Cubic room heated by a fan-coil

MATLAB / Octave files:
t06CubeHeat.m;
fTC2SS.m
fTCAssAll
fSolRadTiltSurf.m
fReadWeather.m

Objectives:

- Implement non-linear controllers.
- Use models with variable coefficients.

1 Defining the problem of control

Consider the same cubic room as in tutorial 3 to 5 but heated by a fan-coil. This time the power is zero if the indoor temperature is larger than the set-point. Then, construct a model for heating and cooling with hysteresis.

2 Implementation

Two models are needed:

- one for controller
- another one for free-running, i.e. the gain of the controller practically zero.

The model with controller is identical to those from tutorial 5.

Consider a proportional controller used only for heating. In this case, the cubic room is heated if the indoor temperature is lower than its set point; otherwise, the heat flow is zero, i.e. the building is in free-running. Uncomment line 48 if you want to consider negligible heat capacities for the air and the glass.

```
46   Gg = lamg/wg*Sg; Cg = Sg*wg*rhogcg; %glass
47   Ca = Va*rhoa*ca;
48   % Ca =0; Cg = 0;  % if air and glass are neglected
```

The model of the free-running room is similar to the controlled room, but the gain of the controller is practically zero ($K = 10^{-4}$ in line 115).

```
113  [Ac,Bc,Cc,Dc] = fTC2SS(A,G,b,C,f,y);    % model with controller
114
115  G4 = diag([Gv 1e-1]);   % no controller: practically Kp = 0
116  TCd{4} = {A4,G4,b4,C4,f4,y4};
117  [TCa, Idx] = fTCAssAll(TCd, AssX);
118  A = TCa{1}; G = TCa{2}; b = TCa{3}; C = TCa{4}; f = TCa{5}; y = TCa{6};
119  [Af,Bf,Cf,Df] = fTC2SS(A,G,b,C,f,y);    % model free-running
```

In the integration loop, select the model with controller or with free-running based on the comparison of indoor temperature (the output of the model) and its set-point.

```
160   for k = 1:n-1
161       if y(k) < TintSP(k)
162           th(:,k+1) = (eye(nth) + dt*Ac)*th(:,k) + dt*Bc*u(:,k);
163           y(:,k+1) = Cc*th(:,k+1) + Dc*u(:,k+1);
164           qHVAC(:,k+1) = Kp*(TintSP(k+1) - y(k+1));
165       else
166           th(:,k+1) = (eye(nth) + dt*Af)*th(:,k) + dt*Bf*u(:,k);
167           y(:,k+1) = Cf*th(:,k+1) + Df*u(:,k+1);
168           qHVAC(:,k+1) = 1e-1*(TintSP(k+1) - y(k+1));
169       end
170   end
```

3 Numerical experiments and discussion

3.1 Study different seasons

Change the season and comment the results.
Winter:

```
126   % Load weather data
127   fileName = 'FRA_Lyon.csv';
128   from = 1*30*24 + 25*24;   % start time: from 24 Jan.
129   period = 10*24; % simulation period: for 10 days
```

Summer:

```
128   from = 1*30*24 + 25*24;   % start time: from 24 Jan.
```

Note the overheating.

3.2 Study the influence of the initial conditions

Note the values of the indoor temperature and the heating power at the beginning of the simulation period.
Change the initial conditions from 0 °C

```
156   % th = 20*ones(size(Ac,2),n);
157   th = zeros(size(Ac,2),n);
```

to 20 °C

```
156   th = 20*ones(size(Ac,2),n);
157   % th = zeros(size(Ac,2),n);
```

Comment the results for indoor temperature and heat flow for heating.

3.3 Implement a heating and cooling with hysteresis

Consider that if the indoor temperature is larger than 24°C, cooling is needed.
Implement the cooling strategy.
Hint: in the `for` loop from line 160, check if the indoor temperature is larger than 24 °C.
If yes, use the model with controller (see `t06CubeHeatCool.m`):

```
161       if y(k) < TintSP(k) || y(k) > 4 + TintSP(k)
```

Note the influence of the controller amplification on the stability ; use different values for K_p: 10^2; 10^3; 10^4 in line 7 :

```
Kp = 1e4;                   % P-controller gain: large for precision
```

Compare the maximum and minimum heat flow rates for different values of K_p. Discuss the consequences on the size of the HVAC system.

3.4 Implement cooling by window shading and air flow rate

Overheating may be avoided by reducing the solar gains and/or by increasing the ventilation rate. Implement these strategies.