Comfort-Driven Thermostats and Hydronic Controls

Perry Kundert

2026-11-10 00:00:00

Abstract

Do Your Thermostats Measure What Your Customers Feel?

People don't feel the air temperature. They feel the heat they lose – to the air, and by radiation to every window, wall and floor around them. A room held at 21°C beside a wall of winter glass feels chilly; the same room over a warm radiant floor feels warm. Every customer complaint of "it's cold in here, but the thermostat says 21°C" is a radiant temperature problem.

Dominion R&D has controlled a multi-zone hydronic radiant home by *comfort* – Fanger's Predicted Mean Vote (ISO 7730) – for over a decade, estimating the temperature of every surface from just a few sensors. The core is now the Open Source Python package pyhydronic. We help thermostat, hydronic control, heat pump and building automation vendors bring comfort control into their products, under a commercial license if desired. (PDF, Text)

Contents

1 People Feel Surfaces, Not Just Air **1**

2 Radiant Temperature From The Sensors You Already Have **2**

3 Comfort Control **2**

4 Who Benefits **3**

5 What We Offer **3**

6 Licensing **3**

6.1 GNU GPLv3 3

6.2 Commercial License 4

7 How To Get Started **4**

1 People Feel Surfaces, Not Just Air

Fanger's comfort equation (the basis of ISO 7730 and ASHRAE 55) predicts how a large group of people will rate a space, from "cold" (-3) through "neutral" (0) to "hot" (+3),

and the percentage who will be dissatisfied. Its inputs are the air temperature, the *mean radiant temperature* of the surroundings, humidity, air speed, clothing and activity.

For a person in winter indoor clothing, sitting or standing, at 40% relative humidity:

Air	Surfaces (mean radiant)	Predicted Mean Vote	Dissatisfied
21°C	21°C (interior room)	-0.24 (neutral)	6%
21°C	17°C (large windows)	-0.64 (slightly cool)	14%
21°C	25°C (warm radiant floor)	+0.16 (neutral)	6%

To be comfortable beside that cold glass needs 26.9°C air; over the warm floor, 19.8°C is enough. A 21°C air setpoint is comfortable only while the surfaces average between 18.5°C and 28.3°C. An air-temperature thermostat cannot know which room it is in.

2 Radiant Temperature From The Sensors You Already Have

Nobody installs a globe thermometer in every room. Instead, a simple thermal model of the building – rooms, walls, windows, ceilings and floors, each with an area and an R value – estimates every surface temperature from a few measurements (outdoor, room air, slab or supply water), and hence each room's mean radiant temperature. These "synthetic sensors" update every minute, and the computation is trivial: two days of minute-by-minute simulation of a small house takes well under a second, in Python.

```
from hydronic import space, portal, environment, fanger
from hydronic import C_to_F, F_to_C, ft

out   = environment( C_to_F( -20 ))    # -20C outdoors
world = space( 'world', ( 1e4, 1e4, 1e4 ), out, now=0 )
room  = space( 'room', ( ft(12), ft(14), ft(9) ),
               environment( C_to_F( 21 )), now=0 )
world.contains( room )
room.connects( portal( 'walls',   'world', ( ft(26), ft(9) ), 20 ))
room.connects( portal( 'window',  'world', ( ft(8), ft(5) ),   2 ))
room.connects( portal( 'ceiling', 'world', room.size,         50 ))

t_r = F_to_C( room.radiant( world ))  # 19.1C; the window is 10.6C
f   = fanger( clo=1.0, met=1.2, hum=0.4 )
f.PMV( t_a=21, t_r=t_r )              # -0.43: how it feels
f.setpoint( t_r=t_r )                 # 24.5C air would feel neutral
```

3 Comfort Control

A comfort thermostat targets a Predicted Mean Vote of zero, not an air temperature. In pyhydronic's simulation of two rooms sharing one radiant slab zone through a -20°C day: holding the glassy room's air at 21°C leaves it feeling slightly cool all day (PMV -0.3); holding its comfort at neutral settles near 22°C air, for 6% more fuel – while the snug room on the same zone overheats. A model makes these trade-offs visible: where comfort control costs energy, where it saves it (eg. over warm radiant floors), and where a room needs its own zone.

Radiant slabs respond over hours, not minutes. The same model, driven by a weather forecast, anticipates the next several hours' heat demand – so slabs are warm before the cold arrives, and are not still overheating rooms after the sun comes out.

4 Who Benefits

Thermostat makers A "comfort" mode and a "feels like" display, from a model of the room configured at installation; per-room comfort (PMV / PPD) reporting.

Hydronic and radiant control vendors Zone and outdoor-reset controllers that account for slab lag, and keep floors at the temperatures that make rooms comfortable.

Heat pump and HVAC manufacturers Lower air setpoints wherever surfaces are warm, for the same comfort; fewer "it's cold" service calls where they are not.

Building automation integrators Comfort as a measurable KPI per room, alongside energy.

5 What We Offer

Commercial license Embed pyhydronic in closed-source firmware, apps or cloud services.

Integration engineering Port the comfort and thermal models to your platform (C, C++, Rust, Python; embedded Linux or microcontrollers), and integrate with your sensors and controls.

Modeling and calibration Derive room models from installer-friendly inputs (floor plans, window areas, construction), and calibrate them against your field data.

Validation Verify your implementation against the ISO 7730 example data (Annex D), and against simulated buildings across climates.

Control strategy Comfort setpoints, zoning, slab-lag and forecast-driven predictive control.

Training and support For your engineering team, with ongoing support agreements.

6 Licensing

Pyhydronic is available from Dominion R&D Corp. under two licenses: the GNU GPL Version 3 (Open Source), and a Commercial (Closed Source) license.

6.1 GNU GPLv3

The GPL allows you to use the software in private scenarios (eg. development, testing and evaluation) free of charge and without obligation. Once you distribute software or devices based on pyhydronic, the GPL requires that you provide your users with the source code of your product, and of any changes to pyhydronic.

6.2 Commercial License

If you wish to keep your product's source code private, a Commercial license authorizes you to develop and deploy products using current and future versions of pyhydronic, in whatever form you desire:

> Dominion Research & Development Corp. hereby grants to <Your Company> a worldwide, non-revocable, non-exclusive, non-transferable and non-sublicensable Commercial License to use and deploy current and future versions of the pyhydronic library, in whatever form they desire.

7 How To Get Started

Try it now:

```
python3 -m pip install pyhydronic
git clone https://github.com/pjkundert/pyhydronic.git && cd pyhydronic
python3 examples/radiant_room.py --mode comfort
```

Then reach out to Perry Kundert at Dominion R&D Corp. (perry@dominionrnd.com) to discuss your product, a commercial license, and the engineering support that fits.