This file explains the blocks that go into the House Heat Simulink model and how RL blocks are configured into it.

SYSTEM DESIGN

We have designed the thermal control system which consists of the following components, namely, setpoint, Heater, Cost Calculator, House, RL system.

1 Setpoint:

It is a constant block. It specifies the temperature that must be maintained indoors. It is 75 degrees Fahrenheit by default.

2 Cost Controller:

It is a Gain block. "Cost Calculator" integrates the heat flow over time and multiplies it by the energy cost. The cost of heating is plotted in the "Plot Results" scope.

3 Heater:

"Heater" is a subsystem that has a constant airflow rate, "Mdot". The RL system actions (0,1) signal turns the heater on or off. When the heater is on, it blows hot air at temperature THeater (50 degrees Celsius = 122 degrees Fahrenheit by default) at a constant flow rate of Mdot (1kg/sec = 3600kg/hr by default). The heat flow into the room is expressed by the equation given below:

$$\frac{dQ}{dt} = (T_{heater} - T_{room}) \cdot Mdot \cdot c$$

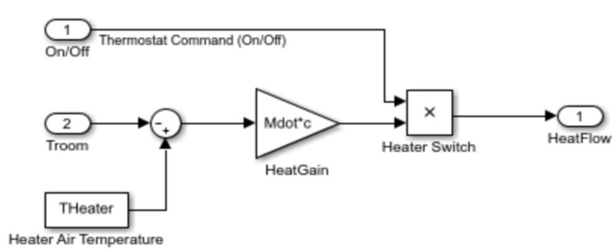
$\frac{dQ}{dt} =$ heat flow from the heater into the room

$c =$ heat capacity of air at constant pressure

$Mdot =$ air mass flow rate through heater (kg/hr)

$T_{heater} =$ temperature of hot air from heater

$T_{room} =$ current room air temperature



Heater Subsystem

4 House:

It is a subsystem that calculates room temperature variations. It takes into consideration the heat flow from the heater and heat losses to the environment. Heat losses and the temperature-time derivative are expressed by:

$$\left(\frac{dQ}{dt}\right)_{losses} = \frac{T_{room} - T_{out}}{R_{eq}}$$

$$\frac{dT_{room}}{dt} = \frac{1}{M_{air} \cdot c} \cdot \left(\frac{dQ_{heater}}{dt} - \frac{dQ_{losses}}{dt}\right)$$

$M_{air} =$ mass of air inside the house

$R_{eq} =$ equivalent thermal resistance of the house

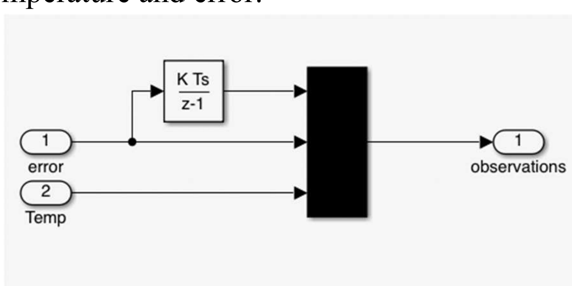
5 RL subsystem

5.1 RL agent:

The RL agent block is an inbuilt MATLAB Simulink block used to train and stimulate the reinforcement learning agent where we associate the block with an agent stored in the workspace so that it receives the observation and computes the reward so as to give out the best action according to the algorithm used.

5.2 Observation block:

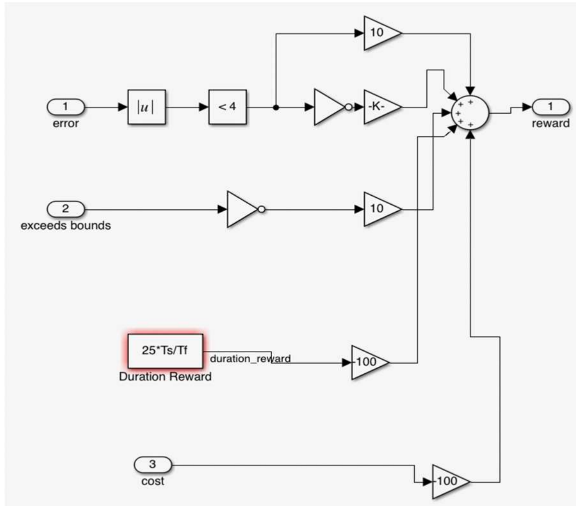
Observes the predictive temperature and error.



5.3 Reward block:

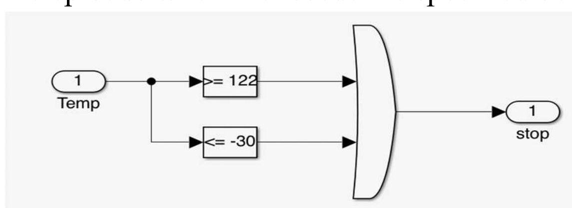
Calculates the reward based on certain conditions namely:

- Minimize the cost function
- Minimize the error
- Achieve temperature control in less time.
- Should not exceed the bounds given.



5.4 Stop Simulation block:

This block is used to stop the episode when it exceeds the specified bounds.



We have implemented all the above-mentioned algorithms using the given set of blocks by tuning them accordingly with their respective hyperparameter tuning.

5.5 About the Agent:

The Deep Deterministic Policy Gradient algorithm is a model-free, online, off-policy reinforcement learning method. A DDPG agent is an actor-critic agent that computes an optimal policy that maximizes the long-term reward.

During the training, a DDPG agent:

- Updates the actor and critic properties at each time step during learning.
- Stores past experience using a circular experience buffer. The agent updates the actor and critic using a mini batch of experiences randomly sampled from the buffer.
- Perturbs the action chosen by the policy using a stochastic noise model at each training step.