User Guide Template

May 2020

Version 1.01

The Australian National University

GROUP MEMBER: MENGZHOU WANG, TIANHAO YU, ZI'ANG LIU, BIN DONG, YUNJIE JIN, WEICONG ZHANG

Table of Contents

1 Installation and setup

1.1 Environment requirement

1.1.1 Test Environment

1. Python 3.7.0
2. TESpy 0.2.2
3. Git 2.21.0

1.2 Prerequisites

Key parts of this project require the following scientific software packages

1.2.1 Packages used

All the following packages can be installed by terminal

Command: “pip install …”

1. NumPy
2. Matplotlib
3. Pandas
4. Warnings
5. Math
6. Prettytable
7. Time

1.3 Installation

Type “git clone https://github.com/Tianhao-Y/Automatic-Control-System-for-Heat-Pump.git” in terminal

2 Sub-system usage

2.1 Room Model

2.1.1 Variables introduction

:param Printer: Print information or not --> boolean

:param room_temp: room temperature in Degrees Celsius --> float

:param environment_temperature: environment temperature in Degrees Celsius --> float

:param power: input heat pump power W --> float

:param method_heatpump: whether it is heater or not --> string "Heater" or "Cooler"

:return: the Update Temperature

2.1.2 Parameters introduction

r"""

:param density_a: air density in kg/m^3

:param capacity_a: the heat capacity of the air J/（kg·℃）

:param area: the size of the room in m^2

:param height: the height of the room in m

:param wall: The heat loss of the walls W/m^2*℃

:param window: The heat loss of the windows W/m^2*℃

:param ventilation: coefficient of ventilation W/m^3*℃

:param wallArea: the area of walls and windows should be a list --> [area_walls, area_windows] m^2

"""

2.1.3 How to change and print parameters

room = Room(room_temp, environment_temperature, power)

room.setRoomInfo()

room.getRoomInfo()

2.1.4 Usage

The output is the temperature after certain amount of time.

1. **Method 1:**

```
temp = roomTemp(room_temp, environment_temperature, power, method_heatpump, Printer)
```

2. **Method 2:**

```
room = Room(room_temp, environment_temperature, power)

temp = room.heatUp(Printer, method_heatpump)
```

2.2 Heat Pump Model

2.2.1 Parameters and Variables Declaration

pr1	Pressure ratio outlet 1	0.99
pr2	Pressure ratio outlet 2	0.98
amb_p.T	Groud source Temperature	12
amb_p.p	Groud source pressure	2
amb_out.T	Coolent Temperature	9
cp1.h0	flow entropy	1700
cd.ttd_u	upper terminal temperature difference of condersor	15
ev.ttd_l	lower terminal temperature difference of evaperater	5
su.ttd_l	lower terminal temperature difference of the heat exchanger	2
cb_dhp.T	input air flow Temperature	20
cd_cons.T	output air flow Temperature	Variable
eta_s	motor efficiency of the pumps	Variable
cons.Q	Output heat energy	Variable
design	Geometry independent friction coefficient heating loop	zeta2 :ζ/ 1 m4.

2.2.2 How to change Variables

```
Hp = HeatPump(q, eff, Temp)

r"""

    :param Temp: Temperature at the air outlet --> float

    :param q: q output

    :param eff: efficient of each part in pump

    """
```

2.2.3 Variables and Outputs declaration

```
r"""

  :param state: the gears of the heat pump --> int 0 - 10
```

```
    :param last_q: the heat output of last seconds --> float

    :param method: type of String, should be 'heater' or 'cooler' to indicate the type of heat
pump

    :return: current heat output of the heat pump --> float

                    Power Consumption in total --> float

                    The efficiency of the heat pump COP --> float

                    Power Consumption for each part --> list

                    The efficiency of the Motor --> float

                    Temperature at the air outlet --> float

                    The heat output in current seconds --> float

    """
```

2.2.4 Usage

1. Method 1:

```
Q, P_total, COP, P, eff, T, current_q = operation (state, last_q, method)
```

1. Method 2:

```
hp = HeatPump(Q_out, eff, T)

P, P_total, COP = hp.caculation()
```

2.3 Control system

2.3.1 Parameters and Variables Declaration

Parameters & Variables	**Declaration**	**Units**
date	Date	Day
room_t	Room temperature	Celsius
deamand	Demand temperature (User demand)	Celsius
price	Price forecast data (input .csv files)	dollars per hundred kilowatts

temp	Weather forecast data (input .csv files)	Celsius

2.3.2 How to change Variables

control= control(demand, date, room_t)

r"""

:param time: total minute which need to analsying--> int should be delete 120

:param date: choose the data to analysing --> int

:param method_heatpump: type of String, should be 'heater' or 'cooler' to indicate the type of heat pump

:param demand: user demand data --> list

:param price_list: daily price data (name) --> 'name.csv' type

"""

2.3.3 Variables and Outputs declaration

r"""

:param time: total minute which need to analsying--> int should be delete 120

:param date: choose the data to analysing --> int

:param method_heatpump: type of String, should be 'heater' or 'cooler' to indicate the type of heat pump

:param demand: user demand data --> list

:param price_list: daily price data (name) --> 'name.csv' type

:param temp: daily weather temperature data (name) --> 'name.csv' type

:return: costlist --> list

p_list --> list

T_room --> list

cost --> float

newP --> float

2.3.4 Usage

Method 1:

```
Costlist, p_list, T_room, cost, newP = controloptimal(time, date, Method, demand, ‘Price data name.csv’, 'weather data name.csv')
```