

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Erik Mukk

Stiimulõppel põhinev nutika kodu
küttelahendus Energy2D
simulatsioonitarkvaras

Bakalaureusetöö (9 EAP)

Juhendaja: Jakob Mass, MSc

Tartu 2020

Stiimulõppel põhinev nutika kodu küttelehendus Energy2D simulatsioonitarkvaras

Lühikokkuvõte:

Käesoleva bakalaureusetöö käigus loodi masinõppel põhinev küttesüsteem. Süsteem põhineb stiimulõppel, mis õpib ja automatiseerib kütmist Energy2D tarkvaral jooksvates simulatsioonides. Loodud lahenduse hindamiseks võrreldakse stiimulõppel valminud süsteeme etalonlahendusega, milleks on termostaadil põhinev kütmine. Valminud stiimulõppel põhinevad lahendused suudavad ruumis olevat küttekeha iseseisvalt juhtida. Süsteemi on enne treenimist võimalik konfigureerida nii, et see arvestaks erineval määral elektri hinnaga. Konfiguratsioonist sõltuvalt muutub ka ruumi kütmissprofiil. Sellise lahendusega suudeti saavutada 5%-15% sääst rahas võrdluses termostaatlahendusega, säilitades sama hea või isegi parema sihttemperatuuri hoidmise oskuse. Töö lõpeb loodud lahenduse kitsaskohtade identifitseerimisega ning nende lahendamiseks näpunäidete andmisega. Samuti pakutakse välja tuleviku arendussuunad.

Võtmesõnad:

Masinõpe, IoT, tark kodu, stiimulõpe, simulatsioon, kütmine

CERCS: P175 - Informaatika, süsteemiteooria

Reinforcement learning based smart home heating solution in Energy2D simulation software

Abstract:

During the course of this bachelor thesis, a machine learning based heating system was created. System is based on reinforcement learning, which learns and automates heating in simulations running on Energy2D software. To evaluate the system, solutions based on reinforcement learning are compared to a benchmark, which is thermostat based heating. Built reinforcement learning based solution is able to operate the heater in the room autonomously. By configuring the system before training in a way that it depends on electricity price, environment heating profile changes. With this reinforcement learning based system, a 5%-15% savings in money spent on heating was achieved compared to the benchmark, whilst being as good or even better at holding target temperature of the room. At the end of the thesis, shortcomings of this system are identified and tips to fix them are given. Also, future works are proposed.

Keywords:

Machine learning, IoT, smart home, reinforcement learning, simulation, heating

CERCS: P175 - Informatics, systems theory

Sisukord

1	Sissejuhatus	4
2	Teooria	6
2.1	Masinõpe	6
2.1.1	Masinõppe liigid	6
2.1.2	Stiimulõpe	6
2.1.3	Q-õpe	7
2.2	Ülevaade kütmist simuleerivatest programmidest	9
2.2.1	Energy2D	10
2.2.2	EnergyPlus	11
2.2.3	Simulatsioonitarkvara valik	12
3	Sarnased lahendused	13
3.1	MavHome	13
3.2	Google DeepMind AI	13
3.3	Süva stiimulõppel põhinev süsteem	14
3.4	Lahenduste võrdlus	15
4	Metoodika	17
4.1	Nõuded tarkvarale	17
4.2	Arhitektuuri kirjeldus	17
4.3	Ülevaade Energy2D keskkonast	21
4.3.1	Energy2D keskkond	21
4.4	Ülevaade Q-õppe süsteemist	23
4.4.1	Q-tabel	23
4.4.2	Tasud ja Q-väärtused	23
4.4.3	Q-õppe süsteemi treenimine	26
5	Valminud lahenduse analüüs	28
5.1	Termostaat- ehk etalonlahendus	28
5.2	Temperatuuriga arvestav RL lahendus	29
5.3	Temperatuuri ja elektri hinnaga arvestav RL lahendus	31
5.4	Arutelu	34
6	Kokkuvõte	37
	Lisad	40
	I. Q-õppe koodirepositoorium	40
	II. Muudetud Energy2D koodirepositoorium	40
	III. litsents	41

1 Sissejuhatus

Tark kodu on leidnud käsitlust Feng jt uurimuses [1] kui keskkond, mis püüab parandada sealsete inimeste elukvaliteeti erinevate toimigutega. Kvaliteeti parandavateks tegevusteks on näiteks erinevate koduseadmete juhtimine. Selleks juhtimiseks võib olla nii kaugjuhtimine inimese poolt, kes ei pruugi viibida piirkonnas, kui ka täielik automatiseeritus, kus keskkonnas viibija ei pea mitte midagi tegema.

Chifor jt on kirjutanud [2], et tark kodu on asjade interneti (inglise keeles *Internet of Things* ehk IoT) lahendus. Nende sõnul seisneb IoT põhimõte selles, et erinevad seadmed, teenused ja süsteemid on omavahel interneti läbi ühenduses. Selle abil saavad nad üksteise vahel andmeid jagada. Targa kodu süsteemis on asjadeks erinevad sensorid ning aktuaatorid. Feng jt [1] lisasid, et interneti abil on need kõik omavahel ühenduses ning sensorid koguvad andmeid. Saadud andmeid töödeldakse keskses arvutis ja väljastatakse mingi tulem. Töödeldud tulem tõlgendatakse ümber käsuks, mis saadetakse seadmele, mis viib leitud tegevuse läbi. Üheks selliseks tegevuseks on näiteks kütmise automatiseerimine.

Kütmise automatiseerimine ning selle nutikaks tegemine võimalab energiat säästa, mis on praeguses olukorras oluline mitmetel põhjustel. Üks selline põhjus on Euroopa Liidu energiatõhususe direktiiv [3]. Euroopa komisjoni andmetel tarbivad Euroopa Liidus rajatud hooned 40% liidus toodetud energiast ning tektiavad 36% CO₂-heitest. Samuti on direktiivis välja toodud, et ligi 35% hoonetest on vanemad kui 50 aastat. Nende hoonete energiatõhususe parandamisega oleks võimalik vähendada nii energia tarbimist kui ka CO₂-heidet ligi 5%. Samuti leidsid Mosser jt uurimuses [4], et kodu automatiseerimine säästab nii energiat kui ka raha. Letiti, et kokkuhoid igakuiste energiaarvete pealt oli 5-15%. Targa kodu küttelahendus ei saa muuta hoonete ehitust ning kasutatud materjale ja omakorda sellest tingitud energiakulu. Küll aga suudab kontrollida kütmist ning seda vajadusel peatada, mis omakorda vähendab pikemas perspektiivis energiakulu.

Töö üks eesmärk on leida eelmises lõigus välja toodud probleemidele praktiline lahendus. Lahendusest peab selguma kas ja kui palju on võimalik kütmist automatiseerides raha säästa. Selle jaoks loob töö autor masinõppel põhineva küttelahenduse. Seda süsteemi ei ehitata füüsiliselt, vaid keskkond luuakse simulatsiooniprogrammis. Programmiga simuleeritakse soojusenergia hajumist ruumis, energia kadu läbi hoone erinevate pindade ning väliskeskkonna mõju köetavale keskkonnale. Selleks, et programmis jooksvas keskkonnas kütmist reguleerida, loob töö autor programmi, mis reguleerib küttekehi. Programm kasutab otsuste langetamiseks masinõpet ning täpsemalt selle alamosa stiimulõpet (inglise keeles *reinforcement learning*). Valik stiimulõppe kasuks langetati sellepärast, et stiimulõppe suudab ise keskkonna peal tegevusi läbi viies õppida, mis on õige ja vale. See ei vaja kasutajapoolset sisendit, mis muudab süsteemi trennimise mugavaks.

Simulatsioonides simuleeritakse kütmist Energy2D [5] keskkonnas ning koostatakse tulemuste analüüs, et leida vastuseid järgnevatele küsimustele:

- Kas ja mis moel on masinõppel põhinev süsteem parem kui tavaline termostaat?
- Kui palju on võimalik säästa energiat ja raha?
- Millised keskkonna parameetrid on olulisemad?
- Milline on parameetrite mõju tulemustele?

Potentsiaalne rahaline sääst leitakse elektri börsihindade põhjal, mis on saadud Nord Pool'i andmetest ¹.

Töö 2. peatükk on orienteeritud teooriale. Autor toob täpsemalt välja, mis on tark kodu. Samuti on seal ülevaade masinõppest laiemalt. Täpsemalt tegeletakse masinõppe ühe osaga, milleks on stiimulõpe ning omakorda selle algoritmiga, milleks on Q-õpe. Töö 3. peatükis toob autor välja sarnased lahendused ning nende tööpõhimõtted ja kirjeldused. 4. peatükk koosneb targa kodu implementatsioonist, kasutatud simulatsiooniprogrammi keskkondadest ja Q-õppe algoritmist - valemist, tasudest ning tabelist. 5. peatükis võrreldakse tulemusi ning analüüsitakse, miks need on just sellised. Viimane peatükk on kokkuvõtlik peatükk, kus antakse näpunäiteid edasisteks uurimusteks.

¹<https://www.nordpoolgroup.com/Market-data1/Dayahead/Area-Prices/EE/Hourly/?view=table>

2 Teooria

Selles peatükis antakse ülevaade töö teoreetilisest poolest. Esimeses alampeatükis käsitletakse nii masinõpet üldisemalt kui ka stiimulõpet põhjalikumalt ning selle algoritmi Q-õpet. Teises alampeatükis tuuakse ülevaade kahest simulatsiooniprogrammist, millega on võimalik läbi viia kütmise simuleerimist.

2.1 Masinõpe

Masinõpe on arvutiteaduse osa, mille eesmärk on arvutile õpetada, kuidas õppida ja käituda, olles selleks programmeerimata. Sellise funktsionaalsuse realiseerimiseks analüüsitakse andmeid ning luuakse mudeleid, mis õpivad läbi kogemuste. Mõningad alad, kus masinõpe on laialdast kasutust leidnud, on keeletöötlus, bioinformaatika, pilditöötlus ja tuvastus ning otsingumootorid. [6]

2.1.1 Masinõppe liigid

Masinõpe jaotub omakorda kaheks alamosaks - juhendatud ja juhendamata õpe.

Juhendatud õpe seisneb õppimises selliste andmetega, mis on sildistatud. Näiteks tekstide teemat kindlaks tegeval süsteemil on treeningandmete hulgal üheks treenitavaks elemendiks paragrahv. Selle elemendi juurde on kirjutatud, mis teema kohta antud paragrahv käib [7]. Singh jt [8] toovad välja, et juhendatud õppe korral luuakse algoritmid, mis suudavad luua soseid ning hüpoteese ja selle põhjal ennustada, mis juhtub andmetega, mida algoritm veel näinud ei ole. Selle saavutamiseks kategoriseeritakse andmeid. Üks juhendatud õppe implementatsioonidest on tehisnärvivõrkudel põhinev masinõppe. Masinal luuakse neuronite struktuur, mis sarnaneb inimese ajule ja loob samuti seoseid.

Juhendamata õpe seisneb õppimises selliste andmetega, kus kindlat õiget vastust ei ole antud. Algoritm otsib ise seaduspärasusi andmetes. Kui selline algoritm peaks pakkuma vastust küsimusele, siis ei saa öelda, et vastus oleks kas õige või vale. Algoritm on leidnud, et küsimus sarnases varem nähtud seaduspärasusega ning pakkus selle põhjal vastuse [7]. Juhendamata õppe hulka kuulub ka stiimulõpe, sest stiimulõppel tegevusi tegeva agent ei tea, mis on õige ja mis on vale.

2.1.2 Stiimulõpe

Stiimulõpe ehk inglise keeles *reinforcement learning* on üks juhendamata õppe liikidest. Sutton ja Barto [9] käsitlevad stiimulõpet kui protsessi, kus õpitakse ära tegevuste jada parima tulemuse saamiseks. Kui käsil on keskkond X , siis leitakse, milline tegevus on sellele vastavuses nii, et tegevusest saadav numbriline kasu oleks suurim. Sel ajal, kui

stiimulõppe algoritm õpib tegevusi, ei ole sellele öeldud, mida teha. Algoritm peab ise katsetama ning leidma, milline tegevus annab suurima numbrilise kasu. Siiski ei ole oluline, et ainult ühele tegevusele järgneks suurim kasu, vaid tegevuste lõpuni tuleb kasu maksimeerida.

Kasu maksimeerimise näitlikustamiseks vaatleme sellist olukorda. On tegevused ja nendele vastavad punktisummad $A = 10, B = 20, C = -20, D = 100$. Ühtegi tegevust ei või teha enam kui üks kord ning tegevust D ei saa teha, kui tegevust C pole tehtud. Seega alustatakse tegevusest A , mille tagajärjel kogu punktisumma on 10. Seejärel tehakse samm B , mille tulemusena kasvab punktisumma 30 punkti. Tegevused A ja B tehti, kuna nende eest ei saa negatiivset kasu ehk kahju. Nüüd jõutakse sammuni C , mis tekitaks kahju. Kuna tegemist on stiimulõppega, siis samm C siiski võetakse, sest muidu ei saaks teha sammu D . Selle tagajärjel punktisumma küll esmalt langeb 10 punkti, kuid peale sammu D on see 110. Kui samm C oleks vahele jäetud, siis oleks sel hetkel küll kasu maksimeeritud, aga järgmine samm oleks jäänud tegemata ning kogu punktisumma oleks olnud vaid 30 punkti.

2.1.3 Q-õpe

Q-õpe tuleneb ingliskeelsest nimest *Q-learning*. Selle algoritmi mõtles välja 1989. aastal Christopher J.C.H. Watkins [9].

Q-õppel arvutatakse tegevuse tasu Q-funktsiooni abil. See funktsioon vaatleb keskkonna praegust ning eelnevat olekut, leiab nende Q-väärtused ja leiab praeguse oleku põhjal tasu [10]. Seejärel antakse leitud väärtused koos veel mõne konstandiga uue Q-väärtuse arvutamiseks järgnevale funktsioonile:

$$\underbrace{Q^{uus}(S_t, A_t)}_{\text{Q-tabeli sissekande väärtus}} \leftarrow \underbrace{Q(S_t, A_t)}_{\text{Vana q väärtus}} + \underbrace{\alpha}_{\text{Õppimiskiirus}} \cdot \underbrace{\left(r_t + \underbrace{\gamma}_{\text{Soodustus}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{Optimaalne uue q väärtus}} - \underbrace{Q(S_t, A_t)}_{\text{Vana q väärtus}} \right)}_{\text{Ajaline erinevus}} \quad (1)$$

Q-funktsioonis (1) määrab γ ehk soodustus selle, kui palju pööratakse uue Q-väärtuse arvutamisel tähelepanu potentsiaalselt parimale (ehk suurimale) võimalikule uuele Q-väärtusele. Soodustus on vahemikus $[0, 1]$. Mida väiksem on soodustus, seda rohkem pööratakse tähelepanu hetkelisele tasule. Suurema soodustuse puhul on süsteemile oluline maksimeerimida tasu pikas perspektiivis. [11]

Q-tabel kujutab endast tabelit, kust saab järele vaadata Q-väärtuseid ning neid sinna kirjutada. Q-väärtus on Q-funktsiooni (1) poolt välja arvatud väärtus, kus "Q" tähendab kvaliteeti (inglise keeles *quality*). Käesolevas töös on Q-tabel räsitabelina struktureeritud. Tabeli võtmeteks on treenitava keskkonna kõik seisundid, mille peal Q-algoritmiga otsuseid tehakse [10, 12]. Olgu meil keskkond, kus mõõdetakse temperatuuri ning selle põhjal langetatakse otsuseid. Olgu selles keskkonnas võimalikud tegevused "minna õue" ja "mitte minna õue". Keskkonna võimalik temperatuur on kas 0, 5, 10, 15, 20 või 25

kraadi. Siis luuakse alustuseks selline Q-tabel, kus kõik Q-väärtused on alguses nullid (vt. Tabel 1).

Tabel 1. Q-tabeli näide: Q-tabeli algseis.

Temperatuur	Q-väärtused ["mitte minna õue", "minna õue"]
0	[0, 0]
5	[0, 0]
10	[0, 0]
15	[0, 0]
20	[0, 0]
25	[0, 0]

Tegevuse valimisega igal temperatuuril kaasneb ka mingi tasu. Kui valiti sobiv tegevus, siis üldjuhul on Q-funktsiooniga (1) leitud tasu positiivne. Vastasel juhul on tasu negatiivne või ainult väiksem kui õige tegevuse tasu. Arvutatud tasu listakse Q-tabelisse. Stiimulõppe alapeatükis (vt. peatükk 2.1.2) on mainitud, et algoritm katsetab iseseisvalt erinevaid tegevusi. Katsetamine seisneb selles, et valitakse suvaline tegevus kõikide võimalike tegevuste hulgast. Selle tagajärjel vaadeldakse läbi rohkem olukordasid ning Q-tabel muutub täpsemaks tegevuste pakkumisel. Aja möödudes katsetamise hulk väheneb ning rohkem pööratakse tähelepanu Q-tabeli väärtustele. Selleks ajaks ei ole Q-väärtused enam ainult nullid, vaid õigete tegevuste väärtused on suuremad kui valede tegevuste väärtused. Seega, kui algoritm on treenimise lõpetanud, on selgelt näha, kumb tegevus mingi keskkonnaseisundi juures on kasumlikum (vaata Tabel 2). Treenimise ilmestamiseks tuleb vaadelda järgnevat pseudokoodi [12].

Algoritm 1: Q-tabeli treenimine

```

1 luuakse Q-tabel Q[keskkonnaOlekud,keskkonnaTegevused] algväärtustega;
2 while treenimine käib do
3   vali ja tee tegevus a;
4   jälgi tasu r ja uut olekut s';
5   arvutada Q-väärtus:  $Q[s,a] = Q[s,a] + \alpha(r + \gamma \max_{a'} Q[s',a'] - Q[s,a])$ ;
6   s = s';
```

Treenimise lõppedes ning tabelit kasutades valitakse tegevus, mille Q-väärtus on suurim. Näiteks kui välistemperatuur on 25 kraadi, siis tabeli järgi oleks vaja minna õue, sest $0.23 < 8.32$ ning valitakse tegevus, mille Q-väärtus on 8.32. Kui juhtub nii, et kõik Q-väärtused on mingis keskkonna seisundis samad (kui temperatuur on 10 kraadi), valitakse endiselt maskimaalne väärtus. Maksimaalse väärtuse valimisfunktsiooni eripärast tingituna võib selleks olla ükskõik milline nendest väärtustest.

Tabel 2. Q-tabeli näide: Treenitud Q-tabel.

Temperatuur	Q-väärtused ["mitte minna õue", "minna õue"]
0	[5.23, 1.36]
5	[4.23, 2.19]
10	[3.34, 3.34]
15	[3.12, 4.18]
20	[1.32, 6.39]
25	[0.23, 8.32]

2.2 Ülevaade kütmist simuleerivatest programmidest

Uurimuse praktiliste katsete läbiviimiseks otsustati simulatsiooniprogrammi kasuks, et katseid käsitsi mitte läbi viia. Selle põhjus on ajapuuduses ning katsete ebatäpsuses. Selleks, et masinõppe algoritmi treenida on vaja jookustada tuhandeid iteratsioone. See võtab päriselus liiga kaua aega. Lisaks tehes vea koodis, mis võib ilmnedas alles peale pikka treenimist, on palju aega kasutult kulutatud. Lisaks sellele, on võimalik programmi abil treenides identset keskkonda korrata igal treenimisel, mis päriselus oleks peaaegu võimatu. Programmile on võimalik ette anda argumente, et saaks läbi viia mitu katset korraga, kuid erinevate parameetritega. See on samuti päriselus ühes treeningkeskkonnas võimatu. Programmides on võimalik simuleerida nii keskkonda, kus katseid läbi viiakse, kui ka keskkonnategureid. Need tegurid on kas ajaloolised andmed või valemite abil välja arvutatud näitajad. Samuti võetakse arvesse ehitiste eripära, näiteks nende seinade paksust ning soojusjuhtivust.

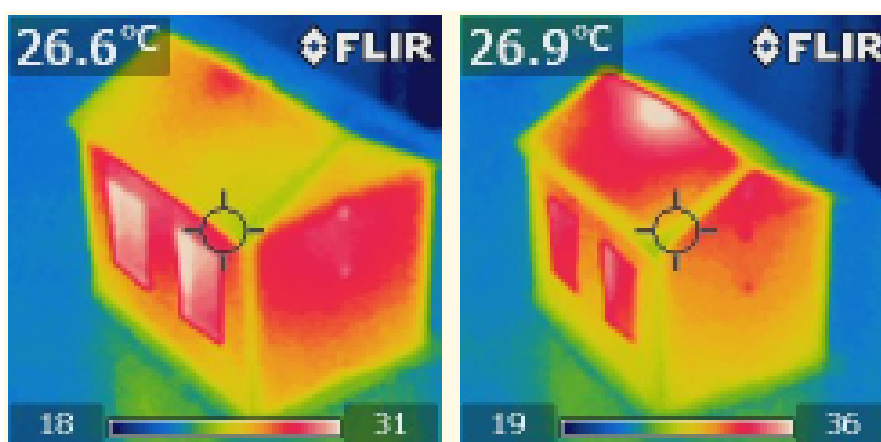
Käesolev töö käsitleb kaht simulatsiooniprogrammi - Energy2D ja EnergyPlus. Valik langetati nende programmide vahel, sest programme otsides leiti need ühena esimesetest. Uurimise käigus leiti, et programmide funktsionaalsused on selle uurimuse jaoks sobilikud. Olulised funktsionaalsused on järgnevad:

- käsurealt käivitatav;
- kütmist simuleeriv;
- öö ja päeva tsükli vaheldumise simuleerimine;
- hoone ja seda ümbritseva keskkonna loomine programmis

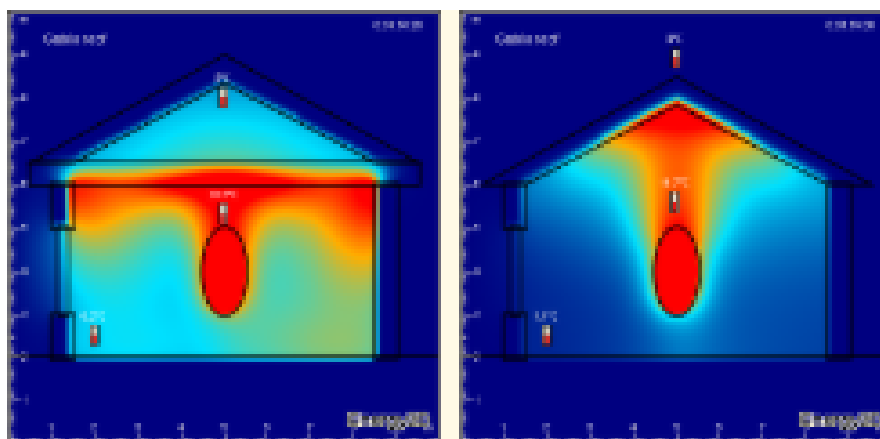
Teised leitud programmid, mis suudavad simuleerida soojuse levimist, kuid käesolevas töös ei ole käsitletud on näiteks MATLAB [13] ja SolidWorks [14]. Neid ei käsitletud, kuna algkood ei ole kättesaadav. Lisaks nendele mainitakse edasises töös korra ka ResiSim simulaatorit. ResiSim ei ole internetist kättesaadav ning selletõttu käsitlust ei leia.

2.2.1 Energy2D

Energy2D [15] on programmeerimiskeeles Java kirjutatud programm soojusülekanne simuleerimiseks - juhtivuse, konduktsiooni ja kiirguse näitlikustamiseks. Selle abil on võimalik läbi viia eksperimente, et saada kinnitust teaduslikele hüpoteesidele. Samuti võib seda kasutada erinevate inseneriprobleemide lahendamiseks, kasutamata keerulist matemaatikat. Energy2D keskkonnas kasutatavaid simulatsioone ei tohi kindlasti lugeda 100% tõetruudeks. Põhjus on selles, et Energy2D simulatsioonide füüsikaliste nähtuste energijahtivus on väga täpne, kuid konvektsiooni ja kiirguse simuleerimine ei ole täiesti täpne. Siiski on kogu süsteem piisavalt täpne, et seda käesolevas töös kasutada [5, 15]. Täpsuse iseloomustamiseks ja võrdlemiseks tuleb vaadata allpool olevaid Jooniseid 1 ja 2.



Joonis 1. FLIR soojuskaamera pilt päris majast. [5]



Joonis 2. Energy2D soojuse levimine analoogses majas. [5]

Joonisel 1 on kujutatud päris maja. Nendest majadest vasakpoolsel on toas lagi ning on kujutatud soojusenergia levikut. Parempoolsel majal puudub lagi ning on näha suuremat soojusenergia kadu läbi katuse. Vastavalt ehitatud majad on kujutatud ka Joonisel 2, kuid need on valmistatud Energy2D programmis.

Simulatsiooniprogrammile on võimalik ette anda XML-formaadis keskkond või luua see interaktiivselt seal samas (XML-keskkonna leiab autori koodirepositooriumist, vt. Lisa 1). Kasutajal on võimalik muuta igasuguseid parameetreid. Olgu nendeks sellised hoone parameetrid nagu seinade paksus ja soojusmahtuvus, õhu tihedus ja soojusmahtuvus, küttekehade asukoht, võimsus ja olek. Või siis väliskeskkonna parameetrid nagu päeva ja öö vaheldumine, päikese intensiivsus ja nurk ning ka simulatsiooni ajasamm, mis määrab programmi töötamise kiiruse.

Energy2D juures on ka puudujääke. Kuna keskkond modelleeritakse kahemõõtmeliselt, mis piirab päriselule vastava keskkonna kujutamist, tuleb teha olulisi lihtsustusi. Samuti on puudu logimisvõimekus, mis tuleb töö autoril ise lisada, et saada tagasisidet keskkonnalt. Logimisvõimekuse jaoks tuli töö autoril aluskoodi muuta.

2.2.2 EnergyPlus

EnergyPlus [16] on hoone energiatarbe simuleerimiseks loodud programm. See on laialdaselt leidnud kasutust nii inseneride, arhitektide kui ka teadlaste käes. EnergyPlus abil on võimalik simuleerida ka ümbritsevat keskkonda, kuid seda on võimalik teha ainult etteantud andmete põhjal. Programm ise ei suuda ümbritsevat keskkonda luua. Hoones on võimalik simuleerida kütmist, jahutamist, ventilatsiooni, valgustust ja veekasutust. Selles on võimalik läbi viia simulatsioone kolmemõõtmelistes keskkondades. Kuna hoone sees on võimalik simuleerida väga paljusid erinevaid tegevusi, on programmile võimalik ette anda ka rohkelt parameetrid, mis konfigureerivad erinevaid tegevusi.

Hoone, mida EnergyPlus simuleerib tuleb luua mudelleerimisprogrammi kasutades. Selliseks programmiks on näiteks SketchUp [17].

2.2.3 Simulatsioonitarkvara valik

Selle uurimuse seisukohalt on autoril kasulikum kasutada Energy2D programmi. Kõige olulistemaks põhjusteks, miks autor just selle programmi valis, on järgnevad:

- Kuna kumbki programm ei ole käsurealt juhitav, tuli aluskoodi muuta, et seda masinõppega siduda. Energy2D on kirjutatud keeles Java, millega on autor tuttav, erinevalt EnergyPlus'ist, mis on kirjutatud C keeles;
- Energy2D koodibaas on mahult väiksem ning lihtsamini arusaadav kui EnergyPlus'i koodibaas;
- Energy2D suudab ise luua andmeid keskkonna kohta, kuid EnergyPlus'il selline võimekus puudub;
- Energy2D keskkonda on palju lihtsam luua ning ei vaja modelleerimisoskuseid.

3 Sarnased lahendused

Käesolevas peatükis tuuakse ülevaade kolmest masinõppel põhinevast küttesüsteemist. Mainitakse ära milliseid sensoreid süsteemid kasutavad ning milleks need on suutelised. Samuti on kirjeldatud tulemusi ning võrreldakse lahendusi omavahel.

3.1 MavHome

Cook jt koostasid süsteemi MavHome [18] (nimi tuleneb ingliskeelsest lühendist **M**anaging **A**n **I**ntelligent **V**ersatile **H**ome ehk eesti keelde tõlgituna mitmekülgse intelligentse kodu juhtimine), mille eesmärk oli luua keskkond, mis suudab ise otsuseid langetada. Need tegevused on välja arvatud kahel erineval masinõppe algoritmil. Esimene ennustab inimese käitumist selles keskkonnas ning teine õpib kodu juhtimise eeskirja. Lisaks teoreetiliselt õigete otsuste arvutamisele on MavHome ka päriselt välja ehitatud süsteem, mis koosneb peale masinõppe veel andmebaasidest, robotikast ja erinevatest sensoritest.

Selles süsteemis ennustatakse inimese käitumist selle põhjal, mida inimene on varem kodus teinud ning milliseid seadmeid kasutanud. On väga oluline, et ennustusalgoritmide vigade arv oleks minimaalne, sest see tooks kaasa energiakulu suurenemise. Samuti tooks see inimesele ebameeldivusi, sest süsteem võib käituda vastupidi inimese ootustele. Kasutatud ennustusalgoritm on *Active-LeZi* algoritm.

MavHome's arvutatakse kodu juhtimise sesiukohalt olulised otsused välja stiimulõppe abil. Arvutus tehakse koheselt, kui vähemalt ühe sensori näit muutub. Seejärel arvutatakse tasufunktsiooni põhjal välja üks Q -väärtus ning see listakse Q -tabelisse. Peale õppimist leitakse optimaalne tegevus a_t järgneva funktsiooni abil

$$a_t = \operatorname{argmax} Q(s_t, a),$$

kus Q märgib Q -tabelit, a võimalike tegevuste kaalude massiivi ning s_t Q -tabeli rida, mis vastab keskkonna seisundile. argmax funktsioon tagastab masiivis olema suurima väärtusega elemendi indeksi, mis viitab ka tegevusele, millega see on seotud.

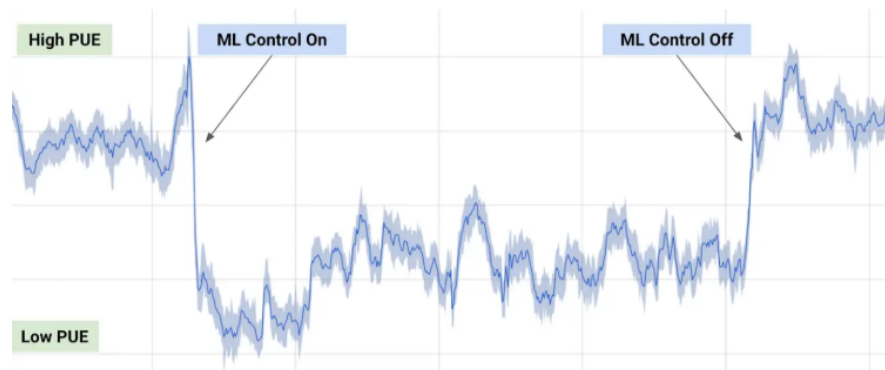
MavHome süsteemi testiti nii simulaatoris kui ka päriselt välja ehitatuna. Simulatsiooniprogrammina oli kasutusel ResiSim, kus loodi osa MavHome keskkonnast. Päriselt ehitati süsteem välja Texas, Arlingtoni Ülikoolis. Seal oli katseisikuks "Bob", kelle tegevusi jälgiti ning ennustati.

3.2 Google DeepMind AI

2016. aastal ilmus artikkel [19] sellest, kuidas Google DeepMind AI vähendas Google'i andmekeskuse jahutamiseks kuluvat energiat 40% võrra. Selleks pandi Google DeepMind nime kandev tehisintellekt andmekeskust jahutama. Selle jaoks treeniti tehisnärvivõrke ajalooliste andmete peal, mida oli kogunenud tuhandetelt sensoritelt andmekeskuses.

Sellisteks andmeteks oli näiteks temperatuurid, energiatarve, jahutusumpade andmed ja veel enam. Kuna igal andmekeskusel on erinev arhitektuur ning need asuvad erinevates keskkondades, siis ühe keskuse andmetel treenitud mudelit ei oleks saanud kasutada teistes keskustes, kui ei oleks kasutatud tehisnärvivõrke.

Mudelit testiti töötaval andmekeskusel ühel päeval. Leiti, et hoiti kokku 40% energiat, mis oleks kulunud jahutamiseks (vt. Joonis 3).



Joonis 3. Google'i andmekeskuse energiatõhusus.

Jooniselt 3 on näha, et ühe päeva lõikes langes PUE (inglise keeles **P**ower **U**sage **E**ffectiveness ehk energiatõhusus) märgatavalt, kui tehisnärvivõrkudel põhinev juhtimine sisse lülitati. Samuti PUE tõusis koheselt kui masinõppel juhtimine välja lülitati.

3.3 Süva stiimulõppel põhinev süsteem

Süva stiimulõppel (inglise keeles *Deep Reinforcement Learning*, edaspidi DRL) süsteem [20] loodi selleks, et saada mööda tavalise abistatud õppe probleemidest. See probleem on vajadus suur arvutusvõimsuse järele.

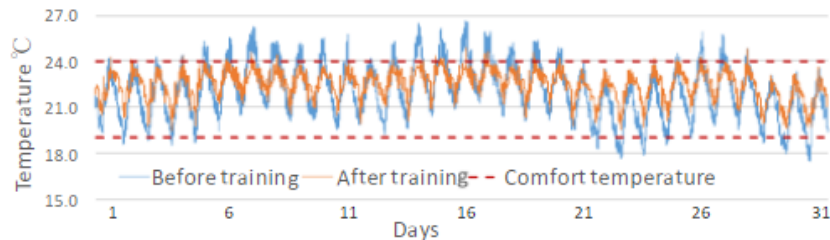
DRL peal põhinev süsteem seati üles EnergyPlus keskkonnas ning selle jaoks loodud honnega. Seda süsteemi treeniti Los Angeles' e ja Riverside' 100 kuu ilmaandmete peal, mis saadi *National Solar Radiation Data Base*'ist. Simulatsiooni elektrikulu hindamiseks saadi andmed *Southern California Edison*'ist. Näitude saamiseks keskkonnast kasutati virtuaalseid temperatuurisensoreid, mis olid EnergyPlus süsteemis. Kokku loodi 3 keskkonda, milles oli vastavalt 1, 4 ja 5 vaadeldavat ala.

Simulatsioonide ja DRL mudeli täpsuse hindamiseks võrreldi saadud tulemusi reeglitel põhineva HVAC (inglise keeles *H*eating, *V*entilation and *A*ir *C*onditioning ehk kütmine, ventilatsioon ja õhukonditsioneer) süsteemi tulemustega. Reeglite näide on järgnev:

- Kui õhutemperatuur on üle 24°, alustada jahutamist.

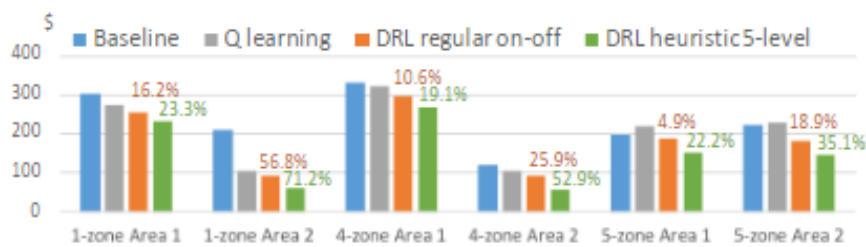
- Kui õhutemperatuur on alla 19°, alustada kütmist.

Leiti, et DRL algoritmil põhinev süsteem suudab temperatuuri hoida soovitud vahemikus (19° kuni 24°) ning seda isegi juba treenimise käigus (vt. Joonis 4).



Joonis 4. DRL süsteemi temperatuurid.

Samuti leiti, et energia kokkuhoid oli kuni 70% võrreldes tavapärase kütmisega (vt. Joonis 5). Joonisel 5 märgib sinine värv tavapärast kütmist, hall värv Q-õppel põhinevat kütmist ning oranž ja roheline värv DRL algoritmil põhinevat kütmist. Oranž tähistab ventilatsiooni sisse lülitamist ühel kiirusel ja välja lülitamist ning roheline näitab ventilatsiooni välja lülitamist ning sisse lülitamist neljal erineval kiirusel.



Joonis 5. DRL süsteemi energiakulu.

3.4 Lahenduste võrdlus

Kõik eelnevalt mainitud lahendused on masinõppel põhinevad. Nii Google Deep Mind kui ka DRL lahendus kasutavad lisaks stiimulõppele ka tehiskäitumise võrke. Nii DeepMind kui ka DRL lahendus tõid märgatava kokkuhoiu energias, vastavalt 40% ja isegi kuni 70%. MavHome süsteemi kohta ei ole teada kui suure energiasäästu see tõi.

Autori tööga on kõige sarnasem just MavHome lahendus, kuna need mõlemad põhinevad täielikult stiimulõppel, täpsemalt Q-õppel. Nii MavHome kui ka Google DeepMind kasutavad keskkonna parameetrite kogumiseks kümneid, sadu ja tuhandeid

sensoreid, aga autori süsteemis on kuni 5 erinevat sensorit. MavHome süsteemis jälgiti ka inimese käitumisharjumusi ning koostati nende põhjal ennustusi. Autori töös inimese tegevusi ei jägita.

4 Metoodika

See peatükk annab ülevaate tehtud töö metoodikast. Töö eesmärk on luua tarkvaralahendus, millega simuleerida ja automatiseerida kütmist. Tarkvara abil on võimalik treenida ja jookustada Q-õppe algoritmi. Selles peatükis käsitletakse nõudeid tarkvarale, tarkvara arhitektuuri, simulatsioonikeskkonda Energy2D tarkvaras. Samuti käsitletakse Q-õpet ning sellega seonduvat.

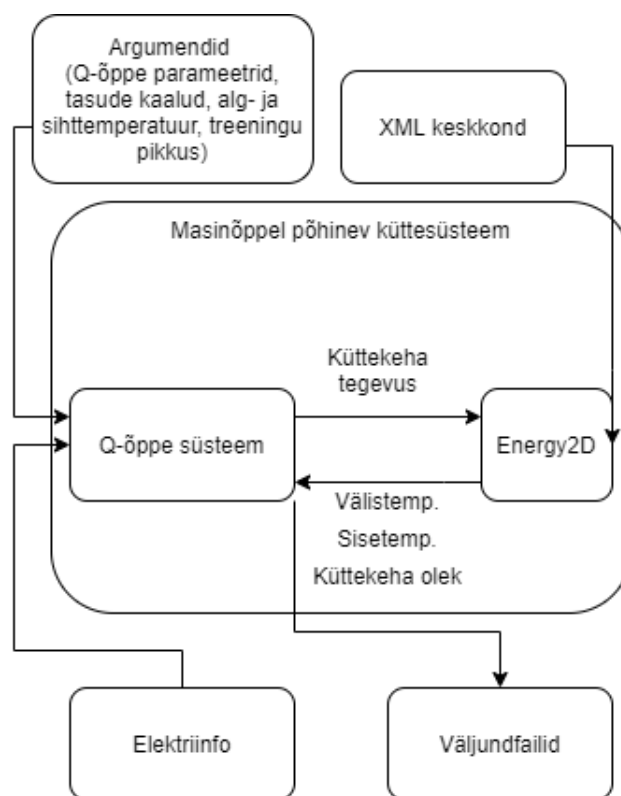
4.1 Nõuded tarkvarale

Valminud tarkvara peab vastama järgnevatele nõuetele:

1. tarkvara peab saama käsurealt juhtida;
2. tarkvara laeb XML formaadis Energy2D keskkonna;
 - tarkvara peab suutma kasutada erinevaid keskkondasid;
 - keskkonnas on temperatuurid vahemikus [0.0, 40.0].
3. tarkvara peab looma JSON formaadis logifaiile järgnevate andmetega:
 - Energy2D keskkonna kütmise aeg perioodide kaupa;
 - Energy2D keskkonna temperatuur perioodide kaupa;
 - Energy2D keskkonna elektritarve perioodide kaupa;
 - Q-õppe tasu perioodide kaupa;
 - kasutatud käsurea argumendid;
 - treenitud Q-tabel;
4. tarkvara peab suutma Energy2D keskkonda juhtida nii treenitud kui ka treenimata Q-tabeli põhjal treenimise eesmärgil;
 - tarkvara saab sisendiks JSON formaadis Q-tabeli;
5. tarkvara peab suutma Energy2D keskkonda juhtida Q-tabelita sisseehitatud küttekeha termostaadi põhjal;

4.2 Arhitektuuri kirjeldus

Tarkvara koosneb kahest suuremast osast. Nendeks on **Energy2D tarkvara** ja **Q-õppe süsteem**. Süsteemi üldine skeem on toodud Joonisel 6.



Joonis 6. Masinõppel põhineva kütteleahenduse üldine skeem.

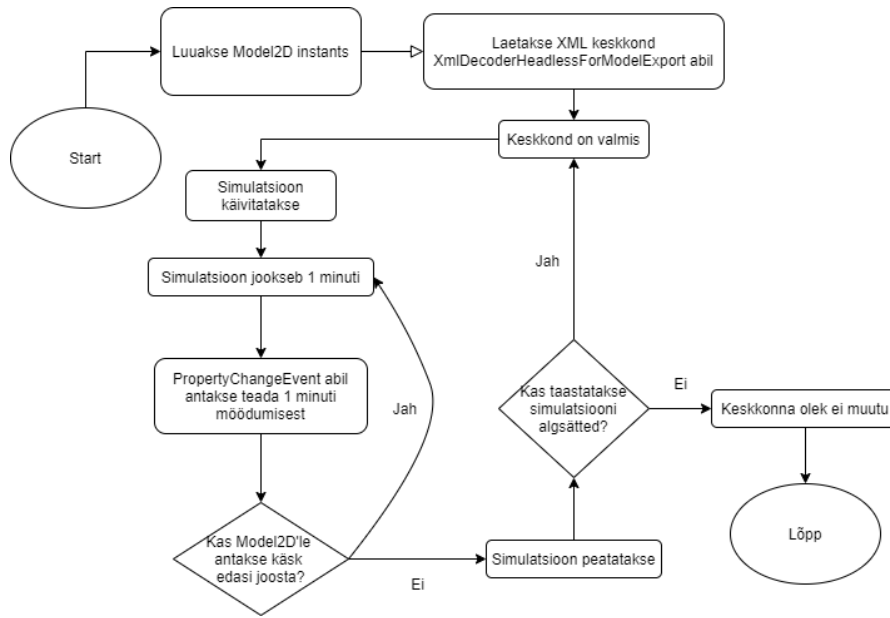
Energy2D tarkvara [21] aluskoodi on muudetud, et kasutada programmi ilma graafilise kasutajaliideseta. Graafilise liideseta on võimalik programmi kasutada serveris töötaval virtuaalmasinal, et simulatsioonid töötaksid kiiremini. Siiski tuli säilitada programmi soojussimulatsiooni funktsionaalsused. Muudetud kood on olemas töö autori Githubi repositooriumis (vt. Lisa 2).

Selleks, et autor saaks Energy2D programmi panna tööle virtuaalmasinas ning selle peale katseid läbi viia, tuli aluskoodi muuta. Graafiline kasutajaliides tuli eemaldada, kuid programmi funktsionaalus säilitada.

Originaalsest Energy2D aluskoodist on eemaldatud System2D klass, millest käivitati graafilise kasutajaliidesega programm. Seal loodi Model2D klassi instants, mis vastutab simulatsiooni eest. Uue Model2D klassis tööpõhimõte on lihtsustatud kujutatud Joonisel 7. Model2D klassi muudeti nii, et see teeks järgnevat, mida varem ei teinud:

- laeb sisse XML formaadis keskkonna;
- käivitab keskkonna;

- peatab keskkonna;
- taastab keskkonna algsätteid;
- teavitab Java PropertyChangeEvent abil iga minuti läbimist simulatsioonis.



Joonis 7. Model2D klassi lihtsustatud tööpõhimõte.

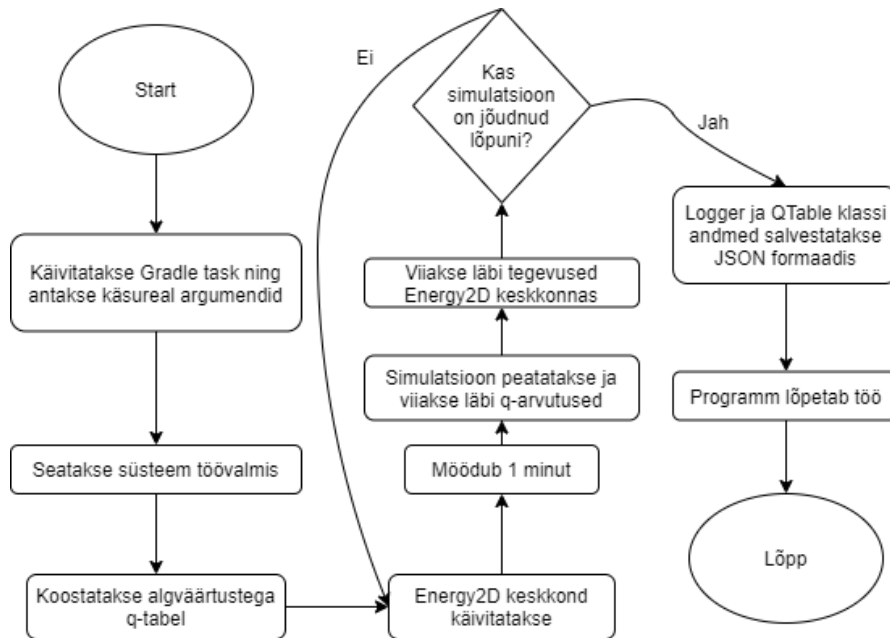
Samuti loodi `XmlDecoderHeadlessForModelExport` klass, mis on alternatiiv `XmlDecoder` klassile, mida kasutati graafilise liidesega programmis Energy2D keskkonnafailide laadimiseks. Muudetud kood pakendati JAR failiks, et seda kasutada Q-õppe süsteemis. Selle jaoks tuleb importida loodud JAR failist `Model2D` ning instantseerida see objekt.

Q-õppe süsteem on programmeerimiskeeles Java kirjutatud projekt. See kasutab Gradle't², et laadida sisse muudetud Energy2D programm ning viia läbi Q-õppeks vajalike tegevusi. Q-õppe programmi tööpõhimõte on kujutatud Joonisel 8 ning koosneb neljast suuremast osast:

- logimine;
- matemaatilised abifunktsioonid;

²Gradle kodulehekülg: <https://gradle.org/>

- Q-õppe algoritm;
- Energy2D keskkonna juhtimine.



Joonis 8. Lihtustatult q-õppe süsteemi tööpõhimõte.

Logimise jaoks on kasutusel Logger klass, milles hoiustatakse Java HashMap andmestruktuure. Andmed, mida seal hoiustatakse, on kütmise aeg, elektrikasutus, temperatuurid ja Q-õppe tasud. Logger klassi logiandmeid hoiustavad isendimuutujad salvestatakse Jackson'i³ abil JSON formaadis.

Matemaatiliste abifunktsioonide jaoks on kasutusel Normalization klass. Normalization klassis olevaid meetodeid kasutatakse arvuliste näitude normaliseerimiseks. Selle abil on võimalik teisendada üks arvuvahemik teise. Näiteks vahemik 0...40 vahemikku 0...1, mis on vajalik Q-õppe tasude arvutamiseks.

Q-õppe jaoks on kasutusel QTable, Environment klassid. Environment klassi kasutatakse peamiselt logimise eesmärgil, kuid mõned q-õppe poolt arvatud tegevused viiakse läbi seal. QTable klassi kasutatakse ainult Q-õppe jaoks. Selles objektis:

- koostatakse Q-tabel;
- täidetakse Q-tabelit;

³Jackson: <https://github.com/FasterXML/jackson>

- arvutatakse Q-väärtuseid;
- küttekeha juhtimine Energy2D keskkonnas Q-tabeli põhjal.

Energy2D keskkonna juhtimiseks kasutatakse Observer klassi. See klass saab Energy2D programmilt iga minuti tagant signaali, et 1 simulatsiooniminut on möödunud. Seejärel kutsub see välja Q-õppe meetodeid ning keskkonna juhtimise meetodeid.

4.3 Ülevaade Energy2D keskkonast

Energy2D keskkonnad on XML failid. Faile kasutatakse Model2D'sse Energy2D keskkonna laadimiseks. Need määravad ära simuleeritavad keskkonna järgnevad parameetrid:

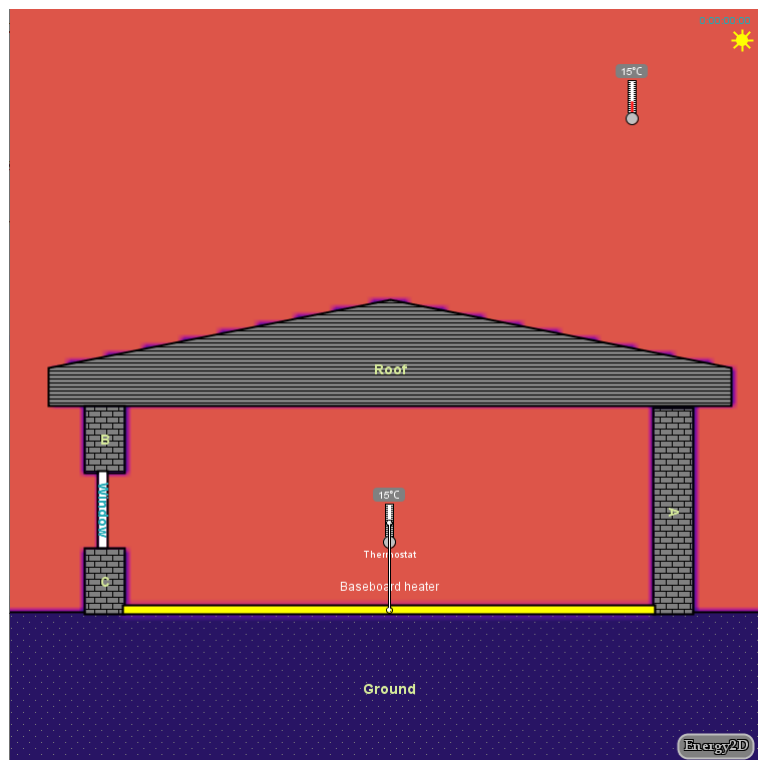
- hoone ja selle omadused;
- küttekehad ning nende omadused;
- keskkonna füüsikalised parameetrid:
 - öö ja päeva tsüklite möödumine;
 - päikese intensiivsus ning nurk;
 - keskkonna soojusmahtuvus.
- simulatsiooni ajasamm.

4.3.1 Energy2D keskkond

Joonisel 9 on välja toodud Energy2D programmis koostatud keskkond. Selles asub üks hoone, mille ainukeseks küttekehaks on põrandaküte. Keskkonna parameetrid on toodud Tabelis 3. Keskkonna eesmärk on simuleerida väga lihtsat keskkonda, mida köetakse ühe küttekehaga. See küttekeha on põrandaküte. Põrandakütte kasuks valiti sellepärast, et see kütab ühtlaselt tervelt põrandapinnalt. Terve pinna kütmine hoiab ära temperatuuride erinevuse samal kõrgusel põrandast, mis teeks Q-õppe süsteemi treenimise keeruliseks. Küttekeha võimsuseks on valitud 2.5kW. Võimsus on valitud ainult näitlikustamiseks, et arvutada kütmisele kulunud raha. Kuna keskkond on kahemõõtmeline, on keskkonna sügavuseks arvutuste lihtsustamiseks võetud 1 meeter. Tegelikult võib küttekeha võimsus olla erinev ning see on määratav Energy2D programmis küttekeha parameetreid muutes.

Tabel 3. Simulatsioonikeskkond 1.

Parameetri nimi	Parameetri väärtus
Algtemperatuur	15°C
Sihttemperatuur	22°C
Simulatsiooni algusaeg	00:00
Simulatsiooni kestus	24h
Päikese tõus	06:00
Päikese loojumine	18:00
Hoone laius	7m
Hoone kõrgus	2.7m
Hoone materjal	Telliskivi
Küttekeha	Põranadaküte
Termomeetri kõrgus hoone põrandast	1m
Termomeetri kaugus hoone seinadest	3.5m
Küttekeha võimsus	2.5kW



Joonis 9. Põranadaküttega Energy2D simulatsioonikeskkond termostaadiga toa keskel.

4.4 Ülevaade Q-õppe süsteemist

Käesolev alapeatükk annab ülevaate loodud Q-õppe algoritmist. Käsitletakse tasusid, mille abil arvutatakse Q-väärtused. Samuti käsitletakse Q-tabelit, mis hoiab arvutatud väärtuseid ning mille põhjal tehakse keskkonnas tegevusi.

4.4.1 Q-tabel

Käesolevas uurimuses on Q-tabel räsitabel, mis sisaldab võtme ja väärtuse paarisid. Q-tabel ei pea olema ühedimensiooniline, vaid saab olla ka mitmemõõtmeline maatriks. Räsitabel valiti andmestruktruuks sellepärast, et see kasutab vähem mälu kui mitmemõõtmeline maatriks. Samuti on sellest lihtsam andmeid küsida ning neid sinna sisestada.

Kasutatavas Q-tabelis on võtmeteks Java sõned. Sõne on koostatud vaadeldavatest parameetritest, milleks on alati sisetemperatuur ja välistemperatuur. Tuues simulatsiooni sisse ka elektrihinna, sisaldab sõne ka elektrihinna ning küttekeha olekut, milleks on 0 ja 1. Kõik parameetrid on alakriipsuga ühendatud ning unikaalsed. Näiteks, kui sisetemperatuur on 21°C, välistemperatuur 8°C, elektrihind 25€/ühik ja küttekeha passiivne, siis on võtmeks 21_8_22_0.

Q-tabeli väärtusteks on Java massiiv, mille andmetüüp on float ning suurus 2. Massiivis esimesel kohal olev väärtus näitab küttekeha sisselülitamise Q-väärtust. Teisel kohal olev väärtus küttekeha väljalülitamise Q-väärtust. Q-väärtused massiivis ei pea olema unikaalsed.

Selleks, et tabeli järgi keskkonda juhtida on vaja teada keskkonna näitajaid. Vaatleme siin eelnevat näidet, milles koostati võti 21_8_22_0. Treenitud tabelist otsitakse üles antud võtmele vastav massiiv - olgu selleks $[-0.18978801, -0.142121]$. Küttekeha sisse lülitamise väärtuseks on -0.18978801 ning välja lülitamise väärtuseks -0.142121 . Õige ehk kasulik tegevus leitakse nende arvude põhjal. Kasutatakse maksimeerimise funktsiooni, mis käib läbi kõik väärtused massiivis ning tagastab suurima väärtuse indeksi massiivis. Tagastatakse indeks 1, mis näitab, et küttekeha tuleb välja lülitada. Seda sellepärast, et massiivis teisel kohal (ehk indeksil 1) on välja lülitamise Q-väärtus ning $-0.18978801 < -0.142121$. Kui väärtused oleksid olnud võrdsed, oleks valitud esimene indeks ehk 0. Selle tagajärjeks oleks olnud küttekeha sisse lülitamine. Nüüd on keskkonda juhitud ning n -ajasammu pärast kordub samasugune tegevus.

4.4.2 Tasud ja Q-väärtused

Tasud on arvulised suurused, mille põhjal arvutatakse Q-väärtuseid. Need on arvutatud keskkonnalt saadud sisendite põhjal, mis on käesolevas töös keskkonna sise- ja välistemperatuur ning elektrihind. Tasu eesmärk on näidata kui palju kasu või kahju tegevuse tegemine keskkonnale annab. Kui keskkonna tasu pannakse kokku mitmest eraldi ar-

vutatud tasust, võib neid läbi korrutada mingi kaaluga, et iga tasu olulisust muuta. Kui keskkonnalt saadavad sisendid on kogu aeg samal skaalal nagu näiteks temperatuurid, siis ei pea algandmeid töötleva. Juhul, kui on ka muud näitajad nagu on käesoleva töö mõndades simulatsioonides elektri hind, tuleb väärtused viia ühisele skaalale ehk normaliseerida.

Tasu sõltub alati temperatuurist ning elektri hinna lisamise ka elektri hinnast. Juhul, kui elektri hinda ei vaadelda, leitakse tasu järgnevalt:

1. arvutatakse keskkonna sihttemperatuuri ja hetketemperatuuri vahe;
2. leitakse absoluutväärtus vahest;
3. absoluutväärtus korrutatakse läbi arvuga -1 ;
4. tulemus normaliseeritakse skaalale $-1.0...0.0$;
5. tulemus korrutatakse läbi temperatuuri kaaluga.

Kui vaadeldakse ka elektri hinda, leitakse temperatuurist sõltuv tasu järgnevalt:

1. arvutatakse keskkonna sihttemperatuuri ja hetketemperatuuri vahe;
2. leitakse absoluutväärtus vahest;
3. absoluutväärtus korrutatakse läbi arvuga -1 ;
4. tulemus normaliseeritakse skaalale $-1.0...0.0$;
 - normaliseerimine seab suurusd $0.0...40.0$ skaalale $-1.0...0.0$, kus -1.0 on vastavuses arvuga 40.0 ning 0.0 arvuga 0.0
5. tulemus korrutatakse läbi temperatuuri kaaluga.

Elektrihinnast sõltuv tasu leitakse järgnevalt:

1. leitakse elektri hind kütmise ajal;
 - kui ei kõeta, valitakse elektri hinnaks 0 ;
2. tulemus normaliseeritakse skaalale $-1.0...0.0$;
 - normaliseerimine seab suurusd $min_elektri_hind ... max_elektri_hind$ skaalale $-1.0...0.0$, kus -1.0 on vastavuses maksimaalse elektri hinnaga ning 0.0 minimaalsega;
3. tulemus korrutatakse läbi elektri hinna kaaluga.

Ainult temperatuuri vaadeldes jääb tasuks temperatuuriga leitud tasu. Vaadeldes ka elektrit, liidetakse temperatuuri ja elektri tasud kokku. Tasud antakse edasi Q-funktsioonile (2), mille põhjal leitakse Q-väärtus. Arvutatud Q-väärtus sisestakse tabelisse vaadeldava võtme masiivis sobivale kohale.

$$\underbrace{Q^{uus}(S_t, A_t)}_{\text{Q-tabeli sissekande väärtus}} \leftarrow \underbrace{Q(S_t, A_t)}_{\text{Vana q väärtus}} + \underbrace{\alpha}_{\text{Õppimiskiirus}} \cdot \underbrace{\left(r_t + \underbrace{\gamma}_{\text{Soodustus}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{Optimaalne uue q väärtus}} - \underbrace{Q(S_t, A_t)}_{\text{Vana q väärtus}} \right)}_{\text{Ajaline erinevus}}$$

(2)

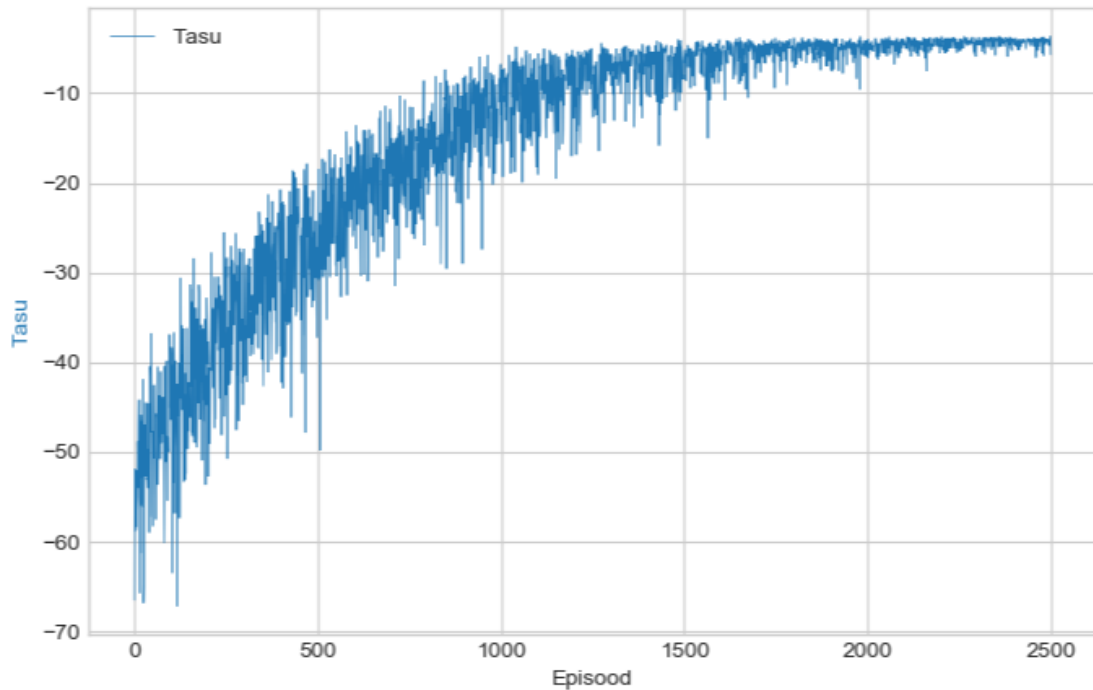
Käesoelvas töös on γ ehk soodustuse algväärtuseks valitud 0.9, et maksimeerida tasu pikemas perspektiivis. Õppimiskiiruseks ehk α on valitud 0.1. See näitab, kui palju pööratakse Q-väärtust arvutades tähelepanu uuele väärtusele, mis vanale Q-väärtusele juurde liidetakse. Samuti on Q-õppe süsteemis kasutusel selline muutuja nagu ϵ ning konstant *EPS_DECAY* (inglise keeles *epsilon decay*, eesti keeles *epsiloni langus*). Selle põhjal valitakse, kas Q-õppe keskkonda uurida ehk teha suvaline tegevus või valida tegevus Q-tabeli põhjal. Kui genereeritud juhunumber on suurem kui ϵ , valitakse tegevus Q-tabeli põhjal. Vastasel juhul uuritakse keskkonda ehk valitakse suvaline tegevus kõikide tegevuste seast. Peale igat iteratsiooni ehk 24 tundi treenides saab ϵ uue väärtuse, mis on arvutatud nii: $\epsilon = \epsilon * EPS_DECAY$. Selle tagajärjel ϵ väheneb ning haktakse rohkem tähelepanu pöörama Q-tabeli väärtustele. Allpool on toodud pseudokood ülalkirjeldatud tegevuste illustreerimiseks (vt. Algoritm 2).

Algoritm 2: Q-tabeli treenimistsükkel.

```
1  $\epsilon = 0.9$  ;
2 EPSDECAY = 0.999 ;
3 while 24h pikkune episood treenib do
4   a = keskkonna seisund;
5   b = eelmise keskkonna seisundi võti Q-tabelis ;
6   sisetemp = a.sisetemperatuur ;
7   elektri hind = a.elektri hind ;
8   norm_sisetemp = normaliseeri(sisetemp) ;
9   norm_elektri hind = normaliseeri(elektri hind) ;
10  preaguse_keskkonna_q_väärtus = null ;
11  if juhunumber >  $\epsilon$  then
12    tegevuste_massiiv_keskkonnal_olekus_a = Q_tabel.get(a) ;
13    preaguse_keskkonna_q_väärtus =
14      max(tegevuste_massiiv_keskkonnal_olekus_a) ;
15  else
16    tegevuste_massiiv_keskkonnal_olekus_a = Q_tabel.get(a) ;
17    preaguse_keskkonna_q_väärtus =
18      random(tegevuste_massiiv_keskkonnal_olekus_a) ;
19  eelmise_keskkonna_seisundi_q_väärtus = Q_tabel.get(b) ;
20  tasu = norm_sisetemp * temperatuuri_kaal + norm_elektri hind *
21    elektri hinna_kaal ;
22  uus_q_väärtus = eelmise_keskkonna_q_väärtus + õppimiskiirus * (tasu +
23    soodustus * preaguse_keskkonna_q_väärtus -
24    eelmise_keskkonna_seisundi_q_väärtus) ;
25   $\epsilon = \epsilon * \text{EPSDECAY}$  ;
26  algab uus 24h pikkune episood ;
```

4.4.3 Q-õppe süsteemi treenimine

Q-õppe süsteemi treenides täidetakse Q-tabel leitud Q-väärtustega. Kui süsteem õpib korrektselt, siis leitud Q-väärtused on kasvava trendiga. Käesolevas töös treeniti igat Q-õppe süsteemi 2500 episoodiga ehk 2500 24 tunni pikkuse tsükliga. Iga episoodi lõpus liideti kokku saadud tasud, et luua treenimist iseloomustav graafik. Üks sellistest graafikutest on toodud Joonisel 10.



Joonis 10. Stiimulõppe treenimise tasud.

Jooniselt on näha, kuidas treenimise algfaasis on tasu -120, aga lõpuks on see jõudnud umbes -7 juurde. Kõik tasud on negatiivsed sellepärast, et nii elektri kui ka temperatuuri tasu tuleb vahemikust $[-1.0, 0.0]$. Treenimisel on tasude tõus enamjaolt ühtlane, kuid näiteks 1300. episoodi juures on üks mitmest ebakorrapärasusest. Keskmine tasu on umbes -15, kuid mõndadel episoodidel langeb see -30 juurde. -15 tähendab, et süsteem võtab väga tihti korrektse tegevuse Q-tabelist. Languse põhjus võib olla see, et juhunumbri generaator genereeris enam kui pooltest kordadest 24h jooksul sellise numbriga, mis on väiksem kui ϵ . Selle tagajärjel valiti enam kui pooltel kordadest suvaline tegevus, mis võis olla keskkonna hetkeseisu jaoks ebasoodne.

5 Valminud lahenduse analüüs

Käesolevas peatükis viiakse läbi andmeanalüüs. Sõnastatakse analüüsi eesmärgid. Võrreldakse lahendusi nii omavahel kui ka baasnäitajaga, milles on termostaadi põhine kütelahendus. Edaspidi kasutatakse väljendit "RL lahendus", mis viitab stiimulõppe põhiste lahendustele. Analüüsiga proovitakse leida vastuseid järgnevatele küsimustele:

- kas ja mis moel on RL lahendus parem kui termostaatlahendus?
- kui palju on võimalik säästa energiat ja raha?
- kuidas muutuvad tulemused, kui muuta RL lahendust treenides erinevaid parameetreid?

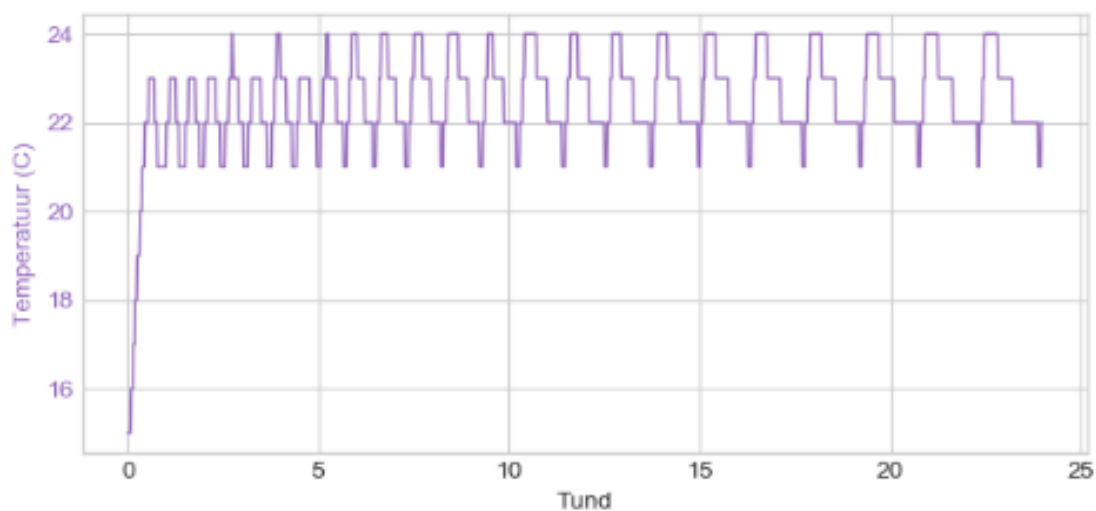
5.1 Termostaat- ehk etalonlahendus

Termostaatlahendus on termostaadiga juhitud küttekehaga köetud simulatsiooni tulemus. Termostaat oli Energy2D programmis konfigureeritud hoidmaks temperatuuri 22°C ümber. Kõikumisruumi oli 1°C. Seega, termostaat küttis kuni 23°C, siis jahtus kuni 19°C ning alustas uuesti kütmist. Selline tsüklus toimus kogu 24h vältel. Kütmise rahaline kulu on leitud etteantud 24h elektrihindade põhjal, mis on saadud NordPool'ist⁴. Igal simulatsioonil olid elektri hinnad samad, et tulemused oleksid võrreldavad. Termostaatlahendust kasutatakse etalonina võrdluses teiste lahendustega. Tulemused 24h kestnud simulatsioonist keskkonnal (vt. peatükk 4.3.1) on Tabelis 4. Temperatuurid on tootud Joonisel 11 ning elektritarve Joonisel 12.

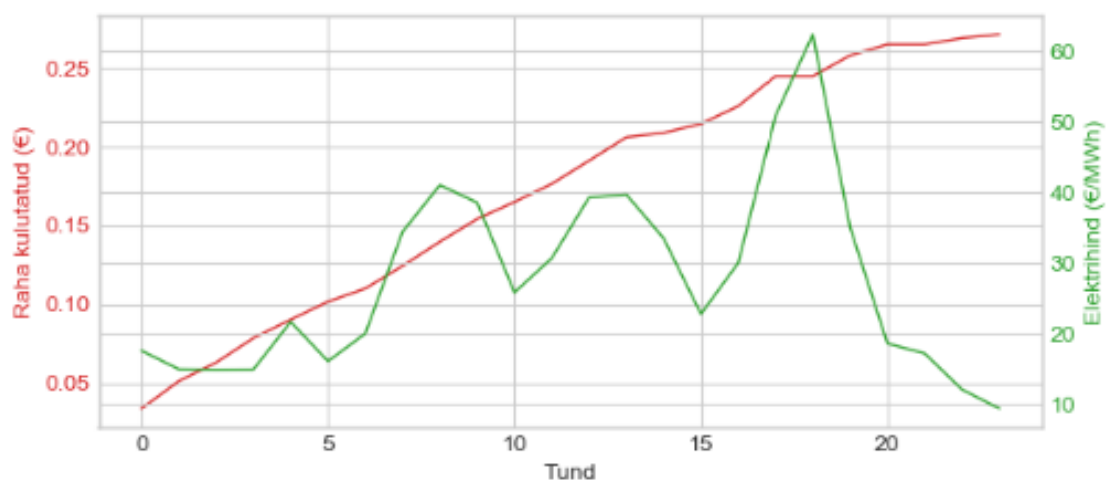
Tabel 4. Termostaatlahenduse tulemus.

Parameetri nimi	Parameetri väärtus
Köetud aeg (min)	281
Keskmine temperatuur (°C)	22.47
Minimaalne temperatuur (°C)	15.00
Maksimaalne temperatuur (°C)	24.00
Temperatuuri standardhälve (°C)	1.121
Kulutatud raha (€)	0.271

⁴<https://www.nordpoolgroup.com/Market-data/Dayahead/Area-Prices/EE/Hourly/?view=table>



Joonis 11. Termostaatlahenduse keskkonna temperatuurid.



Joonis 12. Termostaatlahenduse keskkonna elektritarve.

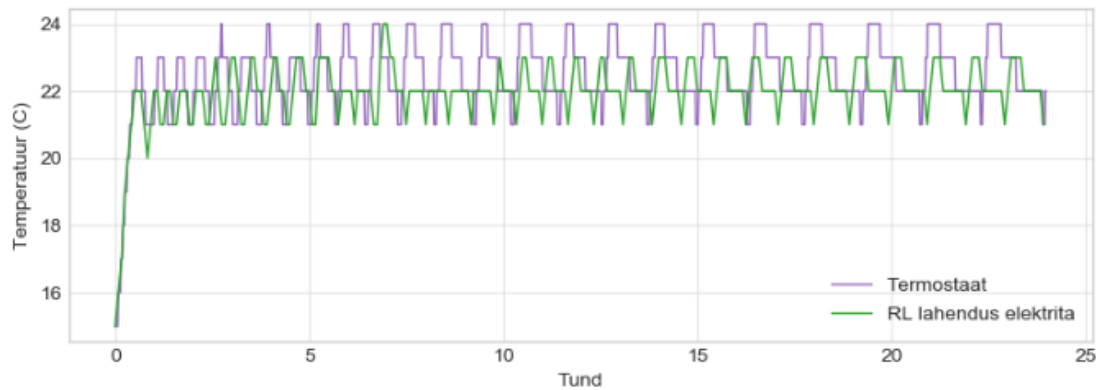
5.2 Temperatuuriga arvestav RL lahendus

Elektrita RL lahendus on stiimulõppe poolt juhitud küttekehaga köetud simulatsiooni tulemus. Selles lahenduses ei kasutanud stiimulõppe algoritm elektrihinda tegevuste arvutamisel. Tulemuseks on tasude põhjal õpitud termostaadiga sarnane käitumine. Simulatsiooni tulemused on toodud Tabelis 5.

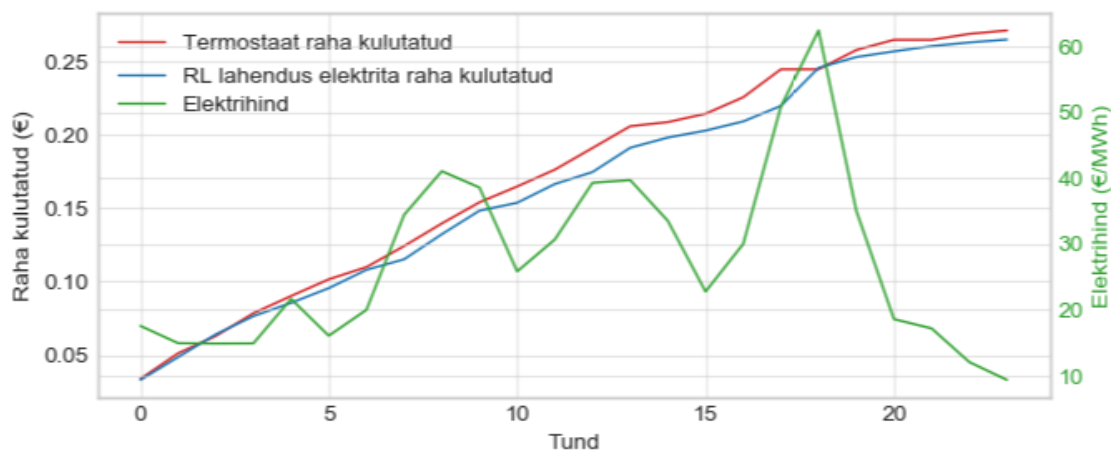
Tabel 5. Termostaatlahendus vs temperatuuriga arvestav RL lahendus.

Parameetri nimi	Termostaat	Temperatuuriga RL lahendus
Köetud aeg (min)	281	265
Keskmine temperatuur (°C)	22.47	21.98
Minimaalne temperatuur (°C)	15.00	15.00
Maksimaalne temperatuur (°C)	24.00	24.00
Temperatuuri standardhälve (°C)	1.121	0.922
Kulutatud raha (€)	0.271	0.265

Võrreldes tulemusi Tabelis 5 selgub, et stiimulõppel põhinev süsteem on tõepoolest õppinud küttekeha juthima sarnaselt termostaadile. Kuna termostaatlahendusse sisseehitatud termostaat tegi mõõtmiseid minutise intervalliga, aga RL lahendus 5 minutise intervalliga, tulevad sisse väikesed erinevused. Põhjus, miks on kasutatud 5 minutilist intervalli, on toodud peatükis 5.4, paragrahvis "Probleem 1 minutise ajasammuga". RL lahendus on suutnud hoida isegi täpsemalt sihttemperatuuri 22°C kui termostaatlahendus. Samuti tuleb välja, et RL lahenduse kütmissaeg oli 16 minutit lühem. Vähem küttes suudeti saavutada rahaline kokkuhoid 0.006€. Võrreldes visuaalselt temperatuure Joonisel 13, on näha, et küttekehad on kogu katse vältel üldjuhul sarnaselt toimunud, kuid RL lahendus on suutnud hoida temperatuuri rangemalt 22° ümber. Samasugust küttekehade käitumist on näha ka võrreldes elektritarvet Joonisel 14.



Joonis 13. Temperatuuriga arvestava RL lahenduse ja termostaatlahenduse keskkonna temperatuurid.



Joonis 14. Temperatuuriga arvestava RL lahenduse ja termostaatlahenduse keskkonna elektritarve.

5.3 Temperatuuri ja elektrihinnaga arvestav RL lahendus

Temperatuuri ja elektrihinnaga RL lahendus on stiimulõppe poolt juhitud küttekehaga köetud simulatsiooni tulemus. Selles lahenduses kasutas stiimulõppe algoritm lisaks temperatuurile ka elektrihinda tegevuste arvutamisel. Tulemuseks on lahendus, mis suudab arvestada elektrihinnaga ning selletõttu ka mitte kütta, kui treenimisel on sellised parameetrid ette antud. Parameetrid, mis treenimisel ette anti, on temperatuuri ja elektrihinna kaalud. Need näitavad, kui suure osa lõplikust tasust, millega arvutatakse Q-väärtus, moodustavad nimetatud parameetrid (vt peatükk 4.4.2). Kui need on võrdsed, siis elektrihind on tasu arvutamisel sama oluline nagu temperatuur. Kui esimene kaal on teisest väiksem, siis esimest parameetrit tasu arvutamisel vaadatakse vähem. Selle tagajärjel kütmise stiil muutub suurema kaaluga parameetri kasuks. Kaalusid muutes on võimalik just kasutaja soovide järgi mudel treenida ning seda ka hiljem rakendada. Nagu temperatuuriga arvestavas lahenduses on ka siin kasutatud 5 minuti pikkust ajasammu.

Kaaludega seotud Q-väärtused arvutati nii nagu on kirjeldatud peatükis 4.4.2. Süsteemi treenimisel kasutati järgnevaid kaalusid:

- temperatuuri kaal: 0.9
- elektrihinna kaal: 0.1, 0.3, 0.5, 0.7, 0.9

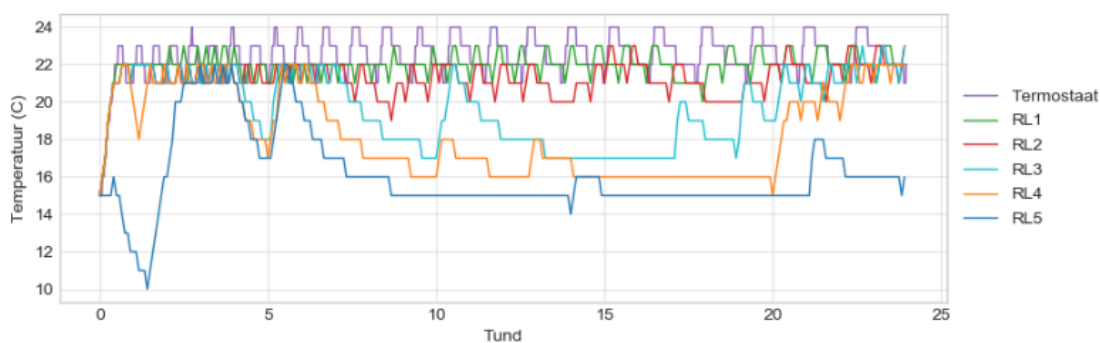
Nagu näha, on ainukeseks muutuvaks parameetriks elektrihinna kaal. Seda põhjusel, et mõlemat parameetrit pole vaja muuta, kui üht kahest muudetakse. Lisades veel parameetreid, peaksid kõik neist olema mingis vahemikus, et saada objektiivseid tulemusi.

Järgnev seksioon toob välja 5 temperatuuri ja elektri hinnaga arvestavat lahendust. Võrreldakse kõikide RL lahenduste tulemusi etalonlahendusega.

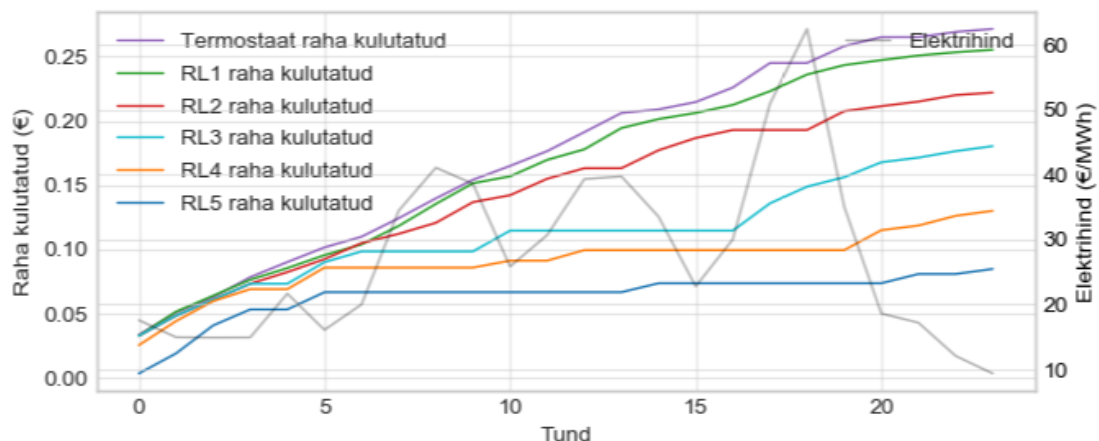
Tabelis 6 on välja toodud kõikide temperatuuri ja elektri hinnaga arvestavate lahenduste tulemused ning lisatud etalonlahenduse tulemused. Lühenditega RL1, RL2, RL3, RL4, RL5 tähistatakse stiimulõppe lahendusi, kus elektri hinna kaalud on vastavalt 0.1, 0.3, 0.5, 0.7 ja 0.9. Lisaks andmetele Tabelis 6 on toodud ka joonised 24h kütmise andmetega Joonistel 15 ja 16.

Tabel 6. Termostaatlahendus vs temperatuuri ja elektri hinnaga arvestavad RL lahendused.

Parameetri nimi	Termostaat	RL1	RL2	RL3	RL4	RL5
Köetud aeg (min)	281	260	250	220	190	130
Keskmine temperatuur (°C)	22.47	21.94	21.28	19.64	18.31	16.10
Minimaalne temperatuur (°C)	15.00	15.00	15.00	15.00	15.00	10.00
Maksimaalne temperatuur (°C)	24.00	23.00	23.00	23.00	22.00	22
Temperatuuri standardhälve (°C)	1.121	0.874	0.965	1.919	2.286	2.169
Kulutatud raha (€)	0.271	0.255	0.222	0.180	0.130	0.085



Joonis 15. Temperatuuri ja elektri hinnaga arvestavate RL lahenduste ja termostaatlahenduse keskkonna temperatuurid.



Joonis 16. Temperatuuri ja elektrihinnaga arvestavate RL lahenduste ja termostaatlahenduse keskkonna elektritarve.

Jooniseid 15 ja 16 vaadeldes tuleb välja, et termostaatlahendusele kõige sarnasem on lahendus RL1. RL1 lahenduse temperatuurikõver ning elektrihinnakõver on kõige sarnasem etalonlahendusele. Siiski joonistub välja, et igal tunnil on RL1 lahendus kütnud veidi vähem kui etalonlahendus ning seda ilemstab ka tõsiasi, et kütmisajast hoiti kokku 21 minutit. See on 7.5% langus, millega hoiti kokku 0.016€ 24h jooksul, kuid ei kaotatud keskmises temperatuuris. Suurem osa kokkuhoiust tuli hetkedest, millal elektrihind tegi järsu tõusu (6.-9. tund, 10.-13. tund ja 16.-18.tund). Tabelist 6 on näha, et RL1 lahendus suutis hoida keskmise temperatuuri kõigest 0.06°C madalamana kui sihttemperatuur, kuid etalonlahendus hoidis seda 0.47°C võrra kõrgemal. Samuti on standardhälve väiksem kui etalonlahendusel, mis näitab, et keskmise temperatuuri ümber on olnud väiksem kõikumine. Lisaks sellele hoidis RL lahendus ka maksimaalse temperatuuri 1° võrra madalamal kui etalonlahendus, et hoiduda liigsest kütmisest.

Vaadeldes lahenduse RL2 jooniseid, on samuti näha, et temperatuurikõver on 22°C ümber, kuid kattumist etalonlahendusega on juba vähem. Keskmist temperatuuri suutis lahendus hoida 21.28°C juures, mis on kõigest 0.25°C võrra suurem kui etalonlahenduse kaugus keskmisest temperatuurist. Seega, temperatuuri suudeti hoida peaaegu sama hästi kui etalonlahenduses, kuid köeti 31 minutit vähem. See on 11% kokkuhoid kütmisajas ning 0.049€ kokkuhoid küttekulult. Lahenduses RL2 on veel selgemalt näha, kuidas elektrihinna tõusud mõjutavad kütmist. Näiteks 18. tunni juures on kütmine lõppenud, sest elektrihind tegi väga suure ja järsu tõusu ning jõudis maksimumini (63€/MWh).

RL3 lahendus erineb selgelt etalonlahendusest ning raha säästmise eesmärgil on keskmine temperatuur suuresti langenud. Keskmist on suudetud hoida kõiges 19.64°C juures, mis on peaaegu 2.5°C võrra madalamal sihttemperatuurist. Sellise keskmise

temperatuuriga suudeti saavutada märgatav kokkuhoid rahas. Joonisel 16 on näha, et kütmist tahetakse peatada koheselt kui elektri hind on jõudnud 30€/MWh juurde. Siiski, et hoida temperatuuri piisavalt kõrgel, on vahepeal kütmine taastunud enne elektri hinna langust allapoole 30€. Jooniseid 16 ja 15 vaadates on näha, et nii 6.-9. ja 10.-16. tunnil ei ole köetud, sest elektri hind on kõrge. Selle tagajärjel on sisetemperatuur langenud 17°C juurde. Hoidmaks ära veel suuremat langust 17. tunni juures, hakati kütma hoolimata sellest, et elektri hind tegi väga järsu hüppe. Siiski ei köetud ruumi sihttemperatuurini, vaid kõigest kuni 20°C. Sellise madala keskmise temperatuuriga suudeti saavutada 22% kokkuhoid kütmise ajas, mis teeb välja 0.091€ suuruse kokkuhoiu.

RL4 lahendus on üldjoontes sarnane RL3 lahendusega. Keskmise temperatuur langes küll 18.31°C juurde, kuid temperatuurikõver ning kulutatud raha kõver on sarnased. Erinevus tuleb välja veel suuremas tundlikkuses elektri hinna suhtes. Kütmisest proovitakse hoiduda, kui elektri hind on kõigest 25€/MWh. Selle tagajärjel langeb temperatuur 16°C juurde. Hoidumaks temperatuuri veel madalamale langemisest köetakse vahepeal kõrgema elektri hinna ajal, kuid kõigest nii kaua kuni saavutatakse sisetemperatuur 18°C. Erinevalt RL3 lahendusest, kus 16. tunnil hakati hoolimata väga kiiresti tõusvast elektri hinnast kütma, hoiduti sellest RL4 lahenduses. Väga kõrge hinna ajal ei köetud, mille tagajärjel langes temperatuur 15°C juurde ning uuesti hakati kütma alles elektri hinnaga 30€/MWh.

Lahendus RL5 erineb täielikult kõigest eelnevatest. Soodsa elektri hinna ajal simulatsiooniperioodi alguses otsustati mitte kütta. Selle tagajärjel langes temperatuur 10°C juurde. Põhjus sellise käitumise taga jääb töö autorile teadmatuks. Peale langust köetakse endiselt soodsa elektri hinnaga alates 2.tunnist sihttemperatuurini. Alates 6. tunnist langeb sisetemperatuur taas. Sel ajal on elektri hind ületanud 20€/MWh kohta, mis tundub olevat ka kriitiline elektri hinna piir selles lahenduses. Kütmisest proovitakse täielikult hoiduda, mille tagajärjel on temperatuur 15°C ümbruses kuni 21.tunnini. Peale seda langeb elektri hind taas alla 20€/MWh, tänu millele hakatakse uuesti kütma, kuid saavutatakse kõigest 18°C ning kütmine lõpetatakse taas. Võit kütmisajas võrreldes etalonlahendusega on 54%, millega hoiti kokku peaaegu 0.02€. Temperatuuri hoidmiseks selline lahendus ei sobi, sest keskmine temperatuur oli kõigest 16.10°C ning kõikus suuresti selle ümber.

5.4 Arutelu

Analüüsi käigus leiti, et stiimulõppel põhinev lahendus on võrreldav termostaatlahendusega. Lisades temperatuurile ka elektri hinna jälgimise, on võimalik saada paremaid tulemusi kui termostaatlahendus on suuteline saama. Selle tagajärjel on võimalik saavutada ka rahaline võit, kaotamata soojuses, nagu lahenduses RL1 selgus. Muutes süsteemi treenides parameetrite kaalusid, on võimalik luua just selline tasakaal elektri hinna ja temperatuuri vahel, et sääst oleks maksimaalne ning temperatuur endiselt sobiv. Näiteks RL lahendus 1 saavutas 6% kokkuhoiu kulutatud rahas, kaotamata soojuses ning hoides sihttemperatuuri paremini võrreldes etalonlahendusega. Kui termostaatlahendusel

oli keskmine temperatuur 0.5°C võrra kõrgem kui sihttemperatuur, siis RL lahendus 1 hoidis keskmist temperatuuri kõigest 0.06°C sihist madalamal. RL lahendus 2 saavutas rahaliseks kokkuhoiuks 18%, langetades keskmise temperatuuri 21.28°C juurde, mis on kõigest 0.7°C võrra sihttempeartuurist madalam. Suurendades elektriinna kaalu, on võimalik saavutada kuni 70% võit rahas. Selle tagajärjel hoitakse keskmist temperatuuri kõigest 16° ümber, langedes isegi kõigest 10°C juurde.

Uurimuse praktilises osas loodi stiimulõppel põhinev süsteem. Kuigi üldjoontes on see hästi koostatud ning treenitud, leidub selles ka problemaatilisemaid kohti. Sellised kohad on järgnevad:

- elektriinnad;
- süsteemi liigne seotus Energy2D tarkvaraga;
- probleem 1 minutise ajasammuga.

Elektriinnad

Treenimisel kasutati alati samu 24 tunni elektrihindasid. Selle tagajärjel on Q-tabel ületreenitud ning annab väga häid vastuseid vaid treenitud elektriandmetega ja väikse kõikumisruumiga. Kõikumisruumi põhjus on järgnev. Elektriinnad Q-tabelis on ümardatud täisarvuni seega kui elektrihind tunnis oli 25.23 ühikut, siis Q-tabelis on sellega seotud kirje hinnaga 25 ühikut. Seega, kui elektrihind oleks 25.49 ühikut, siis Q-tabeli võti on endiselt sama ning sellest tuleneb ka kõikumisruum. Selle probleemi oleks saanud ära hoida, kasutades suuremat andmehulka elektriandmetena. Selleks, et selline probleem mudelit kasutades, mitte treenides, lahendada, tuleks süsteem ümber teha. Praegu ei uuendata Q-tabelit testimise ehk kasutamise käigus. Kui tuleb ette Q-tabeli võti, mida ei ole nähtud treenides, tehakse valik esimese tegevuse kasuks. Peale tegevuse tegemist peaks arvutama Q-valemi ja kaalude põhjal uue Q-väärtuse ning selle tabelisse sisestama, et järgmine kord oskaks süsteem teha kaalutud otsuse.

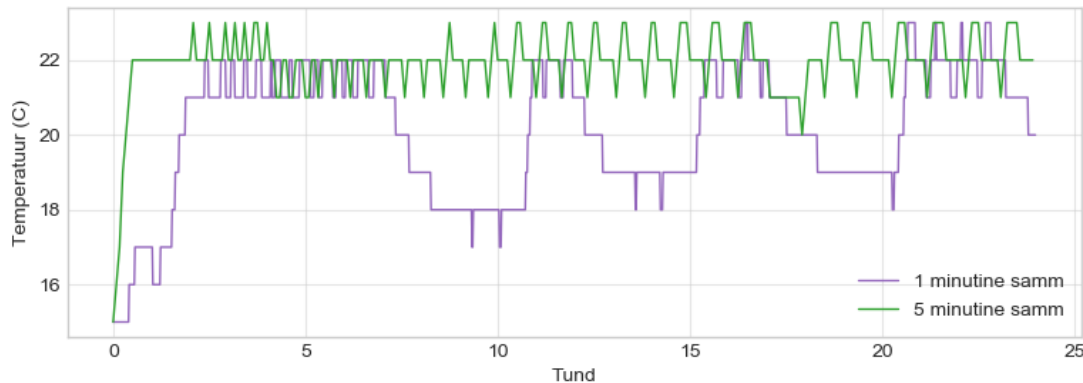
Süsteemi liigne seotus Energy2D tarkvaraga

Kõik simulatsioonid on läbi viidud Energy2D keskkonnas. See on samuti kaasa toonud mingil määral ületreenimise. Vaadates keskkonda peatükis 4.3.1 on selge, et sellist keskkonda tegelikult ei eksisteeri. Tühi ruum, kus pole mitte ühtegi küttekeha ning on kahemõõtmeline, ei ole päriselt järele tehtav. Päriskeskkonnas on veel küttekehi nagu näiteks inimesed ning elektripirnid. Samuti on takistusi, mis muudavad ning segavad soojuste levimist. Energy2D kasutati simulatsioonides selle lihtsuse ja hea visualisatsiooni tõttu. Selleks, et töös loodud lahendust kasutada päriselu keskkonnas, on oluline jälgida, et loodav keskkond oleks lihtne. Tuleb hoiduda potentsiaalsetest küttekehadest (näiteks hüügpirnid ja inimesed) peale tervet põrandat kütva põrandakütte. See ajaks süsteemi segadusse kuna seda on õpetatud käituma kõigest ühe küttekehaga. Samuti pole

süsteem näinud sisetemperatuure, mis on väiksemad kui 10°C ja välistemperatuure, mis pole vahemikus 8°C kuni 20°C. Nähes nimetatud temperatuuridest erinevaid suuruseid, käitaks süsteem suvaliselt ning ei suudaks temperatuuri saavutamiseks õigeid tegevusi teha.

Probleem 1 minutise ajasammuga

Stiimulõppel läbi viidud treeningut ning testimised on kõik tehtud 5 minutise ajasammuga. See tähendab, et küttekeha on võimalike lülitada ainult 1 kord minutis. Etalonlahenduses oli kasutus 1 minutine ajasamm ning see ei tekitanud probleeme. Proovides stiimulõppe lahendustesse sisse tuua 1 minutist ajasammu, tekkisid anomaaliad. Joonisel 17 on näha, et 1 minutise sammuga süsteem ei suutnud korrektselt kütta ruumi, mida 5 minutise sammuga süsteem suutis.



Joonis 17. 1 ja 5 minutise ajasammu võrdlus.

Probleemi põhjust ei oska töö autor tuua. Siiski pakutakse välja hüpotees. Võib-olla ei ole Q-õppe süsteem 1 minutise treenimise ajaga saanud keskkonnalt piisavalt tagasisidet, et arvutada korrektne Q-väärtus. Selle tagajärjel on kütmine toimunud suvaliselt ning treenimisest pole kasu olnud. Sellele probleemi välitmiseks kasutati andmeanalüüsis 5 minutise sammuga treenitud stiimulõppe süsteeme.

6 Kokkuvõte

Käesoleva töö käigus loodi Q-õppel põhinev küttesüsteem, mis suudab kütmist automatiseerida ning saada kokkuhoid rahas. Loodud süsteemis kasutati masinõppe treeninguks vajaliku keskkonna simuleerimiseks Energy2D tarkvara. Seal on võimalik lühikese aja jooksul tuhandeid kordi identsetes tingimustes treenida. Süsteem loodi, et leida, kas ja kuidas on masinõppega võimalik saavutada paremaid tulemusi kui termostaadiga juhtitud küttekeha on suuteline. Selle hindamiseks koostati ja võrreldi erinevaid lahendusi termostaatlahendusega, et leida, kuidas masinõppel põhinevad lahendused aitavad kütmist automatiseerida. Loodud süsteem suudab automatiseerida kütmist simulatsioonides. Simultasioonis olevat keskkonda ning Q-õppe süsteemi edasi arendades on võimalik ka päriselt süsteem käima panna ning saavutada kasu ka päriselus.

Tulemusi analüüsid leiti, et elektriinanga arvestav kütmislahendus suudab saavutada rahalise kokkuhoiu. Kokkuhoiuks saadi vähemalt 5%, mille saab otseselt ümber arvutada kokkuhoiuks energias. Saadud tulemus ühtib ka töös loetletud kirjanduses välja toodud tulemustega, et kokkuhoid kodu kütmist automatiseerides on 5%-15%. Samuti leiti, et elektriinnata lahendus suudab õppida kütmise ära sama hästi kui termostaatlahendus ning saavutada peaagu identseid tulemusi. Lisaks leiti, et rahalise kokkuhoiu suurendamiseks on võimalik süsteemi õpetada nii, et see pööraks vähem tähelepanu sihttemperatuurile ning rohkem elektrihinnale. Sellised lahendused langetasid keskmise temperatuuri juba alla 17°C, kuid kokkuhoid rahas oli kuni 70%.

Lahenduse edasiarenduseks ning paremaks muutmiseks on mitmeid võimalusi. Üks võimalustest on keerulisemate keskkondade loomine Energy2D tarkvaras. Selle tagajärjel on võimalik läbi mängida keerulised keskkonnad, mis on sarnasemad päris keskkondadega. Samuti tuleks lisada suurem andmehulk elektriandemtega, et süsteem ei oleks ületreenitud väikese andmehulga tõttu. Teiseks võimaluseks on sügav stiimulõppe (DRL ehk *Deep Reinforcement Learning*). Sellega lisanduks stiimulõppele veel tehisnärvivõrgud, mis suudavad süsteemi veel paremaks muuta. Lisatud tehisnärvivõrgud suudaksid ära õppida inimese käitumise keskkonnas ning selle järgi teha otsuseid küttekehade juhtimiseks.

Viidatud kirjandus

- [1] Feng S., Setoodeh P. ja Haykin S. “Smart Home: Cognitive Interactive People-Centric Internet of Things”. *IEEE Communications Magazine* 55.2 (veebruar 2017), lk. 34–39.
- [2] Chifor B.-S. *et al.* “Security-Oriented Framework for Internet of Things Smart-Home Applications”. Teoses: *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*. Mai 2019, lk. 146–153.
- [3] *Energiatõhusus | Teabelehed Euroopa Liidu kohta | Euroopa Parlament*. <https://www.europarl.europa.eu/factsheets/et/sheet/69/energiatohusus> (vaadatud 08.03.2020).
- [4] Moser K., Harder J. ja Koo S. G. M. “Internet of Things in Home Automation and Energy Efficient Smart Home Technologies”. Teoses: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, oktoober 2014, lk. 1260–1265.
- [5] *Energy2D - Interactive Heat Transfer Simulations for Everyone*. <https://energy.concord.org/energy2d/> (vaadatud 20.04.2020).
- [6] *Machine Learning Definition | DeepAI*. <https://deepai.org/machine-learning-glossary-and-terms/machine-learning> (vaadatud 05.04.2020).
- [7] Bell J. *Machine Learning: Hands-On for Developers and Technical Professionals*. USA: Wiley, 2014. 408 lk. kokku.
- [8] Singh A., Thakur N. ja Sharma A. “A Review of Supervised Machine Learning Algorithms”. Teoses: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. Märts 2016, lk. 1310–1315.
- [9] Sutton R.S. ja Barto A.G. *Reinforcement Learning: An Introduction*. Second edition. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018. 526 lk. kokku.
- [10] *An Introduction to Q-Learning: Reinforcement Learning*. <https://www.freecodecamp.org/news/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc/> (vaadatud 20.04.2020).
- [11] Zychlinski S. *The Complete Reinforcement Learning Dictionary*. <https://towardsdatascience.com/the-complete-reinforcement-learning-dictionary-e16230b7d24e> (vaadatud 01.05.2020).
- [12] *Demystifying Deep Reinforcement Learning | Computational Neuroscience Lab*. <https://neuro.cs.ut.ee/demystifying-deep-reinforcement-learning/> (vaadatud 05.04.2020).

- [13] *MathWorks - Makers of MATLAB and Simulink*. <https://se.mathworks.com/> (vaadatud 29.04.2020).
- [14] *Simulation Heat Transfer - Confirmation*. <https://www.solidworks.com/sw/simulation-heat-transfer-confirmation.htm> (vaadatud 29.04.2020).
- [15] *Interactive Heat Transfer Simulations for Everyone: The Physics Teacher: Vol 50, No 4*. <https://aapt.scitation.org/doi/10.1119/1.3694080> (vaadatud 25.03.2020).
- [16] *EnergyPlus | EnergyPlus*. <https://energyplus.net/> (vaadatud 20.04.2020).
- [17] *3D Design Software | 3D Modeling on the Web*. <https://www.sketchup.com/page/homepage> (vaadatud 20.04.2020).
- [18] Cook D.J. *et al.* "Learning to Control a Smart Home Environment". Teoses: *Innovative Applications of Artificial Intelligence*. 2003.
- [19] *DeepMind AI Reduces Google Data Centre Cooling Bill by 40%*. </blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40> (vaadatud 24.03.2020).
- [20] Wei T., Wang Y. ja Zhu Q. "Deep Reinforcement Learning for Building HVAC Control". Teoses: *Proceedings of the 54th Annual Design Automation Conference 2017 on - DAC '17*. ACM Press, 2017, lk. 1–6.
- [21] Xie C. *Charxie/Energy2d*. <https://github.com/charxie/energy2d> (vaadatud 20.04.2020).

Lisad

I. Q-õppe koodirepositoorium

Töö käigus kirjutatud Q-õppe kood on olemas autori GitHubi repositooriumis, mille URL on <https://github.com/erikmukk/q-learning-gradle-project> .

II. Muudetud Energy2D koodirepositoorium

Töö käigus muudetud Energy2D aluskood on olemas autori GitHubi repositooriumis, mille URL on <https://github.com/erikmukk/energy2d/tree/headless2>.

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Erik Mukk**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
Stiimulõppel põhinev nutika kodu küttelahendus Energy2D simulatsiooni-tarkvaras,
mille juhendaja on Jakob Mass,
reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Erik Mukk
08.05.2020