Model Name: h6b

Description: H6b hydro turbine governor model

Prerequisite: h6bd data management model ahead of this model in the input data file

generator model ahead of this model in the input data file

Inputs: Shaft speed
Generator electrical power

Parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tw</td>
<td>water inertia time constant, sec</td>
<td>2.00</td>
</tr>
<tr>
<td>ptp</td>
<td>user-defined gate boost input, pu</td>
<td>0.02</td>
</tr>
<tr>
<td>ftp</td>
<td>frequency deviation for transient boost to pick up, pu</td>
<td>0.005</td>
</tr>
<tr>
<td>ttp</td>
<td>definite time delay for transient boost pickup</td>
<td>1.00</td>
</tr>
<tr>
<td>tpw</td>
<td>washout time constant for gate boost</td>
<td>30.00</td>
</tr>
<tr>
<td>vtp</td>
<td>velocity limit on feedforward signal</td>
<td>0.002</td>
</tr>
<tr>
<td>kfp</td>
<td>feedforward gain for speed-load reference</td>
<td>0.00</td>
</tr>
<tr>
<td>tff</td>
<td>feedforward filter time constant</td>
<td>1.00</td>
</tr>
<tr>
<td>fbus</td>
<td>bus number of 'family' model (hyg6d)</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

a) Per unit parameters are on base of turbine MW capability. The base power for the must be stated by means of the "mwcap=" entry in this record. The turbine base power entered in the associated h6bd data record is NOT USED. If no value is entered for "mwcap", the generator MVA base is used.

b) The turbine and the feedback functions of the governor are described by the parameters entered and maintained by the h6bd model. See the h6bd model data page for notes on the turbine and basic governor modeling.

c) Two separate feedforward paths can be used to maneuver the turbine power output by addition of a signal to the gate position command of the governor.

Changes in the speed-load reference, genbc[].pref, are fed forward through the gain, Kfp. This feedforward gives a quick open-loop response to changes in the speed-load reference of the governor.

The externally provided signal, Ptp, is fed forward to provide an open-loop adjustment of gate opening upon command. The application of this feedforward signal is initiated when frequency measured at the generator terminals deviates from 1.0 per unit by more than <ftp> per unit for longer than <ttp> seconds.

d) The load controller model, lcfb1, should have its first parameter, type, set to zero when it is applied to h6b.

Output Channels:

<table>
<thead>
<tr>
<th>Record Level</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>gate</td>
<td>Turbine gate position, pu</td>
</tr>
<tr>
<td>1</td>
<td>pm</td>
<td>Turbine power, MW</td>
</tr>
<tr>
<td>1</td>
<td>bld</td>
<td>Turbine blade position, pu</td>
</tr>
</tbody>
</table>
Figure 1  Governor Schematic Diagram

Figure 2  Turbine Schematic Diagram
Model Name: h6bd

Description: h6bd model to manage parameter data for the h6b hydro turbine governor model

Inputs: n/a

Parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>re</td>
<td>Permanent droop for electrical power feedback, pu</td>
<td>0.0</td>
</tr>
<tr>
<td>rg</td>
<td>Permanent droop for gate position feedback, pu</td>
<td>0.05</td>
</tr>
<tr>
<td>tpe</td>
<td>Electric power feedback transducer time const, sec</td>
<td>0.025</td>
</tr>
<tr>
<td>tsp</td>
<td>Shaft speed transducer time const, sec</td>
<td>-0.50</td>
</tr>
<tr>
<td>fd</td>
<td>Flag for proportional elec power signal</td>
<td>0.00</td>
</tr>
<tr>
<td>kp</td>
<td>Governor proportional gain</td>
<td>***</td>
</tr>
<tr>
<td>ki</td>
<td>Governor integral gain</td>
<td>***</td>
</tr>
<tr>
<td>kd</td>
<td>Governor derivative gain</td>
<td>***</td>
</tr>
<tr>
<td>td</td>
<td>Derivative gain filter time constant, sec</td>
<td>0.05</td>
</tr>
<tr>
<td>kg</td>
<td>Pilot servovalve gain</td>
<td>5.00</td>
</tr>
<tr>
<td>tsp</td>
<td>Shaft speed transducer time const</td>
<td>-0.50</td>
</tr>
<tr>
<td>vmin</td>
<td>Minimum gate actuator velocity, pu/sec</td>
<td>0.20</td>
</tr>
<tr>
<td>gmax</td>
<td>Maximum gate actuator stroke, pu</td>
<td>1.00</td>
</tr>
<tr>
<td>gmin</td>
<td>Minimum gate actuator stroke, pu</td>
<td>0.00</td>
</tr>
<tr>
<td>dturb</td>
<td>Turbine speed sensitivity constant</td>
<td>0.50</td>
</tr>
<tr>
<td>pnl</td>
<td>Hydraulic power needed at full speed no load</td>
<td>0.12</td>
</tr>
<tr>
<td>blg</td>
<td>Gate linkage backlash, pu</td>
<td>0.00</td>
</tr>
<tr>
<td>tbf</td>
<td>Kaplan turbine blade angle filter time const, pu</td>
<td>2.50</td>
</tr>
<tr>
<td>tbs</td>
<td>Kaplan turbine blade angle servo time const, pu</td>
<td>0.50</td>
</tr>
<tr>
<td>dbbl</td>
<td>Blade intentional deadband, pu</td>
<td>0.0025</td>
</tr>
<tr>
<td>hdam</td>
<td>Operating head, pu</td>
<td>1.00</td>
</tr>
<tr>
<td>gv0</td>
<td>First abscissa value for wicket gate</td>
<td>0.00</td>
</tr>
<tr>
<td>gv1</td>
<td>and blade curves</td>
<td>0.40</td>
</tr>
<tr>
<td>gv2</td>
<td></td>
<td>0.50</td>
</tr>
<tr>
<td>gv3</td>
<td></td>
<td>0.55</td>
</tr>
<tr>
<td>gv4</td>
<td></td>
<td>0.60</td>
</tr>
<tr>
<td>gv5</td>
<td></td>
<td>0.70</td>
</tr>
<tr>
<td>gv6</td>
<td></td>
<td>0.80</td>
</tr>
<tr>
<td>gv7</td>
<td></td>
<td>0.85</td>
</tr>
<tr>
<td>gv8</td>
<td></td>
<td>0.88</td>
</tr>
<tr>
<td>gv9</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>agv0</td>
<td>Curve of wicket gate flow area</td>
<td>0.00</td>
</tr>
<tr>
<td>agv1</td>
<td>as function of gate servo position</td>
<td>0.50</td>
</tr>
<tr>
<td>agv2</td>
<td></td>
<td>0.62</td>
</tr>
<tr>
<td>agv3</td>
<td></td>
<td>0.68</td>
</tr>
<tr>
<td>agv4</td>
<td></td>
<td>0.73</td>
</tr>
<tr>
<td>agv5</td>
<td></td>
<td>0.83</td>
</tr>
<tr>
<td>agv6</td>
<td></td>
<td>0.91</td>
</tr>
<tr>
<td>agv7</td>
<td></td>
<td>0.94</td>
</tr>
<tr>
<td>agv8</td>
<td></td>
<td>0.955</td>
</tr>
<tr>
<td>agv9</td>
<td></td>
<td>1.00</td>
</tr>
<tr>
<td>bgv0</td>
<td>Curve of Kaplan turbine blade flow area</td>
<td>0.78</td>
</tr>
<tr>
<td>bgv1</td>
<td>as function of gate servo position</td>
<td>0.78</td>
</tr>
<tr>
<td>bgv2</td>
<td></td>
<td>0.78</td>
</tr>
<tr>
<td>bgv3</td>
<td></td>
<td>0.79</td>
</tr>
<tr>
<td>bgv4</td>
<td></td>
<td>0.81</td>
</tr>
<tr>
<td>bgv5</td>
<td></td>
<td>0.88</td>
</tr>
<tr>
<td>bgv6</td>
<td></td>
<td>0.96</td>
</tr>
</tbody>
</table>
Notes:

a) Per unit parameters are on the base of turbine MW capability which is entered on the data record of the h6b model, or models, that are associated with this model. The "mwcap" entry may be included in the data record of this data model for convenience but it is NOT USED by the dynamic models associated with this data model. Note that the data established by an h6bd model can be used to represent multiple turbine-governor sets of differing real power capabilities.

b) The gates travel over a range of 1.0 per unit from fully closed to fully opened. The gates are at a position greater than zero when the turbine real power output is zero. Gmax and Gmin are operating limits.

c) Tpe, Tsp, Td, Tg, Tbf, Tbs must be greater than zero.

d) Dturb has the dimensions power/speed.

e) Details of the deadband and backlash functions are shown in figure 6.

f) The turbine is described by two curves as follows:

   agv  The effective flow area of wicket gates versus gate servomotor stroke.
   bgv  The effective flow area factor of turbine versus gate servomotor command.

The effective flow area of complete turbine is the product of the two factors:

   (effective flow area) = agv(g) * (bgvmin + (1-bgvmin)*bgv(g)

where g is gate servo stroke

Each curve is specified by ten points (numbered 0-9) corresponding to ten gate servo positions which are given as gv0 - gv9. The same set of servo stroke positions is used for both curves.

The curve of gate flow area, agv, must pass thru (0,0) and (1,1) for all turbines

For a Kaplan turbine the blade flow area curve, bgv, must pass through (0,0) and (1,1).

For a Francis turbine all the points on the blade flow area curve, bgv, curve must be at be (g,1).

Figure 7 shows typical curves for a Kaplan turbine. Figure 8 shows typical curves for a Francis turbine.

The parameter bgvmin specifies the blade flow area when the blade servo is at zero stroke. The value of bgvmin is in per unit of the blade flow area when the blades are fully open.

g) The gate opening and flow required to support no-load operation is described by the parameter, pnl.

h) The adjustment factor, deff, can be used to account for reduction of Kaplan turbine power when the blade angle is at an off-nominal angle. The adjustment is

   (adjusted pwr) = (on-angle pwr)-deff*(ideal blade stroke - actual blade stroke)

Output Channels:

None
Figure 1  Governor Schematic Diagram

Figure 2  Transfer function for calculation of turbine power
Figure 3  Turbine wicket gate flow curve

Figure 4  Kaplan turbine blade flow curve
Figure 5  Off-angle power adjustment

Figure 6  Deadband details
### Figure 7: Spreadsheet for preparation of Kaplan turbine data

<table>
<thead>
<tr>
<th>g</th>
<th>ag</th>
<th>ab</th>
<th>Power Calc</th>
<th>Power Test</th>
<th>Pn1</th>
<th>bgvmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.149</td>
<td>-0.200</td>
<td>0.12</td>
<td>0.78</td>
</tr>
<tr>
<td>0.400</td>
<td>0.500</td>
<td>1.000</td>
<td>0.334</td>
<td>0.220</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.500</td>
<td>0.620</td>
<td>0.000</td>
<td>0.450</td>
<td>0.450</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.550</td>
<td>0.680</td>
<td>0.045</td>
<td>0.517</td>
<td>0.500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.600</td>
<td>0.730</td>
<td>0.136</td>
<td>0.584</td>
<td>0.580</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.700</td>
<td>0.830</td>
<td>0.460</td>
<td>0.757</td>
<td>0.750</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.800</td>
<td>0.910</td>
<td>0.820</td>
<td>0.934</td>
<td>0.920</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.850</td>
<td>0.940</td>
<td>0.980</td>
<td>1.011</td>
<td>1.010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.880</td>
<td>0.960</td>
<td>1.000</td>
<td>1.040</td>
<td>1.040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.090</td>
<td>1.090</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure 8: Spreadsheet for preparation of Francis turbine data

<table>
<thead>
<tr>
<th>g</th>
<th>ag</th>
<th>ab</th>
<th>Power Calc</th>
<th>Power Test</th>
<th>Pn1</th>
<th>bgvmin</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>1.000</td>
<td>-0.149</td>
<td>0.000</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>0.400</td>
<td>0.500</td>
<td>1.000</td>
<td>0.471</td>
<td>0.471</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.500</td>
<td>0.620</td>
<td>1.000</td>
<td>0.619</td>
<td>0.619</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.550</td>
<td>0.680</td>
<td>1.000</td>
<td>0.694</td>
<td>0.694</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.600</td>
<td>0.730</td>
<td>1.000</td>
<td>0.756</td>
<td>0.756</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.700</td>
<td>0.830</td>
<td>1.000</td>
<td>0.879</td>
<td>0.879</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.800</td>
<td>0.910</td>
<td>1.000</td>
<td>0.979</td>
<td>0.979</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.850</td>
<td>0.940</td>
<td>1.000</td>
<td>1.016</td>
<td>1.016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.880</td>
<td>0.960</td>
<td>1.000</td>
<td>1.040</td>
<td>1.040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.090</td>
<td>1.090</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <string.h>

#include < files needed by your main program >

float cmag();
float interp5();

struct h6b
{
    // constants - values to be picked up from dynamics data input process
    float tw;    // water inertia time constant
    float ptp;    // user-defined gate boost input
    float ftp;    // frequency deviation for transient boost to pick up
    float ttp;    // timer for transient boost pickup
    float twp;    // washout time constant for gate boost
    float vtp;    // velocity limit on feedforward signal
    float kfp;    // feedforward gain for speed-load reference
    float tff;    // feedforward filter time constant
    float fbus;    // bus number of 'family' model (h6bd)

    // state variables
    float s[11];
    float ds[11];

    // algebraic variables
    float gv;    // gate servomotor stroke
    float ab;    // blade angle
    float gout;    // governor gate command
    float h;
    float ginit;
    float prefinit;
    float tsptp;
    int tflag;
    float ag;    // gate area after backlash
    float bc;    // blade stroke command
};

struct h6bd
{
    // constants - values to be picked up from dynamics data input process
    float re;
    float rg;
    float tpe;
    float tsp;
    float fd;
    float tp;
    float kp;
    float k1;
    float k2;
    float td;
    float kg;
    float tg;
    float velm;
    float gmax;
    float gmin;
    float dturb;
    float pnl;
    float blgate;
    float tbf;
    float dbbld;
    float blbld;
    float hdam;
    float gv0;
    float gv1;
    float gv2;
    float gv3;
    float gv4;
    float gv5;
    float gv6;
    float gv7;
    float gv8;
    float gv9;
    float agv0;    // ordinate values 0-9 of gate flow area curve
};
float agv4;
float agv5;
float agv6;
float agv7;
float agv8;
float agv9;
float bgv0;  // ordinate values 0-9 of blade flow area curves
float bgv1;
float bgv2;
float bgv3;
float bgv4;
float bgv5;
float bgv6;
float bgv7;
float bgv8;
float bgv9;
float bgvmin;  // base value of blade flow area
float deff;    // off-cam turbine power fall-off factor

int h6b_alge()  
{
  float agf, ab, af, pturb;
  if( (h6b.s[5] - h6bd.blgate) >= h6b.gv) h6b.gv = h6b.s[5] - h6bd.blgate;  // gate backlash
  if( (h6b.s[5] + h6bd.blgate) <= h6b.gv) h6b.gv = h6b.s[5] + h6bd.blgate;
  h6b.ag = interp5( h6b.gv, &h6bd.gv0, &h6bd.agv0 );  // gate area as function of gate stroke
  af = h6b.s[7] * h6b.ag;
  h6b.ab = h6b.s[7];
  if( af > .01 ) h6b.h = (h6b.s[8] / af) * (h6b.s[8] / af);
  else h6b.h = h6bd.hdam;
  pturb = (h6b.s[8] * h6b.h) - ((<speed> - 1.0) * h6bd.dturb * h6b.gv);
  pturb -= h6bd.deff*(h6b.bc - h6b.ab)*(h6b.bc - h6b.ab);
  pturb -= h6bd.pnl;
  pturb /= (1.-h6bd.pnl);
  // pturb is turbine power in per unit on turbine power base
  // convert pturb onto the appropriate base for use in the main simulation program
  <pmech> = <converted value>
  return(0);
}

int h6b_rate()  
{
  float a, temp, gmax, gmin;
  float error, pb, gp, gc;
  <pgen> = <get generator electrical power from network solution and convert to per unit on turbine power base>
  if( h6bd.tpe > 0. ) h6b.ds[0] = (<pgen> - h6b.s[0]) / h6bd.tpe;
  else h6b.ds[0] = h6b.s[0] = 0.;
  if( h6bd.tsp < 0. ) h6b.ds[1] = (<frequency> + 1. - h6b.s[1]) / fabs(h6bd.tsp);
  else
    if( h6bd.tsp > 0. ) h6b.ds[1] = (<speed> - h6b.s[1]) / h6bd.tsp;
    else
      h6b.s[1] = <speed>;
    if( h6bd.re > 0. ) error = <pref> - h6b.s[0] * h6bd.re - h6b.s[1];
    if( h6bd.rg > 0. ) error = <pref> - h6b.gout * h6bd.rg - h6b.s[1];
    if( h6bd.fd ) gp = h6bd.kp * (1. - h6b.s[1]);
    else gp = h6bd.kp * error;
    h6b.ds[2] = h6bd.ki * error;

if ( h6bd.fd )  a = error;
else            a = h6b.s[1] - 1.;
if ( h6bd.fd )  a = error;
else            a = h6b.s[1] - 1.;


if ( h6b.gout > h6bd.gmax ) h6b.gout = h6bd.gmax;
if ( h6b.gout < h6bd.gmin ) h6b.gout = h6bd.gmin;

pb = 0.;
switch( h6b.tflag )
{
  case 0: if ( <speed> < (1.-h6b.ftp) ) { h6b.tflag = 1; h6b.tsptp = dypar.time + h6b.ttp; } break;
  case 1: if ( <speed> > (1.-h6b.ftp) ) { h6b.tflag = 0; h6b.tsptp = 99999.; }
           else if ( dypar.time >= h6b.tsptp )  { h6b.tflag = 2; pb = h6b.ptp; }
  case 2: if ( <speed> < (1.-h6b.ftp) )  pb = h6b.ptp;
           else
    {
      if ( dypar.time > (h6b.tsptp + 3.*h6b.twp) )
        { h6b.tsptp = 99999.; h6b.tflag = 0; h6b.s[9] = h6b.ds[9] = 0.; }
      else
        pb = h6b.ptp;
    }
  if ( h6b.twp && h6b.ptp )
    h6b.ds[9] = ( pb - h6b.s[9] ) / h6b.twp;
else
    h6b.ds[9] = h6b.s[9] = 0.;

pb -= h6b.s[9];

h6b.ds[10] = ( pb - h6b.s[10] ) / h6b.tff;
if ( h6b.ds[10] > h6b.vtp ) h6b.ds[10] = h6b.vtp;
if ( h6b.ds[10] < -h6b.vtp ) h6b.ds[10] = -h6b.vtp;

h6b.gout += h6b.s[10];
if ( h6b.gout > h6bd.gmax ) h6b.gout = h6bd.gmax;
if ( h6b.gout < h6bd.gmin ) h6b.gout = h6bd.gmin;

switch ( gens[kgen].baseload_flag )
{
  case 1:  gmax = h6b.ginit;
            gmin = h6bd.gmin;
            break;
  case 2 : gmax = gmin = h6b.ginit;
            break;
  default: gmax = h6bd.gmax;
            gmin = h6bd.gmin;
            break;
}

if ( h6b.s[4] > h6bd.velm ) h6b.s[4] = h6bd.velm;

h6b.s[5] = h6bd.velm;

h6b.ds[5] = h6b.s[5];

if ( h6bd.tbf > 0. )
{
  error = h6b.gout - h6b.s[6];
a = 0.;
if ( error > h6bd.dbbl )
  a = error - h6bd.dbbl;
else
  if ( error < -h6bd.dbbl )
    a = error + h6bd.dbbl;
h6b.ds[6] = a / h6bd.tbf;
else{
  h6b.s[6] = h6b.gout; h6b.ds[6] = 0.;
}

h6b.bc = h6bd.bgmin + (1.- h6bd.bgmin) * interp5( h6b.s[6], &h6bd.gv0, &h6bd.bgv0 );
if( h6bd.tbs > 0. )
{
  error = h6b.bc - h6b.s[7];
  a = 0.;
  if ( error > h6bd.blbl )
    a = error - h6bd.blbl;
  else
    if ( error < -h6bd.blbl )
      a = error + h6bd.blbl;
  h6b.ds[7] = a / h6bd.tbs;
}
else
  h6b.s[7] = h6b.bc; h6b.ds[7] = 0.;

h6b.ds[8] = ( h6bd.hdam - h6b.h ) / h6bd.tw;
return(0);

int h6b_init( )
{
  int i, l, j, m;
  float se, vt, vmx, vmi, temp, ps, pturb;
  float fa[10], ee, eff, flow, af, oldflow, g;
  float az, bz, cz, dz, aa, bb, cc, g1, g2;
  int kk;

  // Get initial generator electrical power from initial condition load flow solution
  // and convert to turbine initial power in per unit on turbine power base
  pturb = <get initial turbine power in per unit of turbine base power>

  // Turbine initialization
  // Build curve of overall flow area vs gate servo stroke
  // Note use of C-language indirect addressing convention (& denotes pointer)
  for ( i = 0; i < 10; ++i )
  {
    fa[i] = h6bd.bgmin + (1.-h6bd.bgmin) * (*(&h6bd.gv0+i));
    fa[i] *= *(&h6bd.agv0+i);
  }

  flow = (pturb*(1.-h6bd.pnl) + h6bd.pnl)/h6bd.hdam;
  af = flow / sqrt(h6bd.hdam);
  g = interp5( af, fa, &h6bd.gv0 );

  h6b.s[4] = 0.;
  h6b.s[7] = h6b.ab - h6b.bc = h6bd.bgmin + (1.-h6bd.bgmin) * interp5( g, &h6bd.gv0, &h6bd.bgv0 );
  h6b.s[8] = flow;

if( h6b.s[5] > 1. )
{
  <warn - greater than 1.0 gate opening>
}

if ( h6b.s[5] > h6b.gmax)
{
  <warn - greater than gmax gate opening>
}

if ( h6bd.tbs ) h6b.s[7] = h6bd.bgmin + (1.-h6bd.bgmin) * interp5( h6b.s[6], &h6bd.gv0, &h6bd.bgv0 );
else
  h6b.s[7] = 1.;

h6b.ginit = h6b.gv = h6b.gout = h6b.s[2] = h6b.s[5];

h6b.ag = interp5( h6b.gv , &h6bd.gv0, &h6bd.agv0 );

h6b.s[9] = 0.;

h6b.s[10] = 0.;


h6b.s[3] = 0.;

h6b.s[0] = <pgen> / <convert generator power to per unit on turbine power base>

if ( h6bd.re > 0. )<pref> = 1. + h6b.s[0] * h6bd.re;
if ( h6bd.rg > 0. )<pref> = 1. + h6b.s[5] * h6bd.rg;

h6b.prefinit = <pref>;
if ( h6bd.fd )    h6b.s[2] = h6b.gout;
else             h6b.s[2] = h6b.gout - h6bd.kp * (genbc[k].pref - 1.);

h6b.h = h6bd.hdam;

h6b.tflag = 0;
h6b.tsptp = 99999.;
}

float interp5( s, x, y )
float s;
float *x, *y;
{
    float a;
    int i;
    if ( s >= 1. ) return(1.);  
    if ( s <= 0. ) return(y[0]);
    for ( i = 0; i<9; ++i )
        if ( s <= x[i+1] ) { a = y[i] + (s-x[i])*(y[i+1]-y[i])/(x[i+1]-x[i]); return(a); }
    return( -1. );