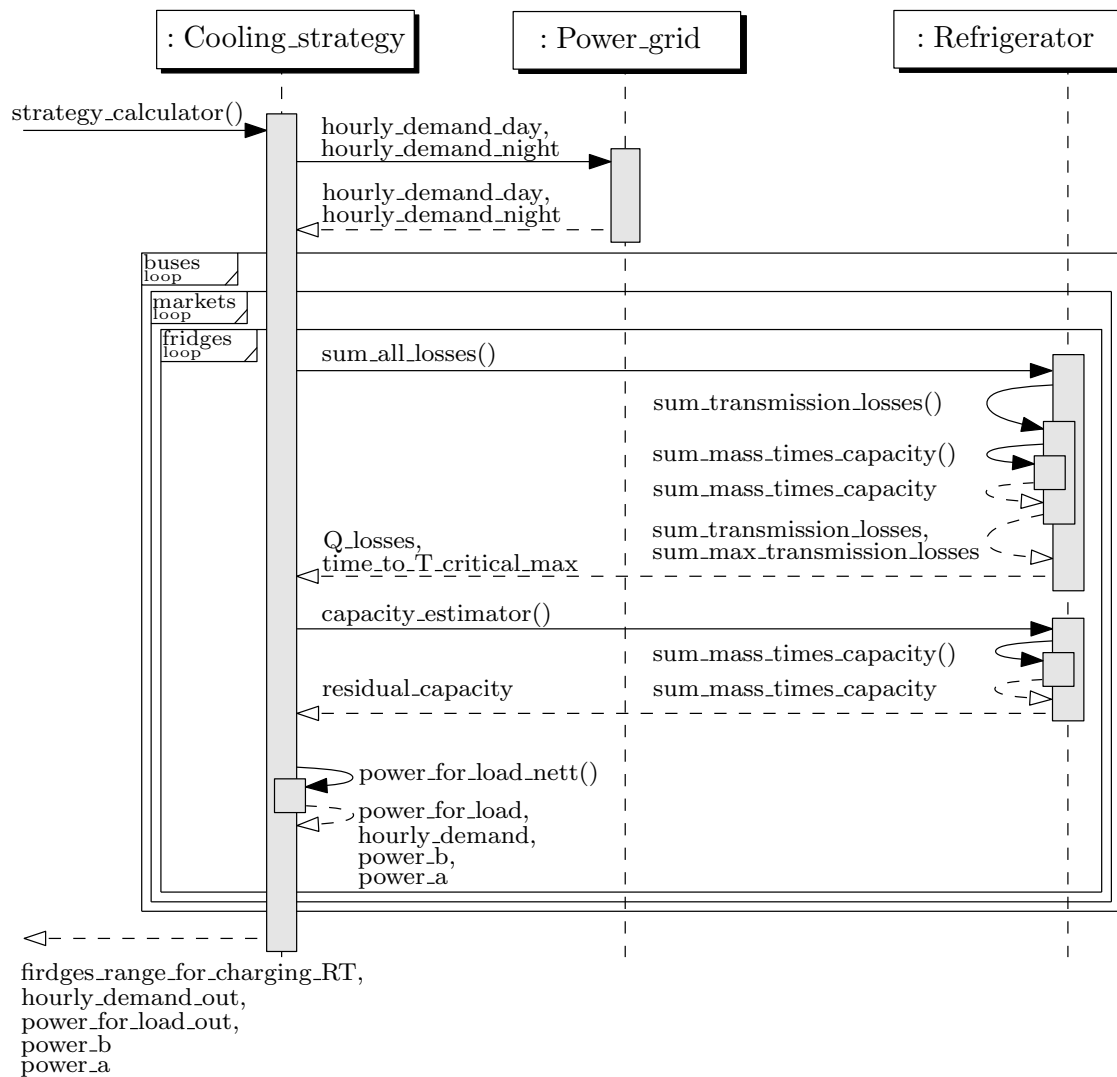


Abbildung 4.6: Sequenzdiagramm Kooperationsstrategie

1. Ein oder mehrere MATLAB[®]-Modelle von Supermärkten können erstellt und gleichzeitig an verschiedenen Knoten im Energieversorgungsnetz verwendet werden.
2. Von einem besonderen Interesse sind die Energieverbrauchsdaten zum Beispiel für eine bestimmte Kälteeinheit oder für die gesamte Kette. Die Realisierung der Speicherung dieser Daten ist zwingend.

Vorbereitung der Input-Information

Je nach Fall oder theoretischer Grundlage können die Konfigurationsdateien verändert werden. Zum Starten des Programms werden folgende Konfigurationsdateien benötigt:

das ist scheiße!

- `config_grid.m`

- config_supermarkets.m
- config_fridges.m

config_grid.m

Es ist wichtig, dass die topologische Eigenschaft des Netzes im Programm berücksichtigt werden, damit bei der Berechnung des Leistungsflusses die Laständerung, die durch den Betrieb der Kältelast entsteht, am Knoten, an dem sie verursacht wird, gezählt wird. Damit das Programm mit den für die Simulation notwendigen Informationen, die das Energienetz beschreiben, versorgt wird, ist die Form, die in MATLAB®-Code 4.1 vorgestellt wird, für die config_grid.m-Datei zwingend:

freaky sentence

MATLAB®-Code 4.1: config_grid.m

```
%% Bus, Supermarkets, Number of Supermarkets
configuration_grid = {...
    1 , {Aldi      ,      1500;      ...
        Netto     ,      500};      ...
    2 , {0        ,      0};      ...
    3 , {0        ,      0};      ...
    4 , {0        ,      0}      ...
};
```

In der **config_grid.m**-Datei wird ein $n \times 2$ -Cell Array ($n \in \mathbb{Z}_0^+$) definiert, der die Information über die Topologie des Netzes sowie die Verteilung der Kältelasten im Netz beinhaltet. Ein Cell Array ist ein Speicherobjekt, der verschiedene Datentypen unterschiedlicher Größe aufnehmen kann [1, Teil 2, Seite 15]. Ein Cell Array wird mit dem Befehl $c = \text{cell}(\dots)$ oder mit Hilfe von geschweiften Klammern $c = \{\dots\}$, wie in diesem Fall, erzeugt. Der Zugriff auf ein Cell Array wird in dem folgenden Kapitel kurz erklärt³. Jede Zeile des Cell Arrays **configuration_grid**, der in der **config_grid.m**-Datei definiert werden muss, steht für ein Knotenpunkt im Netz. In der ersten Spalte wird die Nummer des Knotens bzw. Busses gespeichert. In die zweite Spalte werden die Arten der Kältelasten und deren Anzahl an jeweiligen Knoten festgelegt. Es ist wiederum ein $m \times 2$ -Cell Array ($m \in \mathbb{Z}_0^+$). Jede Zeile dieses Cell Arrays ist für eine eigene Art Supermarktkette reserviert. In die erste Spalte kommt der Name der Supermarktkette deren Eigenschaften in der **config_supermarkets.m**-Datei vergl. MATLAB®-Code 4.2 gespeichert sind. In die zweite Spalte wird die Anzahl der Supermärkte einer Kette festgelegt die an dem bestimmten Knoten simuliert werden soll.

Hier auf Kapitel verweisen.

Um Fehler auf dieser Ebene zu vermeiden, muss die **config_grid.m**-Datei unbedingt die oben vorgestellte Form beibehalten.

config_supermarkets.m

MATLAB[®]-Code 4.2: config_supermarkets.m

```

%%      Kind of fridge      Anzahl
Aldi = { ...
        { NK_KT_S          ,    1 };      ...
        { NK_KR_V          ,    2 };      ...
        { TK_TKT_S         ,    1 };      ...
    };
Netto = { ...
        { NK_KT_S          ,    5 };      ...
    };

```

Die Kältelast in einem Supermarkt bilden die einzelnen je nach Einsatzzweck speziell dafür konstruierten Kühleinheiten. Die Anzahl, die technischen Eigenschaften und die Betriebsweise dieser Kältemaschinen können in der Realität beispielsweise aufgrund von klimatischen, wirtschaftlichen oder auf Kundenverhalten bezogenen Standortbesonderheiten Unterschiede aufweisen. Für eine breite Betrachtung des Einflusses auf ein Energieversorgungsnetz durch den Einsatz unterschiedlicher Supermarktketten ist es zweckmäßig die beschriebenen Besonderheiten im Simulationsprogramm zu verfolgen. In der Konfigurationsdatei **config_supermarkets.m** (vergl. MATLAB[®]-Code 4.2) wird festgelegt, wie die einzelnen Supermarktketten aus verschiedenen Kälteeinheiten zusammengesetzt sind. Die Gruppierung der einzelnen Kühleinheiten zu einem Modellsupermarkt erfolgt in der **config_supermarkets.m**-Datei durch Definition eines oder mehrerer Cell Arrays, die jeweils ein Supermarkt abbilden. Die Dimension eines solchen Arrays muss $i \times 1$ -Cell Array ($i \in \mathbb{Z}_0^+$) sein. Jede Zeile eines solchen Cell Arrays ist für je eine Art Kühleinheit reserviert. Die spezifischen Eigenschaften einer Kühleinheit werden in der Konfigurationsdatei **config_fridges.m** gespeichert.

Kältemaschinen beschreiben?

$i \times 1$ eigentlich noch nicht ideal, Schwachstelle, umdenken

config_fridges.m

Die Konfigurationsdatei **config_fridges.m** kann als eine Datenbank für Kälteeinheiten aufgefasst werden. In dieser Datenbank werden Informationen nach einem bestimmten Muster gruppiert und gespeichert, sodass jede Gruppe eine bestimmte Kühleinheit abbildet. Im Programm kann aus je einem dieser physikalisch fundierten Modellen⁴ ein Kühleinheit-Objekt erzeugt werden. Abhängig von der Zusammensetzung der Konfigurationsdateien **config_grid.m** und **config_supermarkets.m** wird die Zuweisung im Gesamtmodell (Anzahl in einem bestimmten Supermarkt und an einen bestimmten Knoten im Energieversorgungsnetz) durchgeführt.

Beschränkungen hier?

Am Beispiel des Modells einer steckerfertigen Normalkühltruhe⁴, abgekürzt NK_KT_S (vergl. MATLAB[®]-Code 4.3), wird im folgenden die Eingabeform eines

⁴ In dieser Arbeit werden Kühleinheiten-Modelle verwendet, die von Caroline Möller in ihrer Diplomarbeit [10] erarbeitet wurden.

solchen Datenbankeintrages in der Datei **config_fridges.m** erläutert. Zur Beschreibung einer Kühleinheit ist erforderlich, die Daten in einen 1×15 -Cell Array zu erfassen. Die Größe und die Art des Arrays ist vorgegeben durch die im Modell getroffenen Annahmen⁴ zur Anzahl und dem Typus der modellbeschreibenden Daten. Die Struktur eines Cell Arrays ermöglicht neben der zusammenhängenden Speicherung verschiedener Datentypen einen durch MATLAB® darauf direkten Zugriff.

MATLAB®-Code 4.3: config_fridges.m

```

1  % fridge configuration parameters
2  NK_KT_S = { ...
3      1, ... % 1 if plugin module or 2 if combine fridge
4      4.7e3, ... % energy consumption per day Wh/24h
5      2, ... % epsilon power quotient
6      0.66, ... % compressor quotient
7      0 ... % installed cooling power in W
8      [16.4 6.7], ... % area_wall
9      [0.38 0.38], ... % heat_transmission_coefficient
10     [19 15], ... % temperature_outside
11     [200 200], ... % masse_stored
12     [2.3 3.52], ... % specific_mass_capacity
13     -6, ... % temperature_min in Celsius
14     2, ... % temperature_max in Celsius
15     1, ... % averaged cooling room temperature in Celsius
16     0, ... % refrigerating capacity
17     };

```

das ist doch
überflüssig,
oder?

Spalte eins (vergl. Zeile 3 im MATLAB®-Code 4.3) Die Unterscheidung in steckerfertige Kälteanlagen und Kälteverbundanlagen im Programm ist notwendig, da bei der Berechnung der Kälteverluste unterschiedliche Verfahren zu Grunde gelegt werden. Diese Differenzierung erfolgt durch die Zuweisung einer Eins für die steckerfertige Einheit und einer Zwei für eine Einheit an Verbundanlage in der ersten Spalte im Array .

Spalte zwei Die durchschnittliche elektrische Energieaufnahme ist eine Kennzahl in Watt pro 24 Stunden, die durch die Hersteller mit Hilfe eines genormten Verfahrens ermittelt wird und in den technischen Blättern angegeben werden muss.

Wo wird das
Verwendet?
Hier schreiben

Spalte drei In diese Spalte wird die Leistungszahl eingetragen.

Spalte vier Der Anteil des Verdichters an dem Tagesenergieverbrauch der Kühleinheiten wird Leistungszahl genannt. Aus der Gleichung (2.11) im Kapitel 2 geht hervor, wie diese Kennzahl in die Berechnung eingeht. Die Leistungszahl wird in der Spalte vier festgehalten.

Spalte fünf Werden anschlussfertige Kälteaggregate zur Kühlung von Räumen verwendet, so wird die installierte Kälteleistung in Kilowatt in die Spalte fünf anstatt einer Null eingetragen.

Spalte sechs Ein Teil der Wärmeverluste ist auf die Transmissionsverluste durch die Wände zurückzuführen. Die Anzahl der Wände muss für die Rechnung berücksichtigt werden, Sie wird in die sechste Spalte eingetragen.

Spalte sieben Für die Transmissionsverlustberechnung sind unter anderem die Flächengrößen und die jeweiligen Wärmedurchgangskoeffizienten der einzelnen Wände maßgebend⁵. Für je eine Flächengröße in Quadratmeter ist in der Spalte sieben des NK_KT_S-Arrays ein Spaltenplatz im $1 \times w$ -Zeilenarray reserviert. Die w steht für die Anzahl der Wände.

Spalte acht In der achten Spalte werden in einem weiteren $1 \times w$ -Zeilenarray die mit den Flächen korrespondierende Wärmedurchgangskoeffiziente⁵ in gleicher Reihenfolge gespeichert.

Spalte neun Die Temperatur außerhalb der Kühleinheit bestimmt⁵ die Größe der Transmissionsverluste. In der neunten Spalte werden nach dem gleichen Prinzip und in der gleichen Form wie im vorhergegangenen Eintrag die Außentemperatur für jede Wand eingetragen.

Spalte zehn Die Abhängigkeit der Temperaturänderung von der in einer Kühleinheit deponierten Masse an Waren und deren spezifische Wärmekapazität ist aus der Gleichung (2.1) im Kapitel 2 ersichtlich. In der Realität werden in einem Kühlschrank gewöhnlich mehrere verschiedene Lebensmittel gekühlt. Möchte man gemischte Beladung simulieren, so sind die Massen in der Spalte zehn im NK_KT_S in der gleichen Art und Weise wie in der Spalte sieben acht oder neun zu einzutragen. Der Eintrag erfolgt in Kilogramm.

Spalte elf Zu jeder in der Spalte zehn eingetragener Massezahl muss in der Spalte neun nach dem Beispiel der Spalte acht die spezifische Wärmekapazität eingetragen werden.

Spalte zwölf In der Realität fällt die Spanne für Variation der Innentemperatur in einem Lebensmittelkühlschrank eher gering aus, da die Temperatur auf einem bestimmten Niveau gehalten werden muss, damit die Lebensmittel maximal lange frisch bleiben können. Möchte man die Kälte im Lebensmittel speichern, darf die Lebensmitteltemperatur einen bestimmten Bereich nicht verlassen. In der Spalte zwölf im NK_KT_S wird die untere Temperaturgrenze in Grad Celsius festgelegt.

Spalte elf In der Spalte elf wird die obere Temperaturgrenze in Grad Celsius festgelegt.

Spalte zwölf Die Bedeutung⁶ der mittleren Kühlraumtemperatur ist kurz im Kapitel 2 angeschnitten. Notiert wird diese Kennzahl in der Spalte zwölf im NK_KT_S eingetragen.

⁵ Der mathematische Zusammenhang wird im Kapitel 2 in der Gleichung (2.3) deutlich gemacht.

⁶ Ausführlich begründet wird das in [10].

Spalte dreizehn Keine Ahnung

Klären was das ist.

Die main.m Datei

4.2.0.4 Deklamation der berechneten Daten

Die inhaltliche Auswertung der Simulation ist dem Benutzer vorbehalten. Darum müssen die Ergebnisse der Simulation in einer anwenderfreundlichen Form zugänglich gemacht werden. Durch sinnvolle graphische Darstellung der untersuchten Größen kann zum Beispiel die Überprüfung, die Veranschaulichung oder die Bestimmung der funktionalen Abhängigkeit dieser durchgeführt werden.

Das ist noch nicht richtig im Programm implementiert. Das muss DU NOCH MACHEN!

Graphische Ausgabe

Wichtig!!! hier wird nicht der Quellcode erklärt, sondern die Darstellung der Ausgabe.

Funktion mit für die graphische Ausgabe schreiben

4.2.0.5 Tabellarische Ausgabe im Command-Window und Exel

Funktion für tabellarische Ausgabe schreiben

ANHANG A

Anhang

A.1 Anhang Example section

This for other bibtex stye file: only [10] one author [1] and many authors [14].

Literaturverzeichnis

- [1] Angermann, Beuschel, Rau und Wohlfarth. *MATLAB - Simulink - Stateflow: Grundlagen, Toolboxen, Beispiele*. Oldenbourg, München, 6., aktualis. aufl. edition, 2009. 26, 28, 33, 34
- [2] Jaime Arias. *Energy Usage in Supermarkets - Modelling and Field Measurements*. text, Royal Institute of Technology, KTH, Energy Technology ; Stockholm, 2005. 3, 7, 34
- [3] Bernhard Lahres und Gregor Rayman. *Objektorientierte Programmierung. das umfassende Handbuch*. Galileo computing. Galileo Press, Bonn, 2., aktualisierte und erw. aufl. edition, 2009. 4, 34
- [4] EnergieAgentur.NRW. Energieeffizienz im Lebensmittel-Einzelhandel. URL: <http://www.energieagentur.nrw.de/unternehmen/page.asp?TopCatID=3695&CatID=3721&RubrikID=3743> (Zugriff am 20. Juli 2011). 3, 34
- [5] Michael Kauffeld. *Stand der Technik von Supermarktkälteanlagen. Umwelteinfluss und Entwicklungspotential*. 2008. 9
- [6] Martin Kleimaier. Netzintegration von Strom aus erneuerbaren Energiequellen: Zunehmende dezentrale Einspeisung erfordert eine Umrüstung der Netze. *Energy 2.0*, 1(7):45–47, 2008. 34
- [7] F.A.T.M. Ligthart. Untersuchung zur Möglichkeit einer Abdeckung von Kühl- und Tiefkühlmobiliar in Supermärkten . Technical Report ECN-E-08-009, ECN Energy in the Built Environment, 2008. 3
- [8] Elke Lorenz, Johannes Hurka, and Detlev Heinemann. Solarleistungsvorhersage zur netzintegration von solarstrom. 24. *Symposium Photovoltaische Solarenergie : 04. - 06. März 2009, Kloster Banz, Bad Staffelstein*, page 6, 2009. 2, 34
- [9] Sean McGowan. Supermarket refrigeration going natural. *HVAC&R Nation*, pages 8–9, 2007. 3, 34
- [10] Caroline Möller. Spezifikation und Simulation einer Kältelast mit Kältespeicher im Energieversorgungsnetz. Diplomarbeit, Technische Universität Berlin, Berlin, Juli 2010. 7, 29, 31, 33, 34
- [11] P. Pepper. *Programmieren lernen: Eine grundlegende Einführung mit Java*. Springer, 2008. 4
- [12] Arnd Poetzsch-Heffter. *Konzepte Objektorientierter Programmierung: Mit Einer Einführung in Java*. Springer, Berlin, 2009. 4, 34

-
- [13] J. Probst. Ein Netzwerk für effiziente Kältetechnik. URL: <http://www.energieagentur.nrw.de/unternehmen/page.asp?TopCatID=3695&CatID=3721&RubrikID=3743> (Zugriff am 20. Juli 2011), 2009. 3
- [14] Risto Ciconkov und Arnd Hilligweg. Simulationsprogramme für Kälteanlagen Einsatz in Ausbildung und Praxis. *Technik im Bau, Fachzeitschrift für Technische Gebäudeausrüstung*, 77(3):64–70, 2004. 7, 33, 34
- [15] W. Schweizer. *MATLAB kompakt*. Oldenbourg Wissensch.Vlg, 2009. 5
- [16] F. Steimle, H. Kruse, E. Wobst, and et al. Energiebedarf für die technische Erzeugung von Kälte. Technical Report Statusbericht Nr. 22, Deutscher Kälte- und Klimatechnische Verein (DKV) e.V., Stuttgart, 2002. 3
- [17] Steve Völler. *Optimierte Betriebsführung von Windenergieanlagen durch Energiespeicher Elektronische Ressource*. Universitätsbibliothek Wuppertal, Wuppertal, 2010. 2, 34
- [18] Michael Weigend. *Objektorientierte Programmierung mit Python 3. Einstieg, Praxis, professionelle Anwendung*. mitp-Verl., Heidelberg u.a., 4., aktualisierte Aufl. edition, 2010. 4, 34