Fan Efficiency based on Euler Method - Integration Tips -

IBPSA Modelica Working Group

Coordination Meeting

March 01, 2023

Hongxiang (Casper) Fu
Building Technology and Urban Systems
Lawrence Berkeley National Laboratory
hcasperfu@lbl.gov

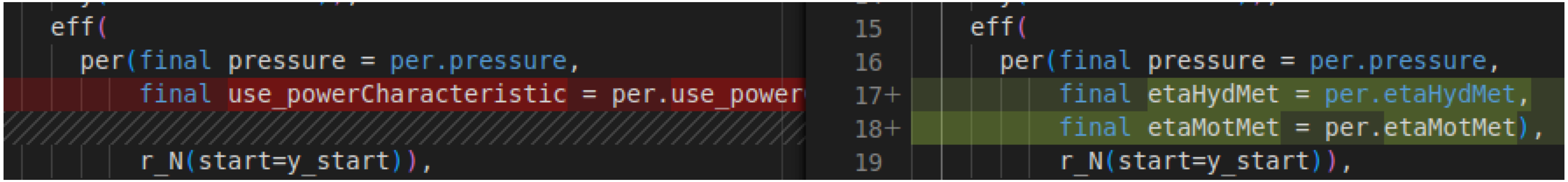
Overview & Recap

- Status: Merged
  - IBPSA issue #1645 “Fan efficiency with Euler number”
  - IBPSA PR #1646 “fan efficiency euler”
- Recap

  $$\eta = \eta_{hyd}\,\eta_{mot}$$

  - Variable hydraulic efficiency η_{hyd} estimated by Euler number
  - Variable motor efficiency η_{mot} estimated by DOE generic curves
  - Preconfigured fan/pump models

Integration Tips
Replaced Parameters

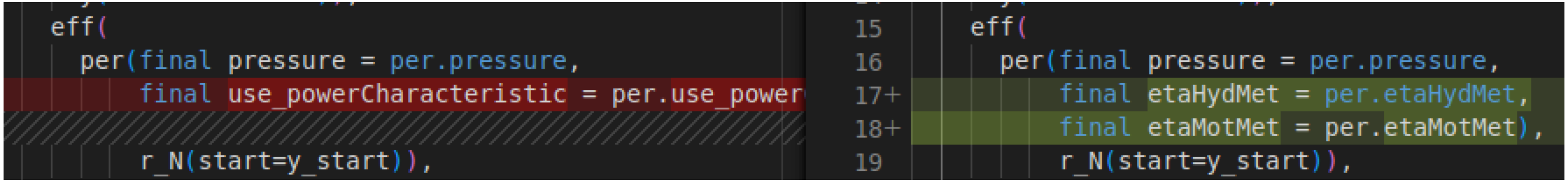


```
eff(
  per(final pressure = per.pressure,
      final use_powerCharacteristic = per.use_power
      r_N(start=y_start)),
```

```
eff(
  per(final pressure = per.pressure,
      final etaHydMet = per.etaHydMet,
      final etaMotMet = per.etaMotMet),
      r_N(start=y_start)),
```

- `use_powerCharacteristic` is deprecated.
  - Removed in `Movers.Data.Generic`.
- Replaced by enumerations:
  - `Movers.BaseClasses.Types.HydraulicEfficiencyMethod etaHydMet`
  - `Movers.BaseClasses.Types.MotorEfficiencyMethod etaMotMet`

Integration Tips
Replaced Parameters

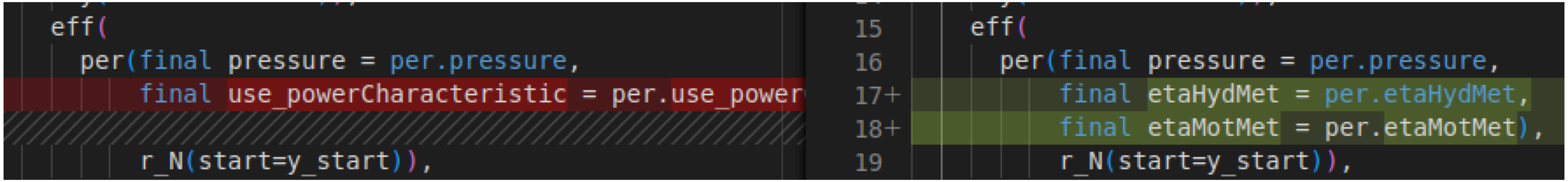


```
eff(
  per(final pressure = per.pressure,
      final use_powerCharacteristic = per.use_power
      r_N(start=y_start)),
```

```
eff(
  per(final pressure = per.pressure,
      final etaHydMet = per.etaHydMet,
      final etaMotMet = per.etaMotMet),
      r_N(start=y_start)),
```

- `Movers.BaseClasses.Types.HydraulicEfficiencyMethod etaHydMet`
  - `NotProvided` – Computed from other efficiency terms or constant (0.7 or 0.49)
  - `Efficiency_VolumeFlowRate` – User provides array of $\eta_{hyd}(\dot{V})$
    - Same as `use_powerCharacteristic == false`
  - `Power_VolumeFlowRate` – User provides array of $P(\dot{V})$
    - Same as `use_powerCharacteristic == true`
  - `EulerNumber` – Uses the Euler Number method
    - DEFAULT
    - User can provide the peak value, or let the model make a *guesstimation*.

Integration Tips
Replaced Parameters

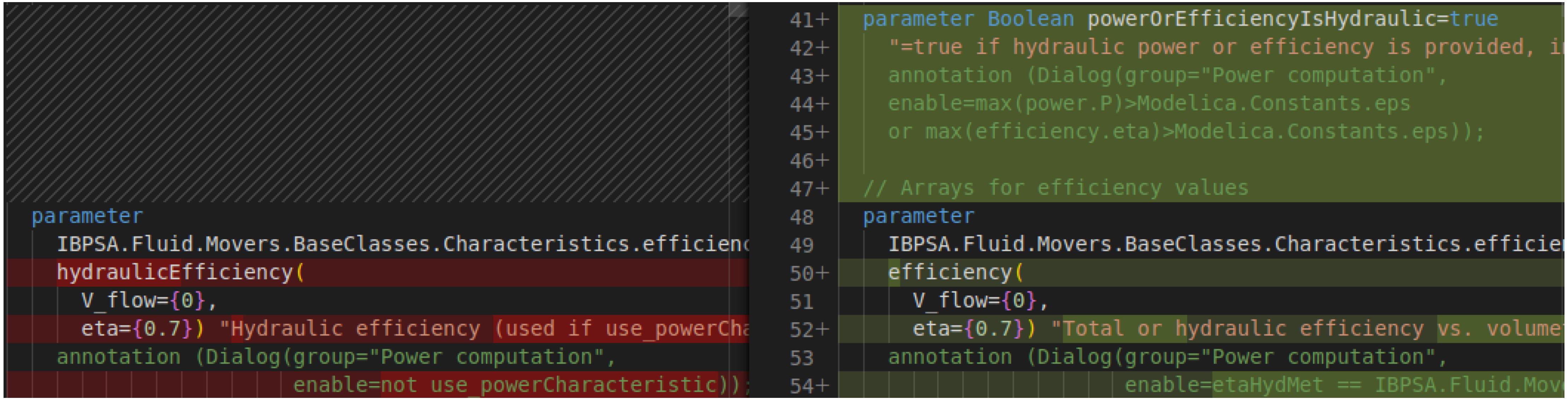


```
eff(
  per(final pressure = per.pressure,
    final use_powerCharacteristic = per.use_power
    r_N(start=y_start)),
```

```
eff(
  per(final pressure = per.pressure,
    final etaHydMet = per.etaHydMet,
    final etaMotMet = per.etaMotMet),
    r_N(start=y_start)),
```

- `Movers.BaseClasses.Types.MotorEfficiencyMethod etaMotMet`
  - `NotProvided` – Computed from other efficiency terms or constant (0.7)
  - `Efficiency_VolumeFlowRate` – User provides array of $\eta(\dot{V})$
    - Same as `use_powerCharacteristic == false`
  - `Efficiency_MotorPartLoadRatio` – User provides array of $\eta_{mot}(\dot{W}_{hyd}/\dot{W}_{mot,nominal})$
    - User can provide rated motor power $\dot{W}_{mot,nominal}$, or the model makes a *guesstimation*.
  - `GenericCurve` – Model generates $\eta_{mot}(\dot{W}_{hyd}/\dot{W}_{mot,nominal})$ based on DOE curves
    - DEFAULT (with exceptions, see `usersguide`)
    - Same treatment for $\dot{W}_{mot,nominal}$ as above.

Integration Tips
Other Notable Parameters



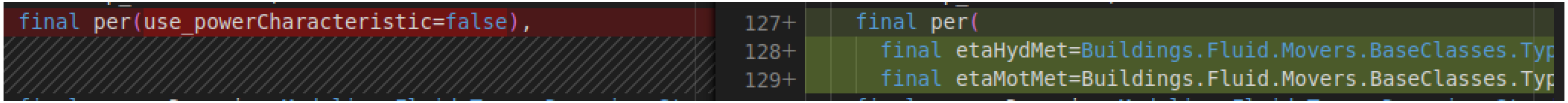
```
parameter
  IBPSA.Fluid.Movers.BaseClasses.Characteristics.efficienc
  hydraulicEfficiency(
    V_flow={0},
    eta={0.7}) "Hydraulic efficiency (used if use_powerCha
  annotation (Dialog(group="Power computation",
                     enable=not use_powerCharacteristic));
```

```
parameter Boolean powerOrEfficiencyIsHydraulic=true
  "=true if hydraulic power or efficiency is provided, i
  annotation (Dialog(group="Power computation",
  enable=max(power.P)>Modelica.Constants.eps
  or max(efficiency.eta)>Modelica.Constants.eps));

// Arrays for efficiency values
parameter
  IBPSA.Fluid.Movers.BaseClasses.Characteristics.efficie
  efficiency(
    V_flow={0},
    eta={0.7}) "Total or hydraulic efficiency vs. volume
  annotation (Dialog(group="Power computation",
                     enable=etaHydMet == IBPSA.Fluid.Mov
```

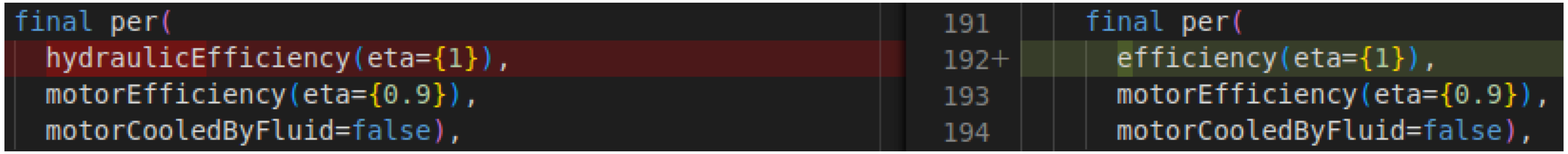
- powerOrEfficiencyIsHydraulic
  - Used to indicate the provided power or efficiency is hydraulic or total.
    - $\underset{\texttt{false}}{\eta} = \underset{\texttt{true}}{\eta_{hyd}}\, \eta_{mot}$
  - Changes equation selections in Movers.BaseClasses.FlowMachineInterface.
- For this reason, hydraulicEfficiency is renamed efficiency.

Integration Tips
Typical Refactoring Examples



```
final per(use_powerCharacteristic=false),
```

```
  final per(
    final etaHydMet=Buildings.Fluid.Movers.BaseClasses.Typ
    final etaMotMet=Buildings.Fluid.Movers.BaseClasses.Typ
```



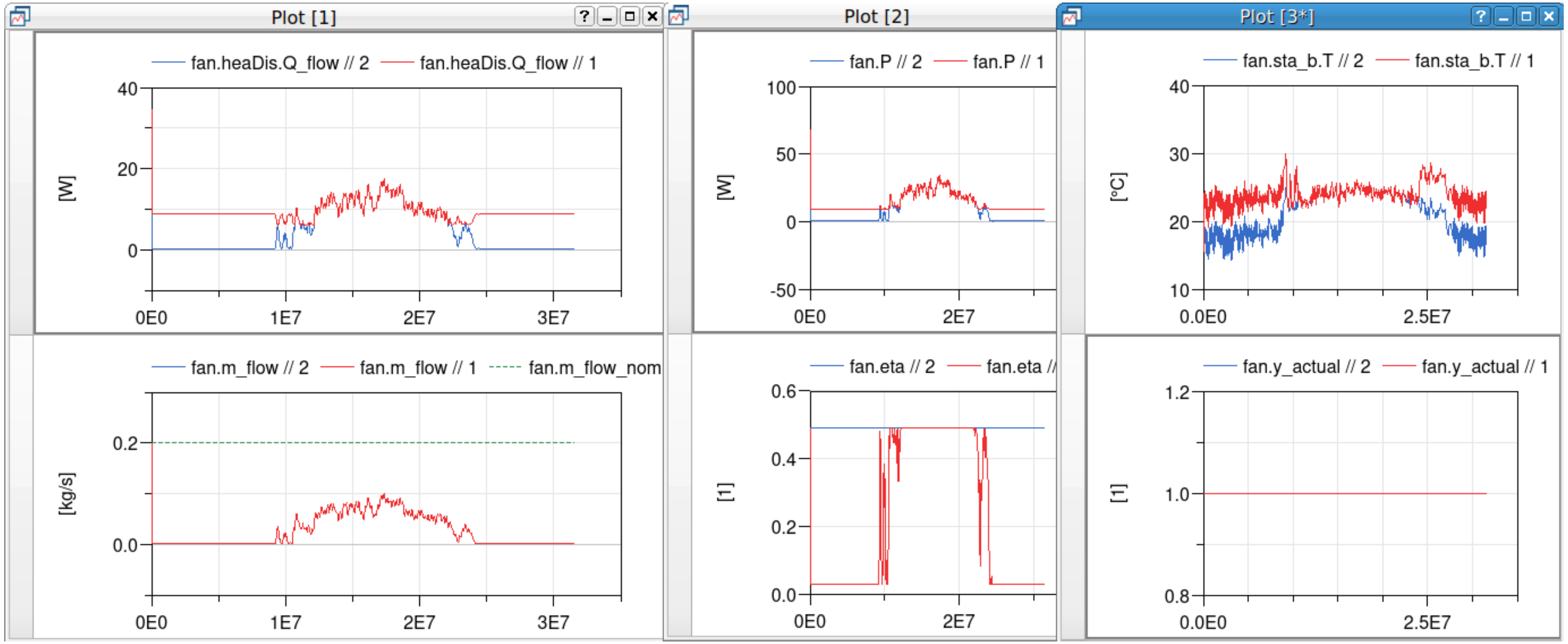
```
final per(
  hydraulicEfficiency(eta={1}),
  motorEfficiency(eta={0.9}),
  motorCooledByFluid=false),
```

```
final per(
  efficiency(eta={1}),
  motorEfficiency(eta={0.9}),
  motorCooledByFluid=false),
```

Integration Tips
Examples

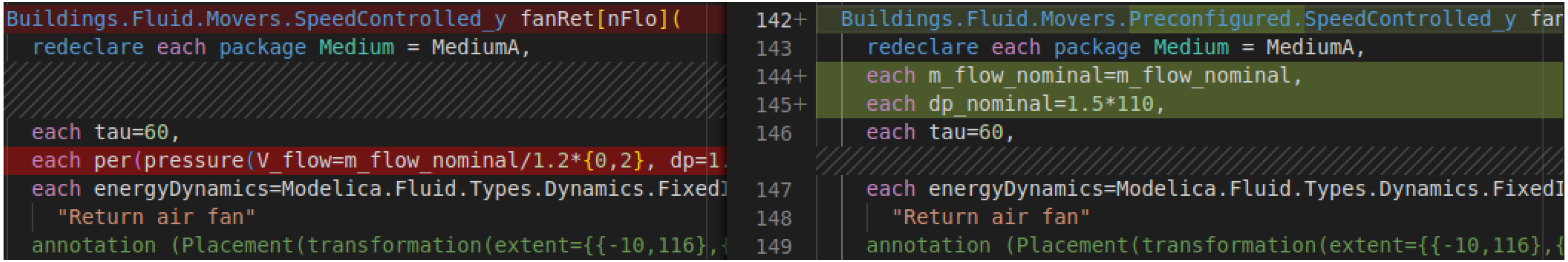
- Refer to these models as examples:
  - `Movers.Validation.PowerSimplified`
  - `Movers.Validation.PowerExact`
  - `Movers.Validation.PowerEuler`

Integration Tips
Watch Out



- The new method can give a VERY low estimation for efficiency.
- This *usually* indicates problems with model parameterisation.

Using the Preconfigured Models Replacing Dummy Pressure Curves



```
Buildings.Fluid.Movers.SpeedControlled_y fanRet[nFlo](
  redeclare each package Medium = MediumA,

  each tau=60,
  each per(pressure(V_flow=m_flow_nominal/1.2*{0,2}, dp=1
  each energyDynamics=Modelica.Fluid.Types.Dynamics.Fixed
    "Return air fan"
  annotation (Placement(transformation(extent={{-10,116},
```

```
 Buildings.Fluid.Movers.Preconfigured.SpeedControlled_y fa
   redeclare each package Medium = MediumA,
   each m_flow_nominal=m_flow_nominal,
   each dp_nominal=1.5*110,
   each tau=60,

   each energyDynamics=Modelica.Fluid.Types.Dynamics.Fixed
     "Return air fan"
   annotation (Placement(transformation(extent={{-10,116},
```

- Thank you!

HCasperFu@lbl.gov

Hongxiang (Casper) Fu
Building Technology and Urban Systems
Lawrence Berkeley National Laboratory
hcasperfu@lbl.gov