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

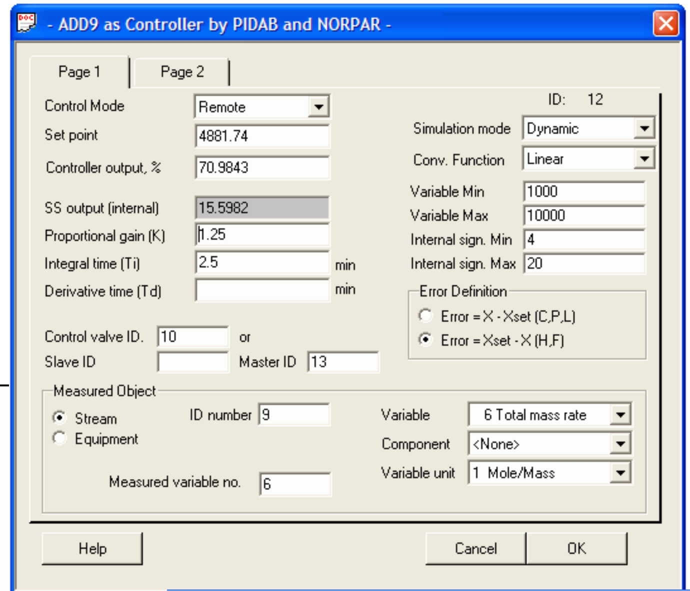
Nor-Par* has the necessary experience and skill to integrate process and control models and has done such integrations. To full and realistically model the control subsystem, Nor-Par can write relevant part of the control code and integrate it with the Blue-Print model. The specialty control code is 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 interaction 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)

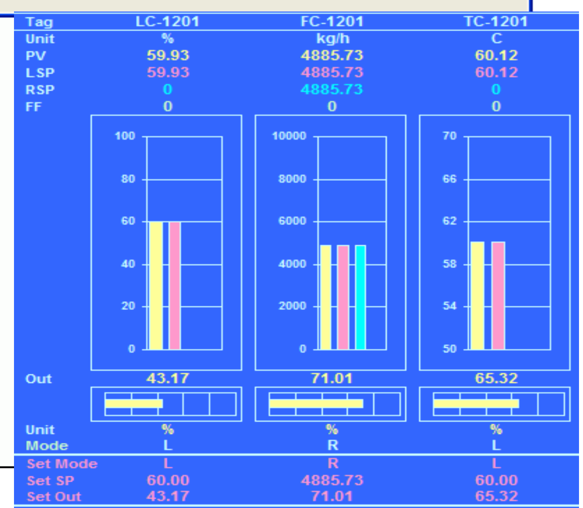
*) In project-cooperation with PID AB, the leading Swedish process automation company

Custom screen for a specialised controller




```
// 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_km1-K*nfd*(pvd_k-pvd_km1))
// Dc=Td/(Td+nfd*dt)
Dc = Td/(Td+nfd*dt);
D_km1 = uspec[43];
D_k = Dc*(D_km1-K*nfd*(pvd_k-pvd_km1));

// I-term
// I_k=I_kp1 calculated at k-1
I_k = uspec[21];

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

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

// Integral for the next step
// I_kp1=I_k+/-K*dt/Ti*ed_k+dt/Tt*(ud_k-vd_k), +/- sign!
I_kp1 = I_k;
if(Tt>1.E-6) I_kp1 = I_kp1+dt/Tt*(ud_k-vd_k);
if(Ti>1.E-6)
{
    if(sign==0)
    {
        I_kp1 = I_kp1-K*dt/Ti*ed_k;
    }
    else
    {
        I_kp1 = I_kp1+K*dt/Ti*ed_k;
    }
```

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



Intelligent Control Interface to the Integrated System