Implementation of an Air-Based Building Integrated PV/Thermal Collector (BIPV/T) Model

Kamel Haddad, Sébastien Brideau, Anil Parekh, 1 Hannel Drive, Ottawa, ON, K1A 1M1,*
*Primary Contact: kamel.haddad@canada.ca, *(613) 947-9822*
Natural Resources Canada, CanmetENERGY-Ottawa

Overview

Currently, E+ includes PV models with building integration capabilities, and a simple PV/T model with constant thermal efficiency. This design document provides details on the implementation of an advanced BIPVT model in EnergyPlus. First the mathematical formulation of the model is provided. Followed by the needed changes to the IDD and source code files.

Mathematical model for BIPV/T air collector

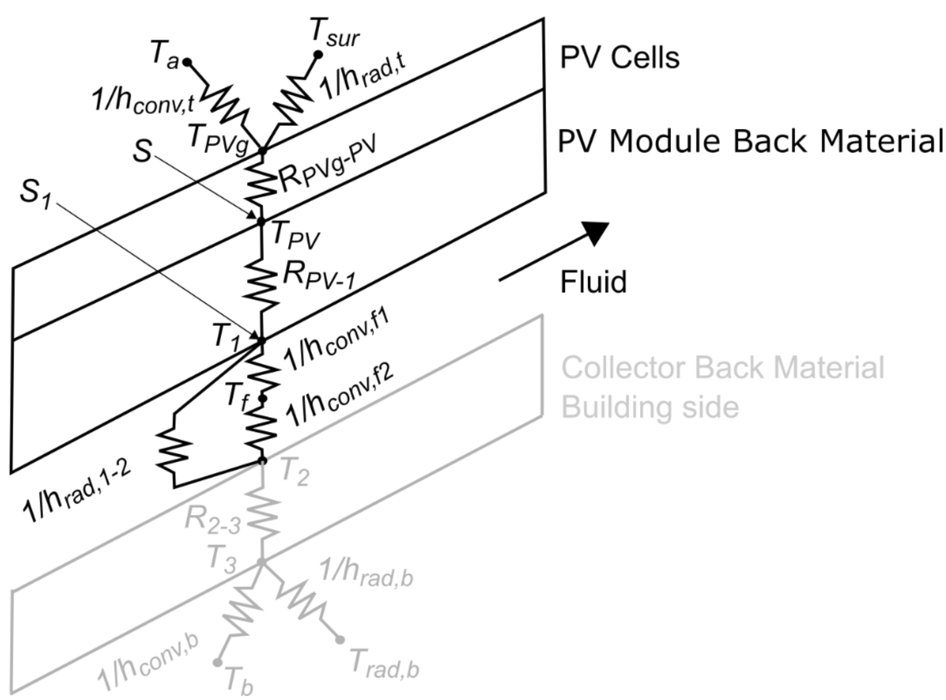

Figure 1 Resistance circuit representation of BIPV/T collector - adapted from Deslisle and Kummert, 2014. Light grey portion is representative of building side and will be solved by E+ building solver.

The heat balances for the various layers of the BIPVT collector discussed here is taken from work by Delisle and Kummert (2014), but the solution methodology is different (discussed in a later section). The thermal resistance circuit representation is shown in Figure 1.

The following equations describe energy balances on the various layers of interest. These are the PV glazing outer layer, the PV cells, the PV backing outer surface, and the air cavity. The energy balances are for a length (along flow direction) of collector dx.

Energy Balances:

PV glazing outer surface:

$$0 = h_{conv,t}(T_{PVg} - T_a) + h_{rad,t}(T_{PVg} - T_{sur}) + \left(\frac{T_{PVg} - T_{PV}}{R_{PVg-PV}}\right) \quad [1]$$

PV cells:

$$S + \left(\frac{T_{PVg} - T_{PV}}{R_{PVg-PV}}\right) = \left(\frac{T_{PV} - T_1}{R_{PV-1}}\right) \quad [2]$$

PV backing outer surface:

$$\left(\frac{T_{PV} - T_1}{R_{PV-1}}\right) + S_1 = h_{conv,f1}(T_1 - T_f) + h_{rad,1-2}(T_1 - T_2) \quad [3]$$

Air in cavity:

$$\dot{m}C_p \frac{dT_f}{dx} = \left[h_{conv,f1}(T_1 - T_f) + h_{conv,f2}(T_2 - T_f)\right] \cdot W \quad [4]$$

Where

$$S = IAM_{PV}(\tau\alpha)_{PV,N} G F_{cell} - \eta_{PV} G \quad [5]$$

$$S_1 = IAM_{bs}(\tau\alpha)_{bs,N} G(1 - F_{cell}) \quad [6]$$

Let

$$h_{PVg-PV} = \frac{1}{R_{Vg-PV}} \quad [7]$$

$$h_{PV-1} = \frac{1}{R_{PV-1}} \quad [8]$$

The radiative heat transfer coefficients are given by:

$$h_{rad,t} = \varepsilon_{PVg}\sigma(T_{PVg}^2 + T_{surr}^2)(T_{PVg} + T_{surr}) \quad [9]$$

$$h_{rad,1-2} = \frac{\sigma(T_1^2 + T_2^2)(T_1 + T_2)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} \quad [10]$$

The absorptance-transmittance product modifier:

$$IAM_{PV} = 1 - b_{0,PV}\left(\frac{1}{\cos\theta} - 1\right) - b_{1,PV}\left(\frac{1}{\cos\theta} - 1\right)^2 \quad [12]$$

$$IAM_{bs} = 1 - b_{o,bs}\left(\frac{1}{\cos\theta} - 1\right) - b_{1,bs}\left(\frac{1}{\cos\theta} - 1\right)^2 \qquad [13]$$

When the flow is turbulent inside the air channel, the heat transfer coefficients $h_{conv,f1}$ and $h_{conv,f2}$ are calculated using the Dittus Boetler equation for the Nusselt Number:

$$Nu = 0.023Re^{0.8}Pr^{n} \qquad [14]$$

where n = 0.4 for heating and n = 0.3 for cooling. When the flow is laminar, a constant Nusselt Number (3.66) is used.

Solving the energy balances

There are two general approaches typically used to solving the energy balance equations for collector models. Delisle and Kummert take a finite difference approach and discretize the collector along the fluid motion. The advantage of this is method is that the calculation of the PV cell temperature at every node might improve the accuracy of the PV model. The alternative approach is to solve for the average temperatures along the flow direction analytically. This approach is taken by Duffie and Beckman and in various models described by TESS for TRNSYS. This methodology has for advantage to be faster than a full numerical solution. However, it is possible that the results are slightly less accurate due to the use of average temperatures to calculate PV efficiency. The relative accuracy of either approach is unclear at this time, and they have both been used extensively in the literature. Because of this it was decided to use the analytical solution approach.

Summing Equations [1] through [3] and subtracting Equation [4] gives:

$$\frac{\dot{m}C_p}{W}\frac{dT_f}{dx} = S + S_1 + h_{conv,f2}\left(T_2 - T_f\right) + h_{conv,t}\left(T_a - T_{PVg}\right) + h_{rad,t}\left(T_{sur} - T_{PVg}\right) + h_{rad,1-2}(T_2 - T_1)$$

[15]

Let

$$A = S + S_1 + h_{conv,t}\left(T_a - T_{PVg}\right) + h_{rad,t}\left(T_{sur} - T_{PVg}\right) + h_{rad,1-2}(T_2 - T_1) \qquad [16]$$

$$D = A + h_{conv,f2}T_2 \qquad [17]$$

$$B = -h_{conv,f2} \qquad [18]$$

Therefore;

$$\dot{m}C_p\frac{dT_f}{dx} = DW + BWT_f \qquad [19]$$

Rearrange and integrate both sides for the entire length of the collector (length=L)

$$\int_{x=0}^{x=L}\frac{dT_f}{T_fB+D} = \int_{x=0}^{x=L}\frac{W}{\dot{m}C_p}dx \qquad [20]$$

For this solution, we assume that T_{PVg} and T_1 are not dependent on distance x ($i.e.$ T_{PVg} and T_1 are average temperatures over the collector length, $\bar{T}_{PVg}$ and $\bar{T}_1$), Equation 20 gives Equation 21.

$$T_f(L) = \left(T_f(0) + \frac{D}{B}\right) e^{BWL/\dot{m}C_p} - \frac{D}{B} \quad [21]$$

To find fluid temperature at any point x, replace L with distance x.

Average fluid temperature:

$$\bar{T}_f = \frac{1}{L}\int_{x=0}^{x=L} T_f(x)dx = \left[\frac{\dot{m}C_p\left(BT_f(0)+D\right)e^{BWx/\dot{m}C_p} - BDWx}{B^2W}\Bigg|_{x=0}^{x=L}\right]\frac{1}{L} \quad [22]$$

$$\bar{T}_f = \frac{\dot{m}C_p}{B^2WL}\left(BT_f(0) + D\right)e^{BWL/\dot{m}C_p} - \frac{D}{B} - \frac{\dot{m}C_p}{B^2WL}\left(BT_f(0) + D\right) \quad [23]$$

The solution of Equation 23 is the average air temperature over the length of the collector, and we have previously assumed that all other layer's temperatures are not dependent on x (i.e. they are average temperatures over the length of the collector). If we also assume that $T_f(0)$, $\bar{T}_2$, S, S_1 are known, we can solve equations 1, 2 and 3. Equations 1-3 can be re-written as the following:

$$\begin{bmatrix} h_{conv,t} + h_{rad,t} + h_{PVg-PV} & -h_{PVg-PV} & 0 \\ h_{PVg-PV} & -h_{PVg-PV} - h_{PV-1} & h_{PV-1} \\ 0 & h_{PV-1} & -h_{PV-1} - h_{conv,f1} - h_{rad,1-2} \end{bmatrix} \begin{bmatrix} \bar{T}_{PVg} \\ \bar{T}_{PV} \\ \bar{T}_1 \end{bmatrix} = \begin{bmatrix} h_{conv,t}T_a + h_{rad,t}T_{sur} \\ -S \\ -S_1 - h_{rad,1-2}\bar{T}_2 - h_{conv,f1}\bar{T}_f \end{bmatrix} \quad [24]$$

This can then be solved with matrix inversion.

The PV temperature will be passed directly to the proper "Photovoltaics.cc" subroutine to calculate the PV efficiency.

The steps to solve each time steps are:

1. Assume (guess) values $\bar{T}_f$ and $\bar{T}_{PV}$. If not the first time step, use previous time step values.
2. Get current values of all boundary conditions
3. Get PV efficiency from PV model using $\bar{T}_{PV}$ from step 1.
4. Update all the coefficients and constants in Equation [24]
5. Solve Equation [24] (Gives values for $\bar{T}_{PVg}$, $\bar{T}_{PV}$, $\bar{T}_1$)
6. Solve Equation [23] (Gives $\bar{T}_f$)
7. Iterate steps 2-6 until convergence.

8. Calculate $T_f(L)$ using Equation [21]
9. Calculate $\dot{Q} = \dot{m}C_p\left(T_f(L) - T_f(0)\right)$

Current EnergyPlus PVT Modeling Approach

A PVT system in EnergyPlus is modeled using object:

“SolarCollector:FlatPlate:PhotovoltaicThermal”

One of the inputs to this object is “Photovoltaic-Thermal Model Performance Name”. Currently the only option available for this input is an object of type:

“SolarCollectorPerformance:PhotovoltaicThermal:Simple”

This object uses a fixed or scheduled thermal efficiency for the PVT system. The new model for BIPVT will be based on a new IDF object named:

“SolarCollectorPerformance:PhotovoltaicThermal:BuildingIntegratedPVT”

Another input to object “SolarCollector:FlatPlate:PhotovoltaicThermal” is “Photovoltaic Name” which is a reference to an object of type “Generator:Photovoltaic”. The photovoltaic generator object has three methods for calculating the electrical performance of the PV cells: “Simple”, “EquivalentOne-Diode”, and “Sandia”. The photovoltaic generator object also has a parameter “Heat Transfer Integration Mode” that can specify a link to a “SolarCollector:FlatPlate:PhotovoltaicThermal” object.

Changes to EnergyPlus IDD File

The new IDF object:

‘SolarCollectorPerformance:PhotovoltaicThermal:BuildingIntegratedPVT”

will be entered in the EnergyPlus IDD file. Below is a description of each of the fields in the IDD file for this new object and, if applicable, the name of the variable in the model associated with input. Some of the descriptions are taken from IDD file entries for the following two objects:

‘SolarCollector:UnglazedTranspired’

‘SolarCollector:FlatPlate:PhotovoltaicThermal’

as they are similar to what this work is trying to achieve. This model will only work for air-based thermal collectors, although the ‘SolarCollector:FlatPlate:PhotovoltaicThermal’ allows for water as well.

‘SolarCollectorPerformance:PhotovoltaicThermal:BuildingIntegratedPVT’

1. **Field: Name**
   This field contains a unique name for the BIPVT solar collector

2. **Field: Boundary Conditions Model Name**
   This field contains the name of a 'SurfaceProperty:OtherSideConditionsModel' object declared elsewhere in the input file. The "Type of Modelling" for this object will be set to "GapConvectionRadiation". This will connect the collector to the exterior boundary conditions for the underlying heat transfer surface in the EnergyPlus building envelope model.
3. **Field: Availability Schedule Name**
   This field contains the name of a schedule to indicate when the solar collector is available. When the schedule value is 0, the collector will be bypassed. When the value is greater than 0 the collector is available to provide heat recovery. If this field is left blank, it is assumed that the collector is available.
4. **Field: Effective Gap Plenum Behind PV modules**
   This field is used to enter a nominal gap thickness for the collector. This is used to calculate the convective heat transfer coefficient behind the PV modules and on the building wall surface.
5. **Field: Effective Overall Height of Collector**
   This field is used to enter the nominal height of the collector. This is defined as the distance from the inlet, (typically at the bottom of the collector), to the outlet (typically at the top of the collector).
   Model variable name: L
6. **Field: Effective Overall Width of Collector**
   This field is used to enter the nominal width of the collector.
   Model variable name: W
7. **Field: PV Transmittance-Absorptance Product**
   This field is used to enter PV Normal Transmittance-Absorptance Product. This value is typically not known and literature gives values between approx. 0.8 to 0.9. Default value is 0.87.
   Model variable name: $(\tau\alpha)_{PV,N}$
8. **Field: Backing Material Normal Transmittance-Absorptance Product**
   This field is used to enter Backing Material Normal Transmittance-Absorptance Product. This value is typically not known. Dependent on backing color. Values in absorptivity for tedlar in literature vary between approx. 0.39 and 0.94. This would yield Transmittance-Absorptance of between approx. 0.37 and 0.87. Default is set to 0.87.
   Model variable name: $(\tau\alpha)_{bs,N}$
9. **Field: Fraction of collector gross area covered by PV cells**
   This field is used to enter the Fraction of collector gross area covered by PV cells. Generally around 0.85 but can vary.
   Model variable name: F_{cell}
10. **Field: PV module thermal resistance - Top**

This field is used to enter the PV module thermal resistance above the cells. This value is the total thermal resistance of the encapsulating glass and EVA above the cells and is typically between 0.0035 and 0.0052 m^2·K/W. Default is set to 0.0044 m^2·K/W.
Model variable name: R_{Vg-PV}

11. **Field: PV module thermal resistance - Bottom**
    This field is used to enter the PV module thermal resistance below the cells. This value is the total thermal resistance of the encapsulating tedlar and EVA below the cells and is typically between 0.0031 and 0.0046 m^2·K/W. Default is set to 0.0039 m^2·K/W.
    Model variable name: R_{PV-1}
12. **Field: Emissivity PV modules**
    This field is used to enter the emissivity of the PV modules. Usually not known. Default is 0.85 for typical PV modules.
    Model variable name: $\varepsilon_{PV,g}$
13. **Field: Emissivity of backing material**
    This field is used to enter the emissivity of the backing material. Usually not known. Default is 0.9 (black tedlar).

File “PhotovoltaicThermalCollectors.hh” List of Changes

A new structure will be added to this file named “BIPVTModelStruct” within “namespace PhotovoltaicThermalCollectors”. The structure will declare all the needed variables associated with the new model IDD object “SolarCollectorPerformance:PhotovoltaicThermal:BuildingIntegratedPVT”.

File “PhotovoltaicThermalCollectors.cc” List of Changes

1. Subroutine “GetPVTcollectorsInput”: Add code to read all the IDF file input variables for object: “SolarCollectorPerformance:PhotovoltaicThermal:BuildingIntegratedPVT”. Also implement any diagnostic messages related to these inputs.
2. Subroutine “PVTCollectorStruct::calculate()”: Add code for implementation of the new BIPVT model as described in Equations 1-25.
3. Subroutine “PVTCollectorStruct::control()”: Add reference to new BIPVT object.
4. Subdroutine “PVTCollectorStruct::update()”: Add code to update variables for object “OtherSideConnectionModel”: *Tconv*, *Hconv*, *Trad*, and *Hrad*.

File “Photovoltaics.cc” List of Changes

1. Add subroutine “GetPVTmodelIndex” to get index for object “SolarCollector:FlatPlate:PhotovoltaicThermal” associated with object “Generator:Photovoltaic”. Also update appropriate header file to refer to this new subroutine.
2. Subroutine “GetPVInput()”: Invoke subroutine “GetPVTmodelIndex” to link object “Generator:Photovoltaic” to associated object “SolarCollector:FlatPlate:PhotovoltaicThermal” through the surface name on which “Generator:Photovoltaic” is mounted.
3. Add subroutine “GetBIPVTTsColl” to get temperature of the BIPVT surface. Update appropriate header file to refer to this subroutine.
4. Subroutine “CalcSandiaPV”: Add call to subroutine “GetBIPVTTsColl” to get temperature of BIPVT surface.

5. Subroutine “CalcTRNSYSPV”: Add call to subroutine “GetBIPVTTsColl” to get temperature of BIPVT surface.

Input Output Reference Documentation

A new section will added to the I/O documentation for the new object: “SolarCollectorPerformance:PhotovoltaicThermal:BuildingIntegratedPVT”.

Engineering Reference Documentation

A new section will be added to the Engineering documentation to on the new BIPVT model added.

Example File

A new example file will be created to demonstrate the new BIPVT model.

References

Delisle, V. and Kummert, M. 2014. A novel approach to compare building-integrated photovoltaic/thermal air collectors to side-by-side PV modules and solar thermal collectors. Solar Energy, vol. 100, pp. 50-65.

Duffie, J. and Beckman, W. 2013. Solar Engineering of Thermal Processes. 4th Edition. John Wiley and Sons. New York.