Documentation

Matlab Codes

PumpModel.mat

1. Function name:

PumpModel.mat

2. Model description:

PumpModel.mat is a matlab function developed for the simulation of volumetric pumps. Three modelling paradigms are currently implemented to simulate the pump, i.e.

- CstEff: a "constant efficiency" model into which constant values for both the isentropic efficiency and the volumetric efficiency are provided by the user, i.e.
  - $\varepsilon_{vol} = \frac{\dot{V}_{pp}}{\dot{V}_{th}} = \bar{\varepsilon}_{vol}$
  - $\varepsilon_{is} = \frac{\dot{m}_{pp}\ (h_{ex,is} - h_{su})}{\dot{W}_{pp}} = \bar{\varepsilon}_{is}$

- PolEff: a "polynomial efficiency" model into which quadratic polynomial regressions are used to evaluate both the volumetric efficiency and the isentropic efficiency, i.e.
  - $\varepsilon_{vol} = \sum_{i=0}^{2} \sum_{j=0}^{2} a_{ij} N_{pp}^{j} \left(\frac{P_{ex}}{P_{su}}\right)^{i}$
  - $\varepsilon_{is} = \sum_{i=0}^{2} \sum_{j=0}^{2} b_{ij} N_{pp}^{j} \left(\frac{P_{ex}}{P_{su}}\right)^{i}$

- SemiEmp: a "semi-empirical" model based on a physics-based approach. On the one hand, the mass flow rate is evaluated as the ideal flow rate obtainable by the pump diminished by an internal recirculation flow rate. The leakage are modeled by means of an incompressible flow through an equivalent orifice, i.e.
  - $\dot{m}_{pp} = \left[\rho_{su} N_{pp} V_{s,pp}\right] - \left[A_{lk}\sqrt{2\rho_{su}(P_{ex} - P_{su})}\right]$

  On the other hand, the mechanical power is calculated as the isentropic power added by mechanical losses. These losses being themselves computed by means of constant losses added to a term proportional to the isentropic power, i.e.
  - $\dot{W}_{pp} = \left[\dot{W}_0 + K_0 \dot{V}_{su}(P_{ex} - P_{su})\right] + \left[\dot{V}_{su}(P_{ex} - P_{su})\right]$

3. Model inputs:

The model inputs are the following ones:

- P_su (Pa) : inlet pressure of the pump;
- P_ex (Pa) : outlet pressure of the pump;
- h_su (J/kg) : inlet enthalpy of the pump;
- fluid (-) : fluid name;
- N_pp (rpm) : pump rotational speed
- param : a structure variable that contains the model parameters (see next section)

4. Model parameters:

Depending of the type of model chosen by the user, *param* will need to include the following variables:

- if param.modelType = 'CstEff':
    - param.V_s (m^3) : machine displacement volume
    - param.V (m^3) : pump casing volume
    - param.epsilon_is (-): isentropic efficiency
    - param.epsilon_vol (-): volumetric efficiency

- if param.modelType = 'PolEff':
    - param.V_s (m^3), machine displacement volume
    - param.V (m^3) : pump casing volume
    - param.N_pp_nom, pump nominal shaft speed
    - param.coeffPol_is (-), polynomial coefficients for epsilon_is
    - param.coeffPol_vol (-), polynomial coefficients for epsilon_vol

- if param.modelType = 'SemiEmp':
    - param.V_s (m^3), machine displacement volume
    - param.V (m^3) : pump casing volume
    - param.A_leak (m^2), leakage surface area
    - param.W_dot_loss (W), constant power losses
    - param.K_0_loss (-), term for the proportional losses

5. Model outputs:

The outputs are two structure variables, namely *out* and *TS*, and they contain the following sub-variables:

- **out** (structure variable with all the relevant model outputs)
    - T_ex (K): outlet temperature of the pump;
    - h_ex (J/kg): outlet temperature of the pump;
    - m_dot (kg/s) : the fluid mass flow rate;
    - M (kg) : fluid mass inside the pump;
    - W_dot (W) : the pump mechanical power consumption;
    - epsilon_is (-) : the pump isentropic efficiency;
    - epsilon_vol (-) : the pump volumetric efficiency;
    - time (sec) : the simulation time
    - flag (-): the model flag (+1 if the model run correctly, -1 if not)
    -
- **TS** (structure variable only used to plot the Ts diagram)
    - T (K) : vector of the fluid temperatures
    - s (J/kg.K) : vector of the fluid entropies

6. External function requirements:

The user must install CoolProp (http://www.coolprop.org/) to run PumpModel.

7. Matlab version:

This code has been developed under Matlab R2015a

8. Contact:

For any further information, please contact one of the main developers of ORCmKit:

- Rémi Dickes (rdickes@ulg.ac.be) – University of Liège (Belgium)
- Davide Ziviani (davide.ziviani@ugent.be) – Ghent University (Belgium)