Modeling and Simulation of a Combined Solar and Wind Systems using OpenModelica

Arash M. Dizqah¹  Alireza Maheri¹  Krishna Busawon¹  Peter Fritzson²

¹ School of Computing, Engineering and Information Sciences, Northumbria University, NE1 8ST Newcastle Upon Tyne, UK
² PELAB Programming Environment Lab, Dep. Of Computer and Information Science, Linköping University, SE-581 83 Linköping, Sweden
Outline

- Combined/hybrid wind and solar energy systems (HRES)
- An overview on the mathematical and electrical modeling
- System modeling and simulation using OpenModelica
- Analysis and discussion
- Conclusion and future works

Majority of the results presented here have been already published in:

Combined/hybrid wind and solar energy systems

- Climate changes caused by greenhouse effect

Renewable Energy Sources

- Rise in price of fossil fuel
- Exhaustive fissile fuel
- Energy Security

Hybrid Renewable Energy System (HRES)

More than one type of Sources of Power

HRES Motivations
The PV, wind turbine, and the battery modules are nonlinear.

The PV, wind turbine, load, and the battery modules introduce algebraic constraints.

The battery module is hybrid and has at least two modes of operation, i.e., charging and discharging modes.

The converter is also a hybrid system including a high frequency state transition. However, in this study an average model has been used for simplicity’s sake.
An overview on the mathematical/electrical modeling

The PV module equivalent electrical circuit and the I-V curve

\[ I_{pv} = I_{ph} - I_0 \left\{ \exp \left( \frac{V_{pv} + R_s I_{pv}}{n_d N_s q \frac{1}{K T_c}} \right) - 1 \right\} - \frac{V_{pv} + R_s I_{pv}}{R_{sh}}. \]

![Diagram of PV module equivalent electrical circuit and the I-V curve]
The battery equivalent electrical circuit and operating modes

\[
\frac{d\text{charge}}{dt}(t) = \frac{1}{3600} I_{\text{bat}}(t).
\]

\[
V_{\text{bat}} = \begin{cases} 
V_0 - R I_{\text{bat}} + V_{\text{exp}} - \frac{P_1 C_{\text{max}}}{C_{\text{max}} - \text{charge}} I_{\text{bat}} & \text{mode=charging,} \\
V_0 - R I_{\text{bat}} + V_{\text{exp}} + \frac{P_1 C_{\text{max}}}{C_{\text{max}} - \text{charge}} I_{\text{bat}} + \frac{P_1 C_{\text{max}}}{0.1 C_{\text{max}}} I_{\text{bat}} & \text{mode=discharging} 
\end{cases}
\]
The boost-type converter electrical circuit and the average model

\[
\begin{bmatrix}
\dot{i}_L(t) \\
v_C(t)
\end{bmatrix} = \begin{bmatrix}
\frac{R_L}{L} - \frac{R_C(1-D)}{L(1+R_C/R)} & -\frac{1-D}{L(1+R_C/R)} \\
\frac{1-D}{C(1+R_C/R)} & \frac{1}{RC(1+R_C/R)}
\end{bmatrix} \begin{bmatrix}
\dot{i}_L(t) \\
v_C(t)
\end{bmatrix} + \begin{bmatrix}
\frac{1}{L} \\
0
\end{bmatrix} V_{in}(t).
\]

\[
V_{out}(t) = \begin{bmatrix}
\frac{R_C(1-D)}{1+R_C/R} & \frac{1}{1+R_C/R}
\end{bmatrix} \begin{bmatrix}
\dot{i}_L(t) \\
v_C(t)
\end{bmatrix}.
\]
The combined/hybrid power generation plant

Non-manipulated variables (perturbations): \([S_x, T_x, U_x, L_x]\)

Manipulated variables: \([\beta, D_w, D_s]\)

Load \(L_x\)

It is ...

- A system describing with nonlinear HDAEs of Index-1
- A MIMO system
System modeling and simulation using OpenModelica

The energy system needs to be modeled for...

- Simulation and analysis
- Being used in **model-based controllers**

**Modelica model** → **System model**

**Employed approach #1 ...**

- Object-oriented design
- Providing a library of Modelica classes

**Employed approach #2 ...**

- Flat HDAE model
- Minimum number of equations
- More suitable for model-based controllers
Modeling the system as Modelica classes

The Modelica model of the whole solar power system

```modelica
model HRES SolarSystem
    Modelica.Blocks.Sources.Constant Sx(k = 300.0); Modelica.Blocks.Sources.Constant Tx(k = 298.15); Modelica.Electrical.Analog.Basic.Ground ground;
    HRES.PVArray pvarray(Npv=10);
    Modelica.Blocks.Sources.TimeTable DutyCycle(table = [...]); Modelica.Blocks.Sources.Step ramp(startTime = 5, offset = 10, height = -6);
    HRES.BoostConverter converter(Rl = 0.001, Rc = 0.3, L = 0.0005,C = 0.005);
    HRES.LeadAcidBattery battery1; HRES.LeadAcidBattery battery2; HRES.LeadAcidBattery battery3; HRES.LeadAcidBattery battery4;

    equation
        connect(Tx.y,pvarray.Tx); connect(Sx.y,pvarray.Sx); connect(pvarray.n,ground.p); connect(pvarray.p,converter.p1); connect(converter.n1,converter.p2);
        connect(battery1.n,battery2.p); connect(battery2.n,battery3.p); connect(battery3.n,battery4.p); connect(battery4.n,ground.p);
    end HRES SolarSystem;
```

The lead-acid battery Modelica class; the red ellipse indicates a segment that handles the mode transition events.

```modelica
class LeadAcidBattery
    ... PositivePin p; NegativePin n; discrete Boolean chargeState(start = true); ...

    equation
        chargeState = if ibat = 0 then true else false;
        der(charge) = (1/3600) * ibat;
        der(V exp) = if chargeState then P2 /3600 *abs(i) *(P3 - V exp)
            else -(P2 *abs(i))/3600 *V exp;
        when change(chargeState) and pre(chargeState) then
            tmp = if not chargeState then
                pre(vbat) - V0 - R *pre(ibat) - ...
            else 0;
            reinit(Vexp, tmp);
        end when;
        soc = 1 – charge/Cmax;
        vbat = if chargeState then
            V 0-R ibat-(P1*Cmax)/(Cmax-charge) *charge-
            else V 0-Ribat-(P1*Cmax)/(Cmax-charge)*charge-
        ...
    end LeadAcidBattery;
```
Modeling the system as Modelica classes

The simulated I-V (solid-line) and P-V (dashed-line) curves of the KC200GT PV module and empirical points provided by the manufacturer (the circle markers)

Validating the PV module simulation results

- The developed PV model has been simulated separately.
- The simulation results validated with the available data in manufacturer datasheet.
- It follows accurately the empirical data available by the manufacturer.
- The simulated MPP is matched to the empirical data provided by the manufacturer (26.3V, 7.61A).
- The datasheet of the PV module is available from www.kyocerasolar.com/assets/001/5195.pdf
Validating the battery simulation results

- The developed Modelica model for Panasonic LC-R127R2PG battery has been simulated separately for all zones.
- The battery Modelica model validated with the available data in manufacturer datasheet.
- According to the simulation scenario, battery is charging for 100 minutes and then it is discharged.
- Discharging with the average current of 7.2A, it takes around 35 min to reach the cut-off voltage (10.2V). It matches perfectly with datasheet.
- The datasheet of the battery is available from [www.farnell.com/datasheets/1624915.pdf](http://www.farnell.com/datasheets/1624915.pdf)

The simulated (a) battery voltage, (b) battery current, and (c) the SOC of the battery.
The simulated voltage and the SOC (State of Charge) of the batteries. The simulated generated and consumed powers.

Simulation scenario

- Before t=5min, the generated power by the PV module experiences a stepwise decrease due to manipulating the control signal, D.
- The load demand suddenly exceeds the generated power at t=5 min.
- The generated power by the PV array declines at t = 6min by manipulating the control signal, D.
The combined solar/wind plant
Modelica model

class HRES_wr

RealInput Sx "The solar irradiance (W/m²)";
RealInput Tx "The ambient temperature (K)";
RealInput Ux "The wind speed (m/s)";
RealInput Rx "The load demand (ohm)";
RealInput beta "The pitch angle (degree)";
RealInput Ds "The boost converter duty-cycle [0,1]";
RealInput Dw "The buck converter duty-cycle [0,1]";

... equation

der(Tc) = 1 / Ct * ((ta - eta) * Sx - Ul * (Tc - Tx));
iPV = iph_Tc_Sx - i0_Tc * (exp((iPV * rs_Tc + vPV) / a_Tc) ;

... Tm = -(cP * (Ux / 12) ^ 3 * 24.3 * Pnom) / wr / 24.3;
cP = (C1 * (C2 / lambdai - C3 * beta - C4) * exp(-C5 / lambdai)

... Te = -9.6 * iWIND * Dw;
der(wr) = (Te - Tm - F * wr) / J;

... vBAT_STACK = Dw * ((1.35 * P * psi * sqrt(3) * ...;

... algorithm

when {change(chargeState) and not pre(chargeState)} then

noOfEqCycles:=noOfEqCycles + 1 - soc;
end when;

end HRES_wr;
Modeling as a flat HDAE using OpenModelica

Simulation results
## Analysis and discussion

### The proposed model vs. the equivalent SIMULINK/SimPowerSystem model for the optimal energy management problems

<table>
<thead>
<tr>
<th>Performance criteria*</th>
<th>The equivalent SIMULINK/ SimPowerSystem model</th>
<th>The proposed Modelica model using OpenModelica</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time (with the step-size of 100 nsec)**</td>
<td>Around 10 hrs for 3 sec of simulation</td>
<td>Around 8 hours for 3 sec of simulation</td>
</tr>
<tr>
<td>Simulation time (for 3600 sec)</td>
<td>Impractical: It is not easy to remove the PWM and make it fast.</td>
<td>Around 30 sec (with the step-size of 720 usec)</td>
</tr>
<tr>
<td>Flexibility***</td>
<td>It cannot be integrated into the collocation method. It is not easy to be integrated into the multiple shooting method.</td>
<td>Adding the “smooth” function or converting to a MPCC problem, it can be used for the OCP applications.</td>
</tr>
</tbody>
</table>

* It is just a **rough comparison** for this application. It is not the results of a systematic comparison.

** The equivalent SIMULINK model consists of **PWM** modules with the frequency of **100 KHz** that causes it to be very slow and memory expensive. While for this application, it is not straightforward to replace the converters with the average model in SIMULINK, it has been done in the proposed Modelica model that make it much faster.

*** For **OCP** applications
Analysis and discussion

The proposed model vs. the equivalent SIMULINK/SimPowerSystem model

The equivalent SIMULINK/SimPowerSystem model

- Simulating the equivalent electrical circuit
- Very slow & memory expensive for the OCP applications (with PWM)

The proposed Modelica model using OpenModelica

- Much faster for the OCP applications
- More flexible for the OCP applications
- Applicable for the model-based controllers
Conclusion and Future works

Structure and characteristics

- **Wind branch**: Wind turbine + Generator + Rectifier + Converter
- **Solar branch**: PV panel + Converter
- **Storage**: Battery bank
- Nonlinear HDAE of Index-1

Modeling

- **Modelica** language has been employed.
- A library of the modelica components has been developed.
- A flat HDAE model has been developed as well.

Simulation results

- The **OpenModelica** tool has been used.
- The complete system has been *simulated*.
- The simulation results have been *verified* with the information available in datasheets.

Future works

- Combining OpenModelica and CasAdi to design nonlinear optimal controllers
- Optimal energy management

Hybrid wind/solar energy system
References

Thank You