

**Integration of Process & Control in Blue-Print models**

The only way to get the Blue-Print model of true plant performance leads through integrating the operation of the process and control subsystems

Nor-Par\* has the necessary experience and skill to integrate process and control models and has done such integrations. To full and realistically model the control subsystem, Nor-Par can write relevant part of the control code and integrate it with the Blue-Print model. The specialty control code is written by Nor-Par based on unique equations from the client to reflect the unique Control System of the client. This includes Advanced Process Control (APC).

Process & Control integration give you the following benefits:

- Since the real world is dynamic, the Blue-Print model gives the closest representation of the operation of a producing plant available with state-of-art simulation tools. It is “no compromise” way of modelling, taking into account the size of the equipment and the performance of the control system
- Possibility to analyse detailed performance of the plant on technical level including the in-

teraction with the Control System. This detail of information is necessary for real optimisation of the process to improve the process economics

- Guidelines to improve the performance of the control loops (valve capacity, controllable area, sensitivity, interaction)

\*) In project-cooperation with PID AB, the leading Swedish process automation company

```
// WORK WITH CONTROL ALGORITHM
// P-Term
P_k=K*((b-1)*spd_k+/-ed_k). +/- sign!
P_k = K*(b-1)*spd_k;
if(sign==0)
{
  P_k = P_k-K*ed_k;
}
else
{
  P_k = P_k+K*ed_k;
}

// D-Term
D_k=Dc*(D_kal-K*nfd*(pvd_k-pvd_kal))
Dc=Td/(Td+nfd*dt);
Dc = Td/(Td+nfd*dt);
D_kal = uspec(43);
D_k = Dc*(D_kal-K*nfd*(pvd_k-pvd_kal));

// I-term
I_k=I_kpl calculated at k-1
I_k = uspec(21);

// Control output
vd_k = P_k+I_k+D_k;
ud_k = vd_k; // Calculated internal signal
if(ud_k>cinmax) ud_k = cinmax; // Realized internal signal
if(ud_k<cinmin) ud_k = cinmin;

// Transformation into %
upcd_k = 100 / (cinmax-cinmin)*(ud_k-cinmin);

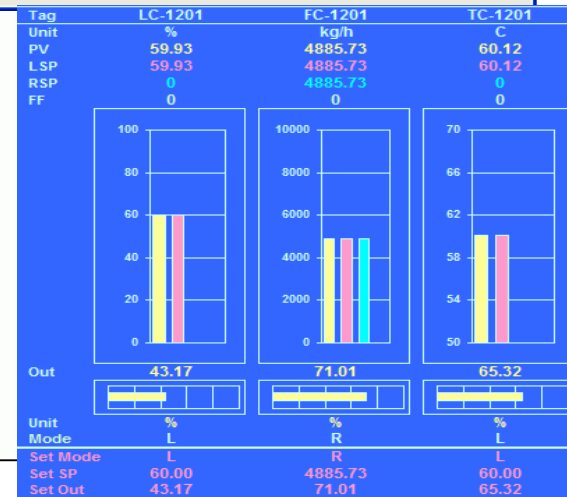
// Integral for the next step
I_kpl=I_k+/-K*dt/Ti*ed_k+dt/Ti*(ud_k-vd_k). +/- sign!
I_kpl = I_k;
if(Ti>1.E-6) I_kpl = I_kpl+dt/Ti*(ud_k-vd_k);
if(Ti>1.E-6)
{
  if(sign==0)
  {
    I_kpl = I_kpl-K*dt/Ti*ed_k;
  }
  else
  {
    I_kpl = I_kpl+K*dt/Ti*ed_k;
  }
}
}
```

Small portion of a specialized controller algorithm in C++ language

**Custom screen for a specialised controller**

The screenshot shows a control configuration window with the following fields and options:

- Control Mode: Remote
- Set point: 4881.74
- Controller output, %: 70.9843
- SS output (internal): 15.5982
- Proportional gain (K): 1.25
- Integral time (Ti): 2.5 min
- Derivative time (Td): min
- Control valve ID: 10 or Slave ID: Master ID: T3
- Simulation mode: Dynamic
- Conv. Function: Linear
- Variable Min: 1000
- Variable Max: 10000
- Internal sign. Min: 4
- Internal sign. Max: 20
- Error Definition:
  - Error = X - Xset (C,P,L)
  - Error = Xset - X (H,F)
- Measured Object:
  - Stream ID number: 9 Variable: 6 Total mass rate
  - Equipment
  - Component: <None>
  - Variable unit: 1 Mole/Mass
  - Measured variable no.: 6



Intelligent Control Interface to the Integrated System