OpenPelt: Python Framework for Thermoelectric Temperature Control System Development

Roman Parise[1] and Georgios Is. Detorakis[1]

1 Bessel Technologies LLC

Summary

Thermoelectric coolers are semiconductor heat pumps that can be used in precision temperature control applications. After designing a thermoelectric temperature control system, the primary challenge is tuning and testing control algorithms. For instance, developing proportional-integral-derivative controllers involves tuning gains until the desired characteristic is observed, a tedious, time-consuming process. Furthermore, experimenting with new algorithms not only takes a long time, but may also run the risk of damaging the hardware. We propose a faster-than-real-time temperature control simulation library, called OpenPelt. OpenPelt contains utilities for developing and verifying temperature control algorithms as well as a model of a thermoelectric cooler to act as the plant. OpenPelt also enables exporting simulation results to Fenics to simulate the control system's impact on three-dimensional heat diffusion models.

Statement of need

Thermoelectric coolers (TECs) are semiconductor heat pumps used in various applications (Chein & Huang, 2004). For instance, when heat sinking and fans will not suffice in electronics cooling applications, a TEC can be used as an alternative. Similarly, TECs can be used to cool down components in lasers and volatile organic compound detectors [Mansour et al. (2006)](Mosier-Boss & Lieberman, 2003).

Designing TEC-based temperature control systems involves overcoming two primary challenges. One of these is hardware design. The other is control algorithm development, such as the tuning of proportional-integral-derivative controllers. The latter is particularly time-consuming. The time to reach a target temperature with a thermoelectric cooler can be on the order of a minute. To observe an oscillatory temperature characteristic fully settle can take a few additional minutes. In order to fully understand the performance of a control algorithm with particular parameters, one may need to endure these several minute delays repeatedly. The result is a very arduous process. Self-tuning algorithms exist in the literature that partially resolve these issues. However, it is wise to prototype any control algorithm in a simulation beforehand for faster-than-real-time result acquisition and to avoid potentially damaging the TEC hardware with overdrive scenarios.

Furthermore, traditional control theory is undergoing a revolution in light of developments in machine learning and artificial intelligence. In recent years, neural networks and even reinforcement learning algorithms have been applied to temperature controllers (Degrave et al., 2022). OpenPelt currently includes rudimentary support for developing such control algorithms. The repo contains an example randomized neural network test, which is included as a proof of concept for using neural networks in OpenPelt. Training the network is out of the scope of the present work. Furthermore, OpenPelt offers support for training reinforcement learning control algorithms. We have an example test case of a naive agent consuming random actions to

42 control the TEC plate's temperature. These capabilities in OpenPelt can enable future neural
43 control theory research, using a thermoelectric cooler as a test actuator.

Object-Oriented SPICE-based Thermoelectric Cooler Model

45 To our knowledge, no out-of-the-box, open source solution for TEC controller simulation exists.
46 This is partly due to the lack of a usable open source model of a thermoelectric cooler and
47 the lack of open source software to simulate it. We investigated the literature and found an
48 electro-thermal circuit model of a TEC (Chavez et al., 2000). Though the SPICE netlist is
49 publicly available, there is no straightforward approach for simulating a traditional, let alone
50 novel, control algorithm.

51 We wrote a circuit netlist for the TEC model and buried the implementation details in a
52 tec_plant class. We then developed test and controller objects that enable rapid testing of
53 control algorithms on instances of this tec_plant class. Firstly, one needs access to an open
54 source SPICE simulator, ngspice in our case. The simulator then needs to support external
55 current/voltage sources, so that a user's Python function can generate the driving value to the
56 simulator on each timestep given the temperatures measured in previous timesteps. Prior to
57 our development effort, the external current source functionality in ngspice was broken. We
58 fixed the bug, thereby enabling such sources.

59 We felt it was critical that users be able to write high-level, object-oriented code and use it to
60 interact with the TEC model. We used the PySpice library as a wrapper around ngspice to
61 export the simulator's shared library functionality to Python (Salvaire, n.d.).

62 We default the tec_plant model parameters, such as thermal capacitance and resistance
63 values, to the values in the original paper. However, users are able to modify them with their
64 experimental results or theoretical approximations. In the future, we would like to provide
65 detailed models of circuit component values and have the class infer those parameters based
66 on the geometric and material properties of the TEC described. We would also like to be able
67 to provide a physical model of the temperature sensor and its coupling to the TEC. Currently,
68 controllers have no overshoot due to the ideality of the sensor and its coupling to the TEC.

69 Furthermore, the model is useful for relatively low-power control of the TEC. However, at
70 higher power levels, the TEC heats up and thermal drift affects the circuit parameters. This is
71 also a desirable future feature, but there are unlikely very many applications in which this is
72 useful.

73 We ultimately chose a circuit model for two reasons. Firstly, circuit models are computationally
74 simple and efficient since they are solved using iterative methods to solve systems of differential
75 equations. However, a more subtle advantage is that electrical, thermal, hydraulic, mechanical,
76 etc. systems can all be modeled using circuits. Thus, the library could in principle be extended
77 to support control systems with a variety of actuators without the simulator becoming any
78 more complicated and without the use of additional simulators. Users need only develop a
79 circuit model for the actuator of interest. This also enables reuse of controller algorithms and
80 test code for systems of diverse control parameters, such as a cell incubator requiring various
81 gas concentrations, humidity levels, and temperature to be controlled.

82 The tec_plant model supports preliminary three-dimensional finite element simulation as well.
83 Class methods enable the user to incorporate the results of the controller simulation into a
84 three-dimensional model described using the
85 Fenics library (A. Logg, 2012) (Logg & Wells, 2010). For the time being OpenPelt supports
86 only legacy Fenics. However, it's up to end-users if they would like to use the most modern
87 Fenics implementation. Thus, users can see how the TEC interacts with more complex systems.

Sample Results

We reproduced figure 11 from the original paper using OpenPelt and a controller that drives a constant 2.1A current ([Chavez et al., 2000](#)).
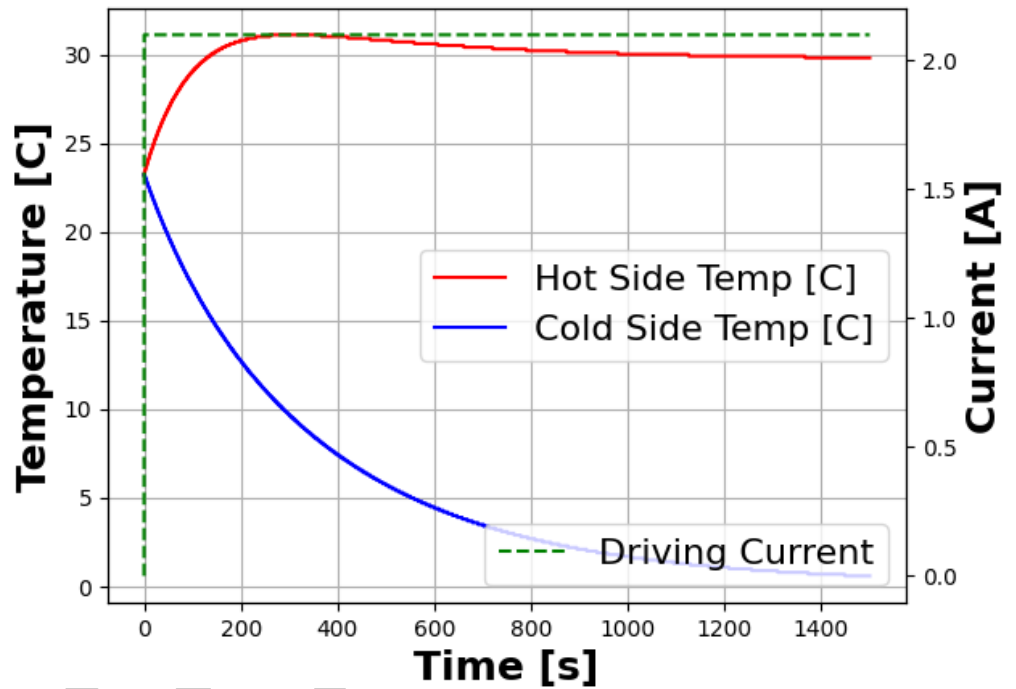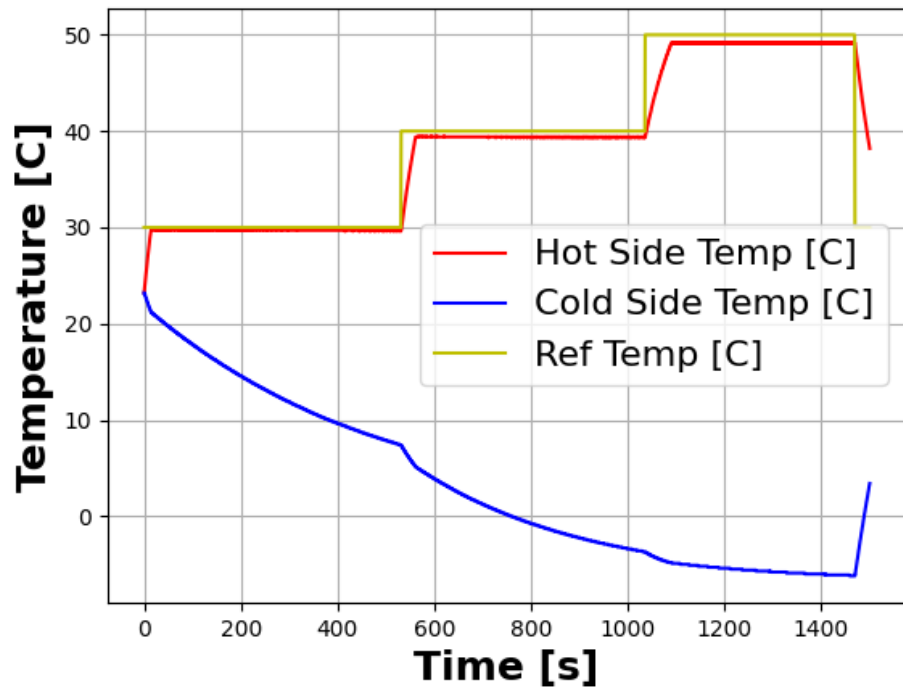


```
plate_select = OpenPelt.TECPlate.HOT_SIDE
pC = OpenPelt.tec_plant("Detector",
                        lambda t , Th_arr : 2.1@u_A,
                        OpenPelt.Signal.CURRENT,
                        plate_select)
pC.run_sim()
pC.plot_th_tc(OpenPelt.IndVar.TIME)
plt.show()
```

We have also developed proportional temperature controllers using OpenPelt. OpenPelt provides functionality for cycling through different reference temperatures in a test sequence.

```
102
103     plate_select = OpenPelt.TECPlate.HOT_SIDE
104     pC = OpenPelt.tec_plant("Detector",
105                             None,
106                             OpenPelt.Signal.VOLTAGE,
107                             plate_select=plate_select,
108                             steady_state_cycles = 400)
109     cbs = OpenPelt.circular_buffer_sequencer([30.00, 40.00, 50.00], pC.get_ncs())
110     pidc = OpenPelt.pid_controller(cbs, 15.00, 0.00, 0.00, plate_select=plate_select)
111     pC.set_controller_f(pidc.controller_f)
112     pC.run_sim()
113     pC.plot_th_tc(OpenPelt.IndVar.TIME, plot_driver = False, include_ref = True)
114     plt.show()
```
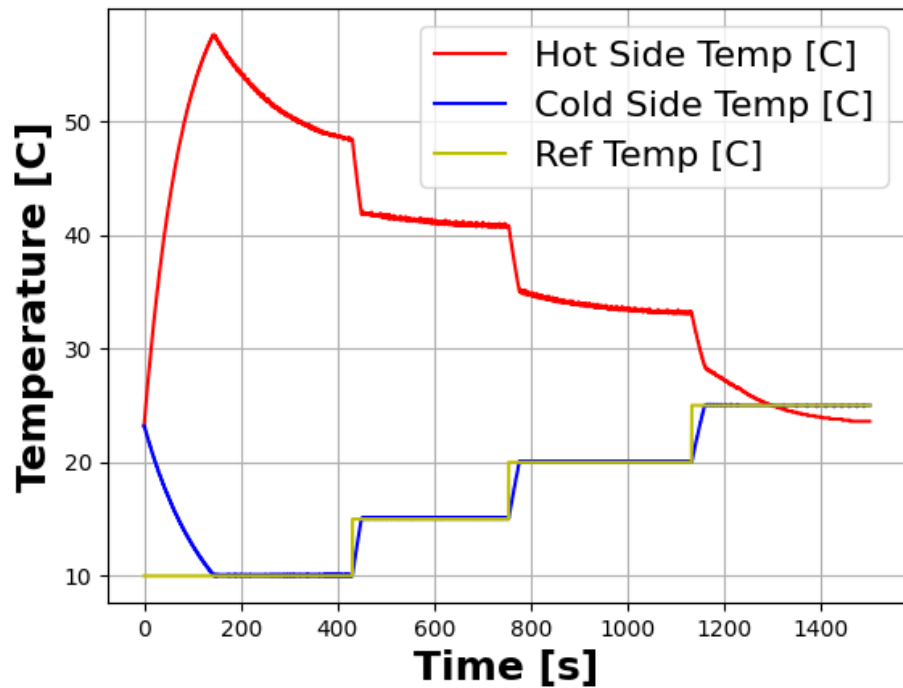
```
115
116        plate_select = OpenPelt.TECPlate.COLD_SIDE
117        pC = OpenPelt.tec_plant("Detector",
118                                None,
119                                OpenPelt.Signal.VOLTAGE,
120                                plate_select=plate_select)
121        cbs = OpenPelt.circular_buffer_sequencer([10.00, 15.00,
122                                                  20.00, 25.00],
123                                                 pC.get_ncs())
124        pidc = OpenPelt.pid_controller(cbs, -150.00, 0.00, 0.00,
125                                       plate_select=plate_select)
126        pC.set_controller_f(pidc.controller_f)
127        pC.run_sim()
128        pC.plot_th_tc(OpenPelt.IndVar.TIME, plot_driver = False, include_ref = True)
129        plt.show()
```

Bibliography

A. Logg, G. N. W. et al, K.-A. Mardal. (2012). *Automated solution of differential equations by the finite element method.* Springer. https://doi.org/10.1007/978-3-642-23099-8

Chavez, J. A., Ortega, J. A., Salazar, J., Turo, A., & Garcia, M. J. (2000). SPICE model of thermoelectric elements including thermal effects. *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference [Cat. No. 00ch37066]*, *2*, 1019–1023 vol.2. https://doi.org/10.1109/IMTC.2000.848895

Chein, R., & Huang, G. (2004). Thermoelectric cooler application in electronic cooling. *Applied Thermal Engineering*, *24*(14), 2207–2217. https://doi.org/10.1016/j.applthermaleng.2004.03.001

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., Las Casas, D. de, & others. (2022). Magnetic control of tokamak

142   plasmas through deep reinforcement learning. *Nature*, *602*(7897), 414–419.

143   Logg, A., & Wells, G. N. (2010). DOLFIN: Automated finite element computing. *ACM*
144   *Transactions on Mathematical Software*, *37*. https://doi.org/10.1145/1731022.1731030

145   Mansour, K., Qiu, Y., Hill, C., Soibel, A., & Yang, R. (2006). Mid-infrared interband
146   cascade lasers at thermoelectric cooler temperatures. *Electronics Letters*, *42*, 1034–1035.
147   https://doi.org/10.1049/el:20062442

148   Mosier-Boss, P. A., & Lieberman, S. H. (2003). Detection of volatile organic compounds
149   using surface enhanced raman spectroscopy substrates mounted on a thermoelectric cooler.
150   *Analytica Chimica Acta*, *488*(1), 15–23. https://doi.org/10.1016/S0003-2670(03)00676-7

151   Salvaire, F. (n.d.). *PySpice*. https://pyspice.fabrice-salvaire.fr