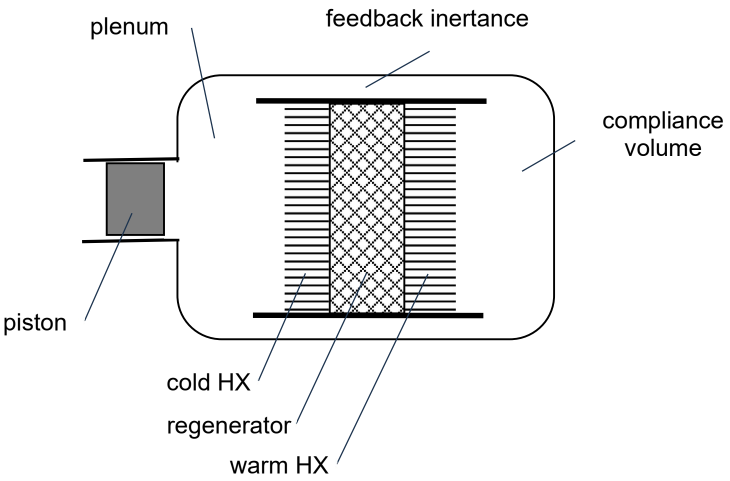
Sage Model Notes

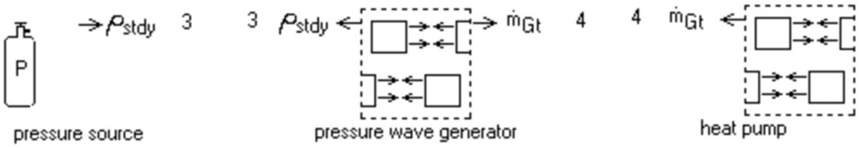
ThermoacousticHeatPump.scfn

D. Gedeon
3 May 2026

A model for a thermoacoustic heat pump of the type designed to reject heat at a relatively high temperature for industrial processes[1].



The root-level Sage model contains a pressure source and two submodels:



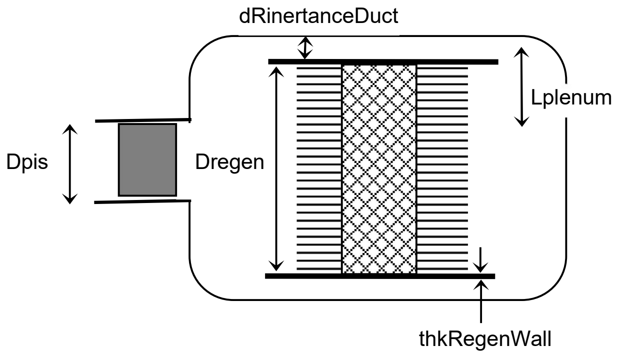
It also includes some user-defined inputs and outputs. The inputs and first several outputs are used to recast built-in inputs of the submodels, according to the diagram below. As always, using the Tools | Explore Custom Variables dialog is the best way to trace these relationships and understand how the model works. The last several outputs summarize the heat pump performance.

```
Inputs
  Tcold            cold-end temperature (K)                   3.000E+02
  Twarm            warm-end temperature (K)                   4.000E+02
  Dpis             piston diameter (m)                        1.000E-01
  XpAmpDesign      design piston amplitude (m)                1.000E-02
  Dregen           regenerator matrix diameter (m)            1.125E-01
  Lregen           regenerator length (m)                     5.654E-02
  thkRegenWall     regen wall thickness (m)                   1.000E-03
  dRinertanceDuct  radial flow gap (m)                        5.000E-03
```

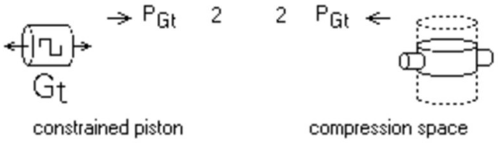
---

[1] M.E.H. Tijani and J.A. Lycklama à Nijeholt, *Thermoacoustic heat pump for very high temperature applications*, 14th IEA Heat Pump Conference, (2023)

```
Outputs
  Apis             piston frontal area                      7.854E-03
    0.25*Pi*Sqr(Dpis)
  Aregen           regenerator  frontal area                9.941E-03
    0.25*Pi*Sqr(Dregen)
  IDPwall          pressure wall ID                         1.245E-01
    Dregen + 2*(thkRegenWall + dRinertanceDuct)
  Apwall           cross section area                       1.217E-02
    0.25*Pi*Sqr(IDPwall)
  Lplenum          effective flow length                    3.475E-02
    0.5*IDPwall - 0.5*dRinertanceDuct - 0.25*Dpis
  Wp               net PV power                             5.000E+02
    Wpwg
  Qin                                                       3.438E+02
    QinHP
  Qrej             net heat rejection                      -8.438E+02
    QrejHP
  heatingCOPpv     PV COP heating mode                      1.688E+00
    -Qrej/Wp
  coolingCOPpv     PV COP cooling mode                      6.875E-01
    Qin/Wp
```



Pressure Wave Generator



The pressure wave generator consists of a constrained-motion piston that displaces volume within a compression space. The piston motion is set by the recast `FX` input:

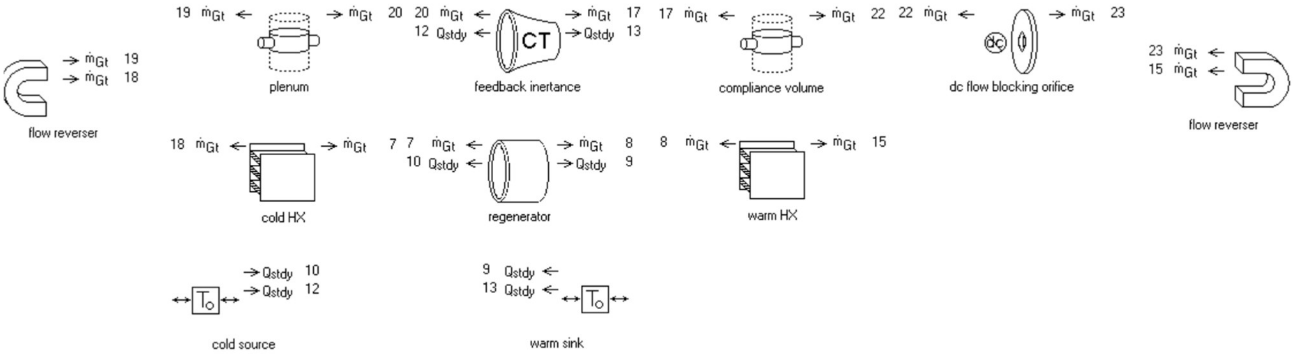
```
Recasts
 FX = 0.000E+00...
    (XpAmpDesign) Amp
    (0.000E+00) Arg
```

In other words, the piston amplitude is the `XpAmpDesign` input of the root model and the phase angle is zero. The recast A input of the *pos-facing area* component inside the *constrained piston* defines the piston frontal area:

```
Recasts
  A = Apis
Outputs
  A                face area (m2)                           7.854E-03
```

A more refined model might replace the *constrained piston* with a *reciprocator* driven by a rotary mechanism or a linear actuator, depending on the actual hardware design.

Heat Pump

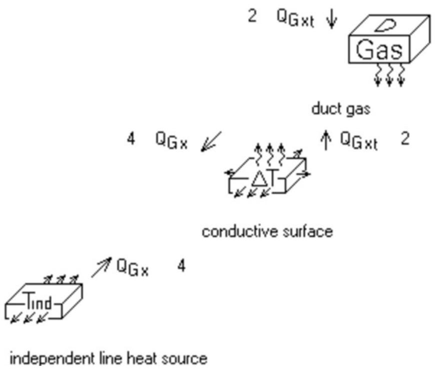


The core components of the *heat pump* submodel are arranged roughly as shown in the drawing at the beginning of this document. The components in the second row (*cold HX, regenerator, warm HX*) constitute the main components of the stirling thermodynamic cycle. The components in the first row (*plenum, feedback inertance, compliance volume*) function as the displacer in a conventional stirling-cycle machine and also a means to transfer PV power to the far end, which functions like the compression space of a conventional stirling cycle. *Flow reverser* components turn the flow between the two into a closed loop. A mass flow connector within the *cold HX* is raised to the root level for connection to the *pressure wave generator*.

The *dc flow blocking orifice* is an optional component, used to prevent DC flow circulation, the bane of closed-loop thermoacoustic machines. DC flow can lead to lower efficiency or unstable performance. In a Sage model it can also destabilizing the solution process. In this particular model, removing the *dc flow blocking orifice* causes a catastrophic drop in performance, although changing the DC flow value (input `DCRhoUA` by a small amount in the right direction can be helpful. Since DC flow in actual hardware depends on design details like duct entrance conditions and temperature stability it is not captured very accurately by a Sage model. Some amount of experimentation controlling dc flow effects can be expected in a hardware development project.

The *cold source* and *warm sink* components anchor the conduction walls of the *regenerator* and *feedback inertance* components. They contain user-defined outputs `QcoldSource` and `QWarmSink`, exported to the submodel level as part of the `QinHP` and `QrejHP` energy calculations.

The *cold HX* and *warm HX* components are essentially identical, except for their boundary temperatures. Each consists of a *duct gas* component connected to a *conductive surface*, representing an aluminum fin conduction path (defined by the `Solid` input of the *conductive surface*), anchored to an *independent line heat source* that imposes a temperature boundary condition at the base of the fins. (see sample model *HeatExchangers-SimpleIsotherma*l)



The fin conduction path length of the *conductive surface* is recast to the channel height input *Hchan*:

```
Recasts
  D = Hchan
```

The boundary temperature is the `Tsrc` cubic spline input for the *independent line heat source* recast for the *cold HX* as

```
Recasts
  Tsrc = unit spline...
    (0.000E+00, Tcold)
    (1.000E+00, Tcold)
```

Net heat transfer to the heat source is available in user-defined output `QchxHp`, exported to the submodel level as part of the `QinHP` energy calculation. And similarly for the *warm HX*.

The fluid-filled coolant lines that transfer heat to external heat exchangers are not part of the model. They must be designed independently to correspond to the boundary temperatures of the *independent line heat source* components.

The *regenerator* contains a stainless-steel *random-fiber matrix* and a *pressure wall* representing thermal conduction down the outer wall.

The *random fiber matrix* inputs are

```
Inputs
  Porosity         porosity (void/total)                      9.000E-01
  Dfiber           fiber diameter (m)                         2.500E-05
```

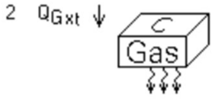
Other matrix types are available. The reason for using random fibers is that its heat-transfer and flow-resistance calculations are more accurate at high porosity than those for woven-screen regenerators, and the Sage optimizer wants the porosity to be high (90%), which is difficult to achieve with woven screen.

In a stirling-cycle machine the pressure amplitude (output `FPMean` of the *matrix gas* component inside the *random fiber matrix* component) and the mass flow rate (output `FRhoUAmean`) should be close to in phase to maximize PV power flow with minimum windage loss. In this model that is not the case, suggesting the "displacer" components (*plenum, feedback inertance, compliance volume*) are not functioning very well. Possibly because of the competing requirement to transfer compression-space PV power to the far end. You can experiment with the effects of "optimizing" the regenerator by constraining the value of this user-defined output to -1:

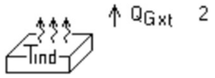
```
  PdotM            cosine  P -  Mdot phase                  -5.870E-01
    (FPMean.Cos.1*FRhoUAmean.Cos.1 + FPMean.Sin.1*FRhoUAmean.Sin.1) /
(FPMean.Amp.1 * FRhoUAmean.Amp.1)
```

The defining expression evaluates the dot product of the `FPMean` and `FRhoUAmean` phasors and divides by their amplitudes according to the usual geometrical definition of dot product in terms of two vectors and the angle between them. You could also directly constrain difference of the phases `FPMean.Arg.1` and `FRhoUA.Arg.1` to -180 degrees but that risks a sudden shift of a phase near -180 degrees to suddenly jumping to one near 180 degrees.

The *plenum* and *compliance volume* components are modeled as fixed volume *generic cylinders* with isothermal walls.



cylinder-space gas



independent isothermal surface

Adiabatic walls (thick surface component) are also possible but seem to adversely affect solution stability in this model. Net heat transfers to the walls are available in user-defined outputs `QplenHp` and `QcmplHp`, exported to the submodel as part of the `QinHP` and `QrejHP` energy calculations.

The length and wetted-surface inputs of the *cylinder-space gas* are recast in terms of root-level inputs as:

```
Recasts
  Length = Lplenum
  Swet = 2*Apwall
```

`Length` represents the effective flow length. It is important because the inertial pressure drop of the helium flowing inside (important in thermoacoustic machines) depends on the passage length and cross section area, which is defined as `Volume / Length` in a *cylinder-space gas*, where `Volume` is a built-in input. In order that the model resolve the variation of pressure along the length the *plenum* and *compliance volume* components are divided into 5 computational cells (`NCell` input).

Input `Swet` represents the surface area exposed to the back-and-forth boundary heat flow that produces so-called hysteresis loss. The bottom-line performance is not very sensitive to this value so only a rough approximation is required. Surface area in passages with hydraulic diameters less than the thermal penetration depth need not be included. Thermal penetration depth is given by the formula.

$$\delta = \sqrt{2\,\alpha / \omega}$$

Where α is thermal diffusivity and ω is operating angular frequency.

The *inertance duct* component is modeled as an equivalent number of round tubes with the same wetted perimeter and cross-section area as the actual annular passage, by recasting inputs

```
Recasts
  Dtube = unit spline...
    (0.000E+00, 2*dRinertanceDuct)
    (1.000E+00, 2*dRinertanceDuct)
  Ntube = IDpwall / dRinertanceDuct - 1
```

It is modeled with Sage's confusingly named *compliance duct* component since there is a temperature gradient along the wall and this component models some of the streaming losses in such ducts, in particular boundary and streaming convection, captured in these two outputs.

```
  QoscMean         mean boundary convection (W)             -5.707E+00
  QstrMean         mean streaming convection (W)            -3.492E+01
```

Regarding the `Dtube` and `Ntube` recasts, the formulation is derived from an annulus with outer diameter D and radial thickness dR. The perimeter and cross-section area work out to

$$W = 2\pi(D - dR)$$

and

$$A = \pi(DdR - dR^2)$$

On the other hand, the perimeter and cross section area of N round cylinders of diameter d are

$$\mathrm{W} = N\,\pi d$$

and

$$\mathrm{A} = \mathrm{N}\frac{\pi}{4}\mathrm{d}$$

Equating the left-hand sides and solving for d and N gives

$$d = 2\,\mathrm{dR}$$

and

$$N = D/dR - 1$$

Optimization

This model supports an optimization process, currently setup to maximize cooling power subject to a fixed PV power of 500 W. This is equivalent to maximizing the cooling COP. Also the heating COP since it is just 1 more than the cooling COP. The summary optimization specification (Tools | Explore Optimization) is:

```
Objective Function
```

```
  Maximize Qin

OPTIMIZED VARIABLES                 SUBJECT TO CONSTRAINTS

Thermoacoustic Heat Pump
  Freq                              Wp = 500
  dRinertanceDuct                   Qin  > = 10
  Dregen                            dRinertanceDuct > = 5.0E-3
  Lregen

3 heat pump

3.5 regenerator

3.5.1 random fiber matrix
  Porosity                          Porosity < = 0.90

3.7 compliance volume
  Volume

3.8 feedback inertance
  Length
```

The inequality constraints are there to prevent the optimization from drifting into forbidden territory. If they are satisfied they have no effect on the process.

Optimizing `dRinertanceDuct` and *feedback inertance* `Length` provides slack to optimize the *feedback inertance*. Optimizing the `Volume` input of the *compliance volume* provides additional slack for tuning the thermoacoustic circuit, as does optimizing frequency `Freq`.

Notably, charge pressure (`Pcharge` of *pressure source*) and *constrained piston* diameter and stroke amplitude (`Dpis` and `XpAmpDesign`) are fixed in this model. They and other inputs may be optimized as part of designing an actual heat pump for a particular application.