Multivariable control

I. Single-loop control (decentralized)
II. Decoupling (similar to feedforward)
III. Model predictive control (MPC)

I. Multivariable control using single loops

- Pairing rules
- Interactions
- Choice of pairings (RGA)

- MVs: u $= (q_h\ q_c)$
- CVs: y = (T q)

What pairing?

$$G = \begin{bmatrix} 20 & -10 \\ 1 & 1 \end{bmatrix}$$

Problem 3: Water mixer

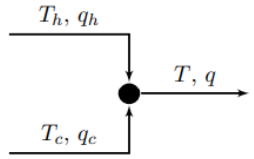


Figure 1: Mixer system

Consider the process of mixing hot and cold water, as shown in Figure 1. The process has inputs $u_1 = \Delta q_h$ [ℓ/s], $u_2 = \Delta q_c$ [ℓ/s], and outputs $y_1 = \Delta T$ [°C], $y_2 = \Delta q$ [ℓ/s].

The control objective is to have a mixing temperature $T = 40$°C and a total flow leaving the mixer of $q = 1$ ℓ/s. At the nominal operating point we have $T_c = 30$°C and $T_h = 60$°C.

1. Formulate the energy and mass balances. The dynamics of this process are very fast; so, a steady-state model is sufficient to get T and q.
2. Linearize the model and show that the linear model can be written $y = Gu$, where:

$$G = \begin{bmatrix} k_1 & k_2 \\ 1 & 1 \end{bmatrix}$$

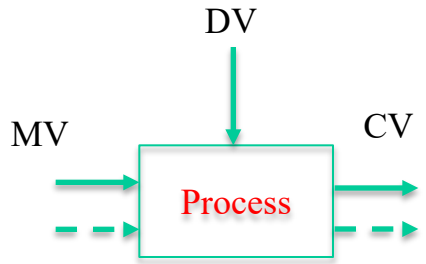
with: $k_1 = (T_h^* - T^*)/q^*$ $k_2 = (T_c^* - T^*)/q^*$ $u = [u_1 \quad u_2]^T$

The symbol * denotes the steady state value.

3. What are the steady state values for q_c and q_h?
4. Find the gain matrix G at the nominal operating point.
5. Based on G, which stream (q_h or q_c) would you use to control the temperature (T)? Explain briefly.

In Exercise 10, you will find out if your intuition was right.

Which input (MV) to pair with a given output (CV)?
Two main pairing rules (based on process insight):



1. ***"Pair-close rule"***
   - *The MV should have a large, fast, and direct effect on the CV*
   - *Sometimes not obvious:*
     - *RGA may help for selecting pairing*
     - *or use multivariable control*
2. ***"Input saturation rule"***
   - *Pair a MV that may saturate with a CV that can be given up (when the MV saturates)*

Shower example, MV=(q_h q_c), CV= (T q)

1. ***"Pair-close rule:*** *The MV should have a large, fast, and direct effect on the CV"*
   - *Shower, Rule 1:*
     - *Use largest flow (of qc and qh) for flow control*
     - *and smallest flow for temperature control*

2. ***"Input saturation rule:*** *Pair a MV that may saturate with a CV that can be given up (when the MV saturates)"*
   - *Shower: Assume hot water (qh) may saturate (at fully open) and that flow control can be given up (it's most important is to control temperature). Rule 2 gives:*
     - *Use hot water for flow control*
       - *Give up flow control if hot water saturates at max*
     - *Use cold water (qh) for temperature control*

But shower is very coupled, so maybe we need decoupling

Multivariable process

Distillation column

"Increasing reflux L from 1.0 to 1.1 changes y_D from 0.95 to 0.97, and x_B from 0.02 to 0.03"

"Increasing boilup V from 1.5 to 1.6 changes y_D from 0.95 to 0.94, and x_B from 0.02 to 0.01"

Steady-State Gain Matrix

$$\begin{pmatrix} \Delta Y_D \\ \Delta x_B \end{pmatrix} = G(0) \begin{pmatrix} \Delta L \\ \Delta V \end{pmatrix}$$

$$G(0) = \begin{bmatrix} g_{11} & g_{12}(0) \\ g_{21} & g_{22}(0) \end{bmatrix} = \begin{bmatrix} \frac{0.97-0.95}{1.1-1.0} & \frac{0.94-0.95}{1.6-1.5} \\ \frac{0.03-0.02}{1.1-1.0} & \frac{0.01-0.02}{1.6-1.5} \end{bmatrix} = \begin{bmatrix} 0.2 & -0.1 \\ 0.1 & -0.1 \end{bmatrix}$$

$\text{Effect of input } 1(\Delta L) \text{ on output } 2(\Delta x_B)$

Can also include dynamics :

$$G_{(s)} = \begin{bmatrix} \frac{0.2}{1+50s} & -\frac{0.1}{1+50s} \\ \frac{0.1}{1+40s} & -\frac{0.1}{1+40s} \end{bmatrix} \begin{matrix} \rightarrow \Delta y_D \\ \rightarrow \Delta x_B \end{matrix}$$

(Time constant 50 min for y_D)

(time constant 40 min for x_B)

Analysis of Multivariable processes

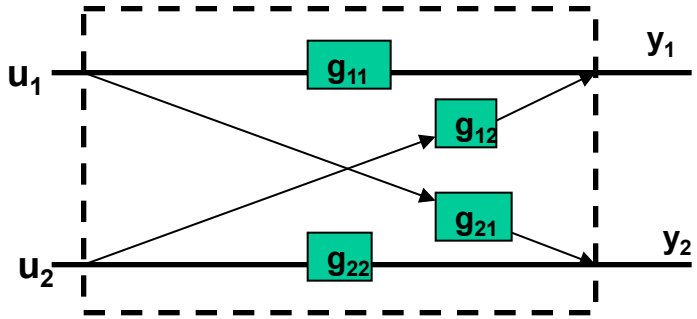
Process Model 2x2

$$\text{"}Open - loop\text{"}$$

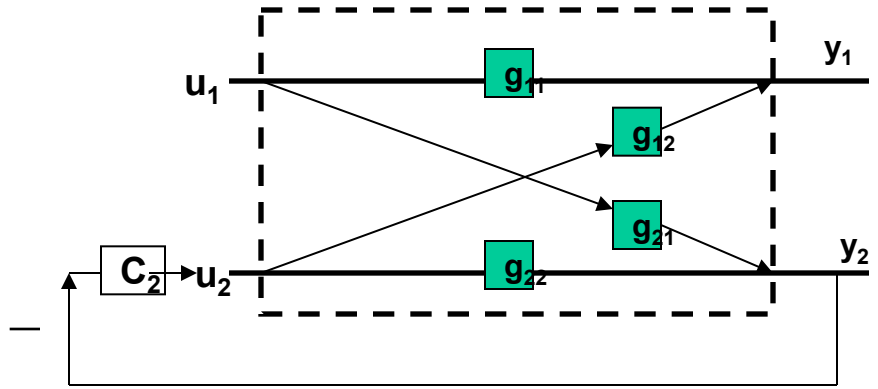
$$y_1(s) = g_{11}(s)u_1(s) + g_{12}(s)u_2(s) \qquad (1)$$

$$y_2(s) = g_{21}(s)u_1(s) + g_{22}(s)u_2(s) \qquad (2)$$

INTERACTIONS: Caused by nonzero offdiagonal elements (g_{12} and/or g_{21})



RGA: Consider effect of u_1 on y_1



1) **"Open-loop" ($C_2 = 0$):** $y_1 = g_{11}(s) \cdot u_1$

2) **"Closed-loop" (close loop 2, $C_2 \neq 0$):** $y_1 = \left(g_{11}(s) - \frac{g_{12} g_{21} \cdot C_2}{1 + g_{22} \cdot C_2} \right) u_1$

Change caused by "interactions"

Derivation.

Close loop 2: $u_2 = -c_2(y_2 - y_{2s})$

Here: $y_2 = g_{21}u_1 + g_{22}u_2$ and assume $y_{2s} = 0$:

$\Rightarrow u_2 = -c_2(g_{21}u_1 + g_{22}u_2) \Rightarrow u_2 = \frac{-c_2 g_{12}}{1+g_{22}c_2} u_1$

Effect of u_1 on y_1 with loop 2 closed is then:

$$y_1 = g_{11}u_1 + g_{12}u_2 = \underbrace{g_{11}\left(1 - \frac{g_{12}g_{21}c_2}{1+g_{22}c_2}\right)}_{\widehat{g}_{11}} u_1$$

Limiting Case $C_2 \rightarrow \infty$ (perfect control of y_2) *

$$y_1 = \left(g_{11}(s) - \frac{g_{12}\, g_{21}}{g_{22}} \right) u_1 = g_{11}(1/\lambda_{11}) \cdot u_1$$

How much has "gain" from u_1 to y_1 changed by closing loop 2 with perfect control?

$$\text{Relative Gain} = \frac{(y_1/u_1)_{OL}}{(y_1/u_1)_{CL}} = \frac{g_{11}}{g_{11} - \frac{g_{12}\, g_{21}}{g_{22}}} = \frac{1}{1 - \frac{g_{12}\, g_{21}}{g_{11}\, g_{22}}} \overset{\text{def}}{=} \lambda_{11}^{RGA}$$

The relative Gain Array (RGA) is the matrix formed by considering all the relative gains

$$RGA = \Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \frac{(y_1/u_1)_{OL}}{(y_1/u_1)_{CL}} & \frac{(y_1/u_2)_{OL}}{(y_1/u_2)_{CL}} \\ \frac{(y_2/u_1)_{OL}}{(y_2/u_1)_{CL}} & \frac{(y_2/u_2)_{OL}}{(y_2/u_2)_{CL}} \end{bmatrix}$$

* Alternative derivation (on board) : Can derive by setting y2=0 in (2) and put resulting u2 into (1). See lecture

Example from before

$$G = \begin{bmatrix} 0.2 & -0.1 \\ 0.1 & -0.1 \end{bmatrix}, \quad \lambda_{11} = \frac{1}{1 - \underbrace{\frac{0.1 \;\; 0.1}{0.2 \;\; 0.1}}_{0.5}} = 2$$

$$RGA = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

Only acceptable pairings :

$u_1 \leftrightarrow y_1$

$u_2 \leftrightarrow y_2$

Not recommended :

$u_1 \leftrightarrow y_2$

$u_2 \leftrightarrow y_1$

With integral action :

Negative RGA $\Rightarrow$ individual loop unstable OR overall system unstable when individual loops saturates

Use of RGA:

(1) Interactions

- RGA-element (λ)> 1: Smaller gain by closing other loops ("fighting loops" gives slower control)
- RGA-element (λ) <1: Larger gain by closing other loops (can get instability when other loop is closed)
- RGA-element (λ) negative: Gain reversal by closing other loops (Oops!)

Rule 1. Avoid pairing on negative steady-state relative gain – otherwise you get instability if one of the loops become inactive (e.g. because of saturation)

Rule 2. Choose pairings corresponding to RGA-elements close to 1

Traditional: Consider Steady-state

Better (improved Rule 2): Consider frequency corresponding to closed-loop time constant

Example 3x3 process: 6 possible pairing options

$$G = \begin{pmatrix} 16.8 & 30.5 & 4.30 \\ -16.7 & 31.0 & -1.41 \\ 1.27 & 54.1 & 5.40 \end{pmatrix}, \quad RGA(G) = \begin{pmatrix} 1.50 & 0.99 & -1.48 \\ -0.41 & 0.97 & 0.45 \\ -0.08 & -0.95 & 2.03 \end{pmatrix}$$

Only diagonal pairings give positive steady-state RGA's!

Property of RGA:

- **Columns and rows always sum to 1**
- **RGA independent of scaling (units) for u and y.**

RGA for general case:

$$[RGA]_{ij} = (g_{ij})_{OL}/(g_{ij})_{CL} = [G]_{ij}[G^{-1}]_{ji}$$

= element-by-element multiplication of G and ${G^{-1}}^T$.

Matlab: `RGA = G.*pinv(G).'`

Example

```
G = [5 10 1; 20 -10 0; 18 0 2]
G =
     5    10     1
    20   -10     0
    18     0     2
```

```
>> rga=G.*pinv(G).'
rga =
    0.3125    1.2500   -0.5625
    1.2500   -0.2500         0
   -0.5625         0    1.5625
```

Conclusion: of the 6 possible pairings only one has positive RGA's

Decentralized control tuning

- Independent design of each controller
  - Use when small interactions (RGA close to I)
- Sequential design (similar to cascade)
  - Start with fast loop
  - NOTE: It's possible to pair on negative RGA, but then system will go unstable if the inner loop is not active (saturates)
  - Sequential vs. independent design
    - + Sequential may have better performance, but
    - - outer loop gets slow, and
    - - loops depend on each other

Summary

Single-loop control = Decentralized control

Use for: Noninteracting process

+ Tuning may be done on-line
+ No or minimal model requirements
+ Easy to fix and change

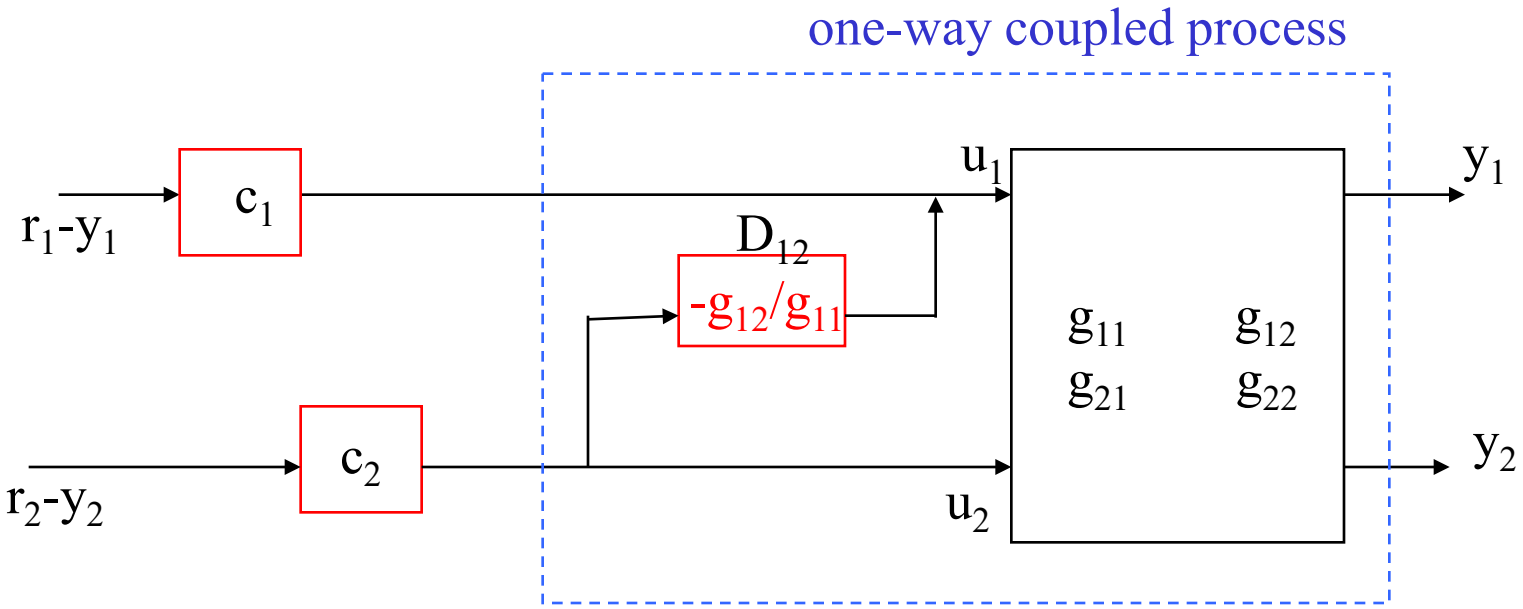
- Need to determine pairing
- Performance loss compared to multivariable control

Multivariable control

1. Single-loop control (decentralized)
2. **Decoupling (similar to feedforward)**
3. Model predictive control (MPC)

II. Decoupling

a) One-way Decoupling (improved control of y_1)



DERIVATION

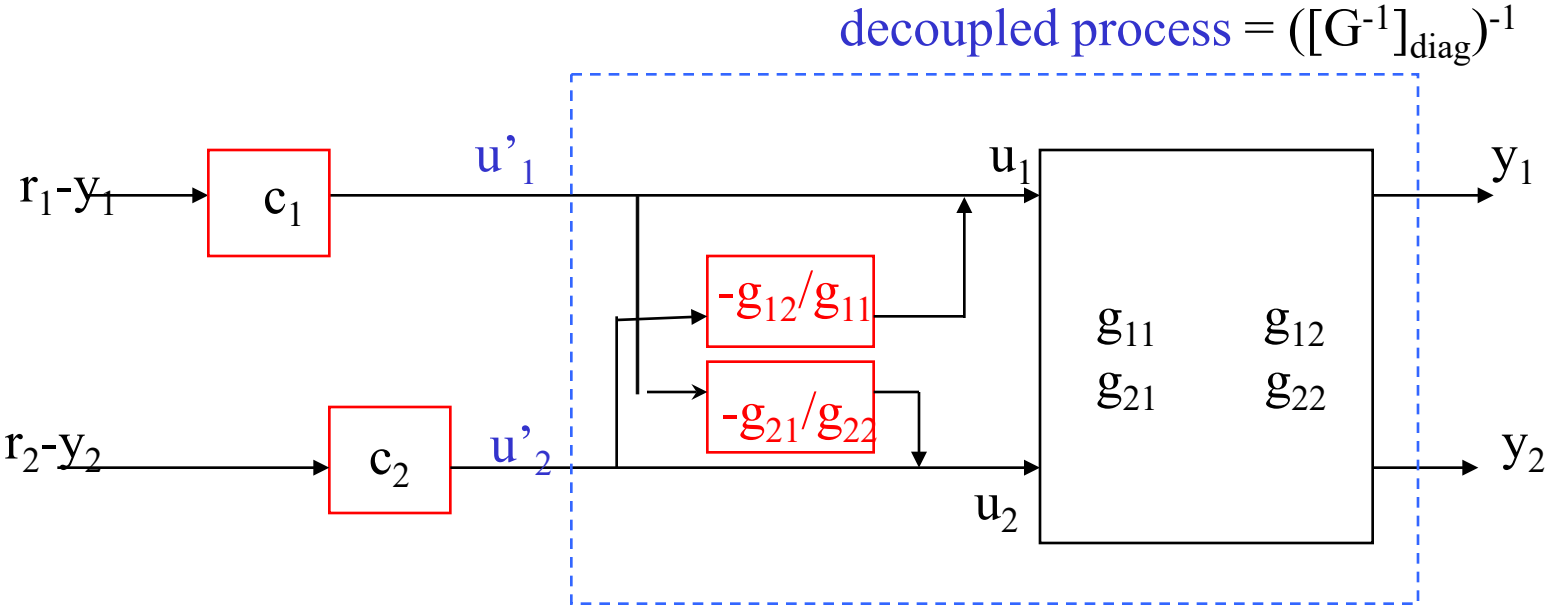
Process: $y_1 = g_{11}\, u_1 + g_{12}\, u_2$ (1)

$y_2 = g_{21}\, u_1 + g_{22}\, u_2$ (2)

Consider u_2 as disturbance for control of y_1.

Think «feedforward»: Adjust u_1 to make y_1=0. (1) gives $u_1 = -(g_{12}/g_{11})\, u_2$

b) Two-way Decoupling: Standard implementation (Seborg)

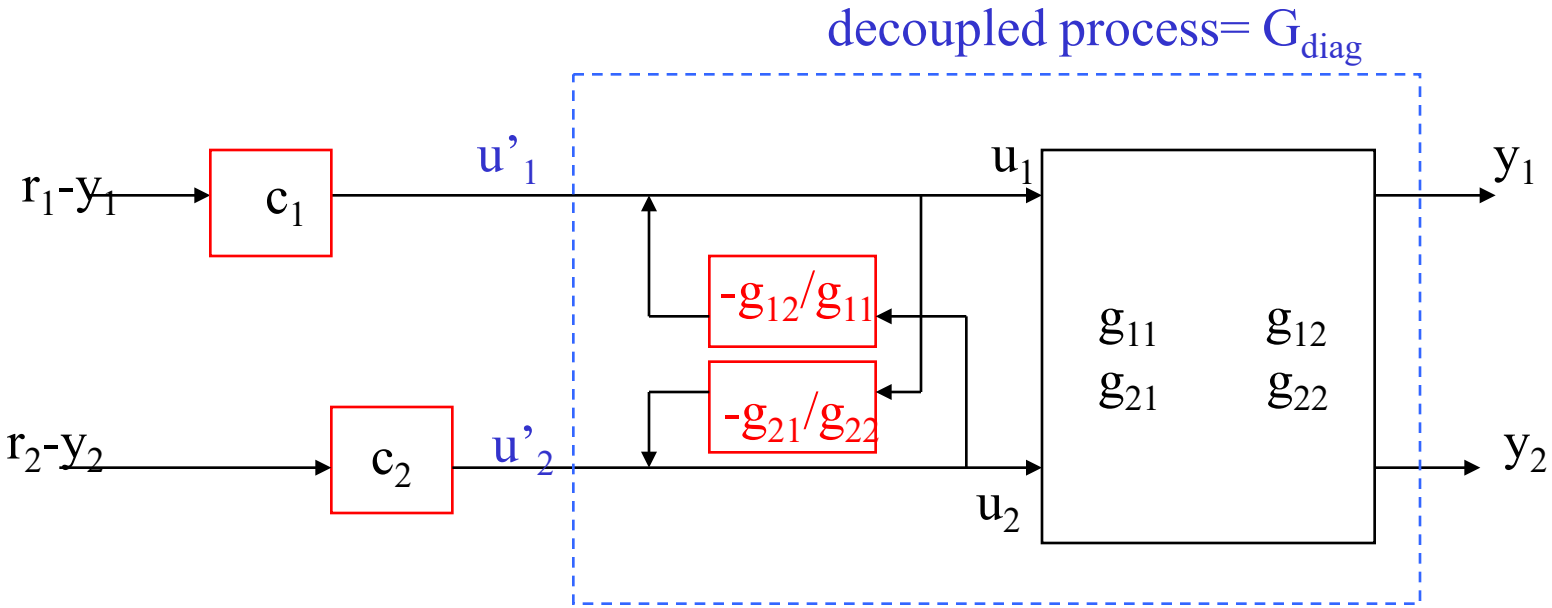


… but note that diagonal elements of decoupled process are different from G
Problem for tuning!

Process: $y_1 = g_{11}\, u_1 + g_{12}\, u_2$
Decoupled process: $y_1 = (g_{11}\text{-}g_{12}*g_{21}/g_{22})\, u_1' + 0*u_2'$
Similar for y_2.

Sigurd recommends this alternative!

Two-way Decoupling: «Inverted» implementation (Shinskey)



Advantages: (1) Decoupled process has same diagonal elements as G. Easy tuning!
(2) Handles input saturation! (if u1 and u2 are actual inputs)

Proof (1): $y_1 = g_{11}\, u_1 + g_{12}\, u_2$, where $u_1 = u_1' - (g_{12}/g_{11})u_2$.
Gives : $y_1 = g_{11}\, u'_1 + 0^*\, u_2'$
Similar: $y_2 = 0^*u_1' + g_{22}\, u_2'$

See Exercise

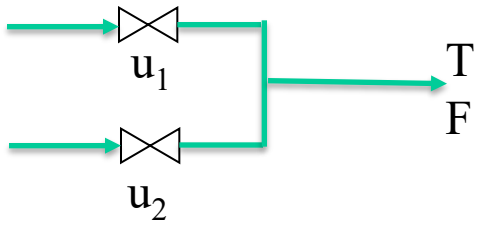
Pairing and decoupling

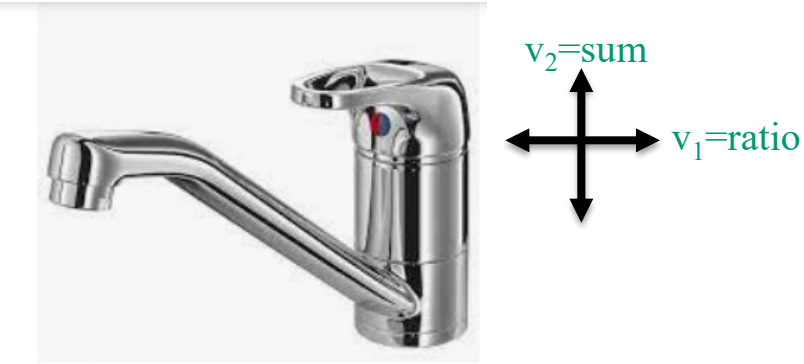
- To get ideal decoupling, diagonal elements should have smaller effective delay than the off-diagonal elements
- Thus, we should pair on elements with small effective delay (“pair close rule”)
  - This is usually achieved if we follow the “normal” RGA-rules of pairing on elements as close to 1 as possible.

Nonlinear decoupling and feedforwrd

- Linear decoupling and feedforward often work poorly because of nonlinearity
- Example of nonlinear feedforward: Ratio control
- It’s often easier to make nonlinear decoupler / feedforward based on static model or insight

Example decoupling: Mixing of hot (u_1) and cold (u_2) water





- Want to control
  - y_1 = Temperature T
  - y_2 = total flow F
- Inputs, u=flowrates
- May use two SISO PI-controllers
  - TC
  - FC
- Insight: Get decoupled response with transformed inputs
  - TC sets flow ratio, $v_1 = u_1/u_2$
  - FC sets flow sum, $v_2 = u_1 + u_2$
- Decoupler: Need «static calculation block» to solve for inputs

$$u_1 = v_1 v_2 / (1+ v_1)$$
$$u_2 = v_2 / (1 + v_1)$$

TRANSFORMED INPUTS v: GENERAL APPROACH FOR COMBINED FEEDBACK, FEEDFORWARD, DECOUPLING AND LINEARIZATION

Example: Mixing of hot and cold water

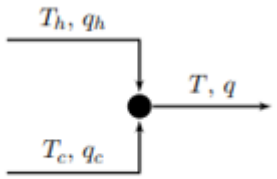
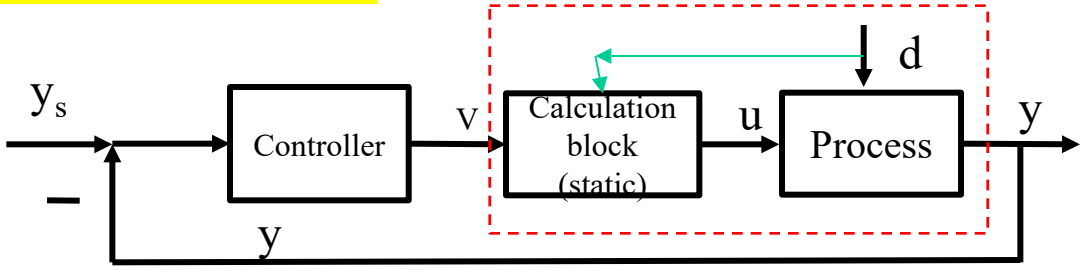


Figure 1: Mixer system



$$u = \begin{pmatrix} q_h \\ q_c \end{pmatrix} \quad d = \begin{pmatrix} T_h \\ T_c \end{pmatrix} \quad y = \begin{pmatrix} T \\ q \end{pmatrix}$$

Steady-state model written as y=f(u,d):

$$T = \frac{q_h T_h + q_c T_c}{q_h + q_c}$$

$$q = q_c + q_h$$

Select transformed inputs as right hand side, v = f

$$v_1 = \frac{q_h T_h + q_c T_c}{q_h + q_c} \quad (1)$$ Generalized ratio

$$v_2 = q_c + q_h \quad (2)$$

Model from v to y (red box) is then decoupled and with perfect disturbance rejection:

$$T = v_1$$

$$q = v_2$$

It's almost magic!

- Can then use two single-loop PI controllers for T and q!
  - These controllers are needed to correct for model errors and unmeasured disturbances
- Note that v_1 used to control T is a generalized ratio, but it includes also feedforward from Tc and Th.

Implementation (calculation block) : Solve (1) and (2) with respect to u=(qc qh):

Decoupler with feedforward:

$$q_h = \frac{v_2(v_1 - T_c)}{T_h - T_c}$$

$$q_c = v_2 - q_h$$

V. Pole placement (state feedback using P-only control)

- Place closed-loop poles. Old design method
- Useful for insight, but difficult to choose K. Not used much in practice, at least not for linear controllers
- Basis:
  - Linear system on state space form

$$dx/dt = A\,x + B\,u$$

  - and use "State feedback" P-control (assuming we can measure all the states)

$$u = K\,x$$

Note

1. SIMC is "pole placement" (p=-1/tauc), but with output feedback (y), and we also place zeros
2. If we cannot measure all the states, then we can estimate x from y using a "Kalman filter".
3. State feedback uses extra measurements – an alternative is cascade control

8.5.1 Stability and state feedback

The poles of the transfer function, which are the zeros of its denominator polynomial, determine the dynamic characteristics of the system, in particular its stability and its damping characteristics. Transferring this statement to equation (8.60), it follows that the roots of the equation

$$\det(s \cdot \boldsymbol{I} - \boldsymbol{A}) = 0 \tag{8.67}$$

are essential for the behaviour of the system. The determinant in equation (8.67) is a n-th order polynomial in s and corresponds to the characteristic polynomial. The roots of the determinant in equation (8.67) are also designated as the eigenvalues of the matrix $\boldsymbol{A}$. All of them must exhibit negative real parts, if the system described by the matrix $\boldsymbol{A}$ is supposed to be stable.

which will be combined to yield

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B} \cdot \boldsymbol{K}) \cdot \boldsymbol{x} \tag{8.71}$$

Equation (8.71) describes a system without any input variables with the system matrix

$$\boldsymbol{A}_K = \boldsymbol{A} - \boldsymbol{B} \cdot \boldsymbol{K} \quad . \tag{8.72}$$

8.5.2 Pole placement

One possibility for the controller design is to select desirable eigenvalues of the matrix A_K and to determine from this and the known matrices A and B the controller or feedback matrix K.

As an example a state feedback is to be determined according to the mentioned procedure of pole placement for a transfer system with a single input and a single ouput variable. Figure 8-6 shows the functional diagram of the system with feedback.

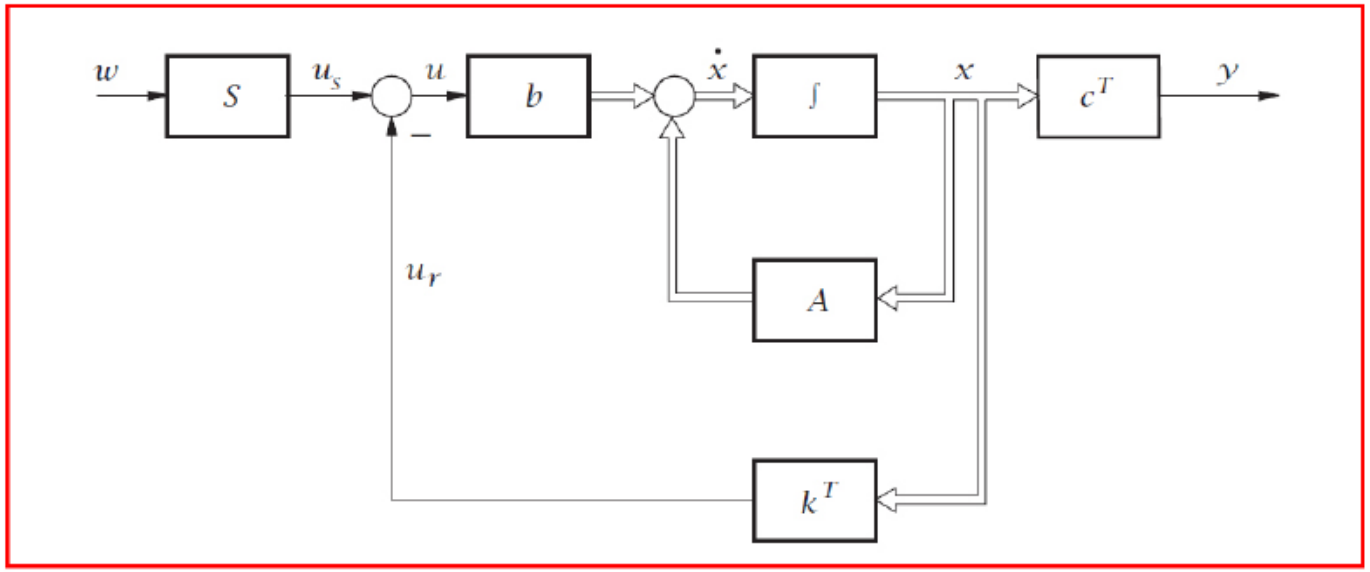


Figure 8-6: SISO-system with state feedback

The transfer system may be be stated in controller canonical form according to equation (8.23). The state variables of the controller canonical form can be obtained for this purpose by transformation of the original state variables in the way described in chapter 8.2. According

MPC

Advanced multivariable control with explicit constraint handling = MPC

Use for: Interacting process and changes in active constraints

+ Easy handling of feedforward control
+ Easy handling of changing constraints
  - no need for logic
  - But does not always work
- Requires multivariable dynamic model
- Tuning may be difficult
- Less transparent; not clear how it handles a given situatiion
- Used if "conventional advanced control" is not good enough
  - Typically implemented after some time of operation (more than one year)
- "Everything goes down at the same time"

MPC = model predictive control

Multivariable control:
MPC versus (linear) decoupling

- Both MPC and decoupling require a multivariable process model
- MPC is usually preferred instead of decoupling because it can also handle feedforward control, nonsquare processes (cascade, input resetting) dynamic, etc.
- MPC can also handle constraints
  - Has “built-in” anti-windup

MPC = Model predictive control

Model predictive control (MPC) = "online optimal control"

The quadratic program of equations (1)-(5) is solved each control sample to find the optimal control actions.

$$\min_{\Delta u} y_{dev}^T Q_y y_{dev} + u_{dev}^T Q_u u_{dev} + \Delta u^T P \Delta u \quad (1)$$

$$u_{\min} < u < u_{\max} \quad (2)$$

$$\Delta u_{\min} < \Delta u < \Delta u_{\max} \quad (3)$$

$$y_{\min} < y < y_{\max} \quad (4)$$

$$y = M(y,u,d,v) \quad (5)$$

$y_{dev}=y-y_s$

$u_{dev}=u-u_s$

Discretize in time:

$$y = [y_1\ y_2 \ldots y_n]$$

$$u = [u_1\ u_2 \ldots u_k]$$

$$\Delta u = [\Delta u_1\ \Delta u_2 \ldots \Delta u_k]$$

$$\Delta u_i = u_i - u_{i-1}$$

Note: Implement only current input Δu_1

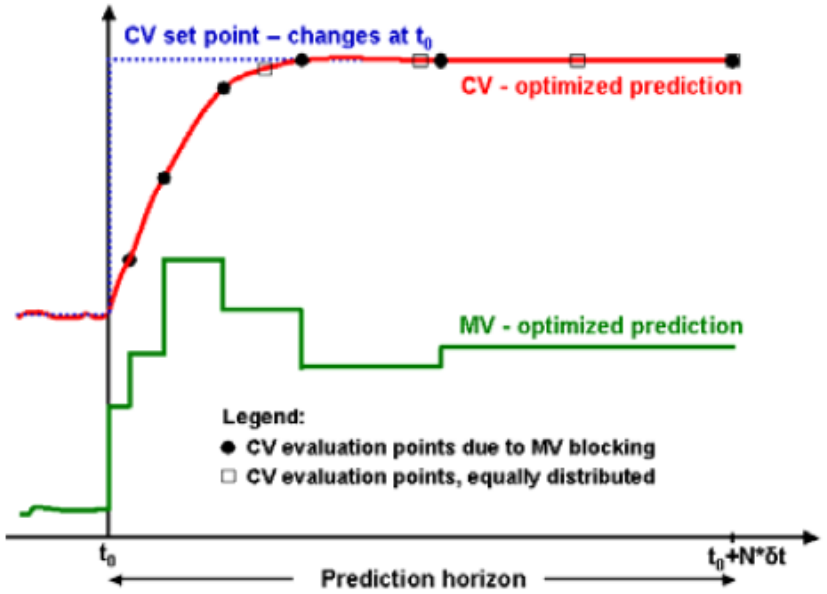


Fig. 1. MV blocking and CV evaluation

The quadratic objective function (1) penalizes CV (y) deviations from set point, MV (u) deviations from ideal values, and MV moves. The constraints are: MV high and low limits (2); MV rate of change limits (3); and CV high and low limits (4). The dynamic model (5) predicts the CV response from past and future CV and MV values as well as past DV (d) values and estimated and optionally predicted unmeasured disturbances v.

Implementation MPC project (Stig Strand, Equinor)

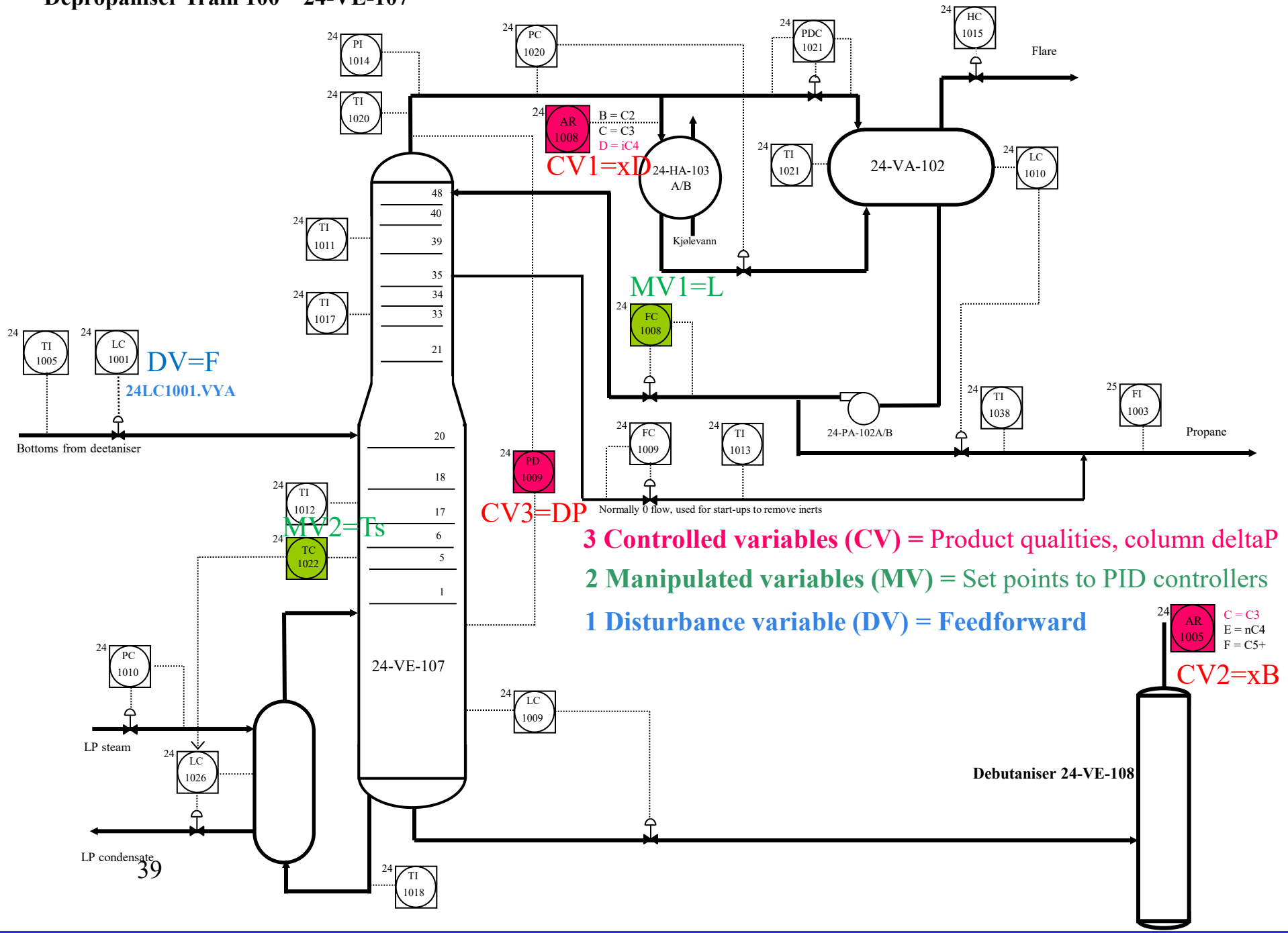
- Initial MV/CV/DV selection
- DCS* preparation (controller tuning, instrumentation, MV handles, communication logics etc)
- Control room operator pre-training and motivation
- Product quality control → Data collection (process/lab) → Inferential model ("soft sensor")

- MV/DV step testing → dynamic models

- Model judgement/singularity analysis → remove models? change models?
- MPC pre-tuning by simulation → MPC activation – step by step and with care – challenging different constraint combinations – adjust models?
- Control room operator training
- MPC in normal operation, with at least 99% service factor

*DCS = "distributed control system" = Basic PID control layer

Depropaniser Train 100 – 24-VE-107



39

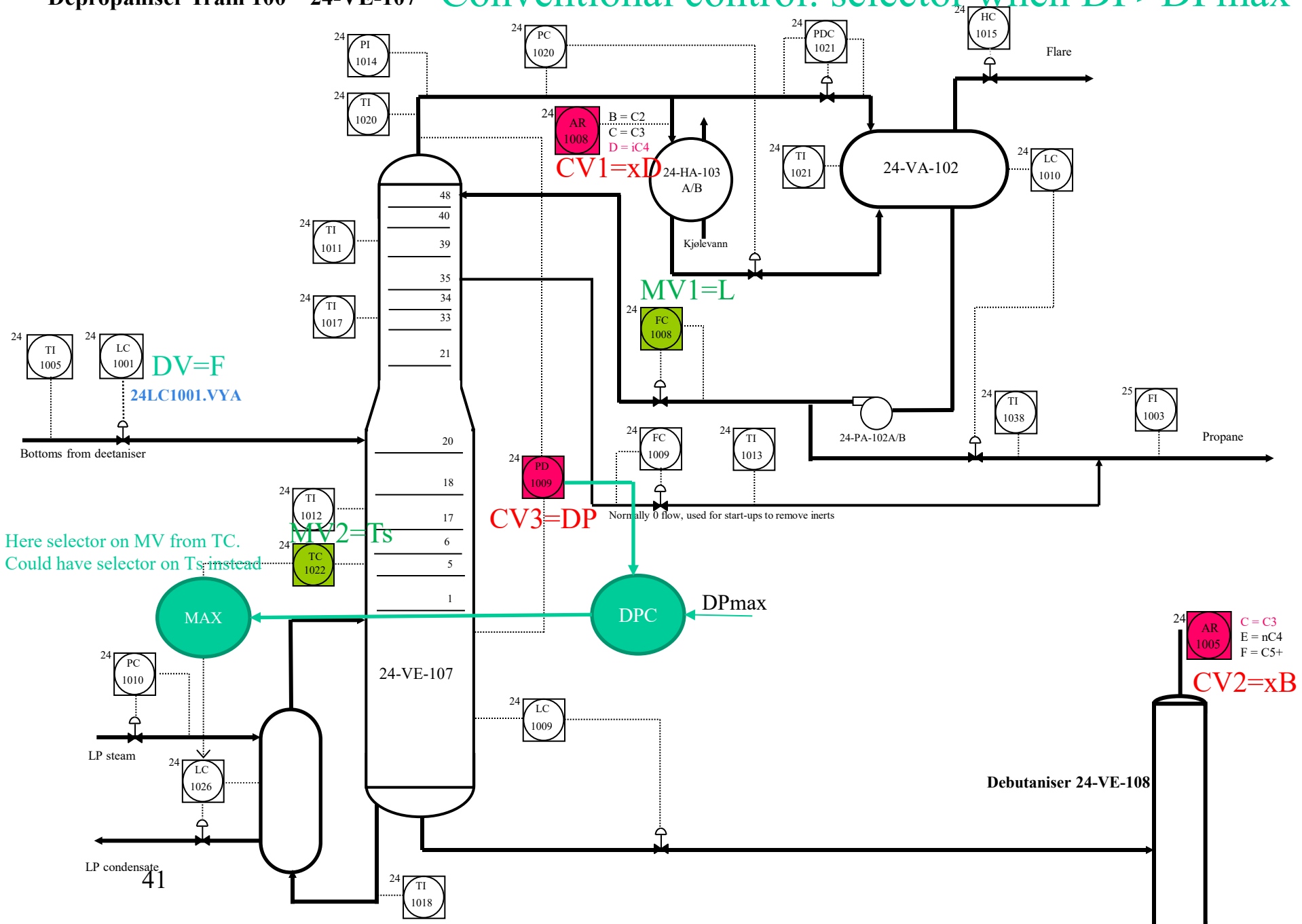
Conventional control (PID)

Select pairings MV-CV (obvious here: L-xD, Ts-xB)
Selector: Give up xB when we meet DP constraint

Tune 3 PI controllers

Depropaniser Train 100 – 24-VE-107

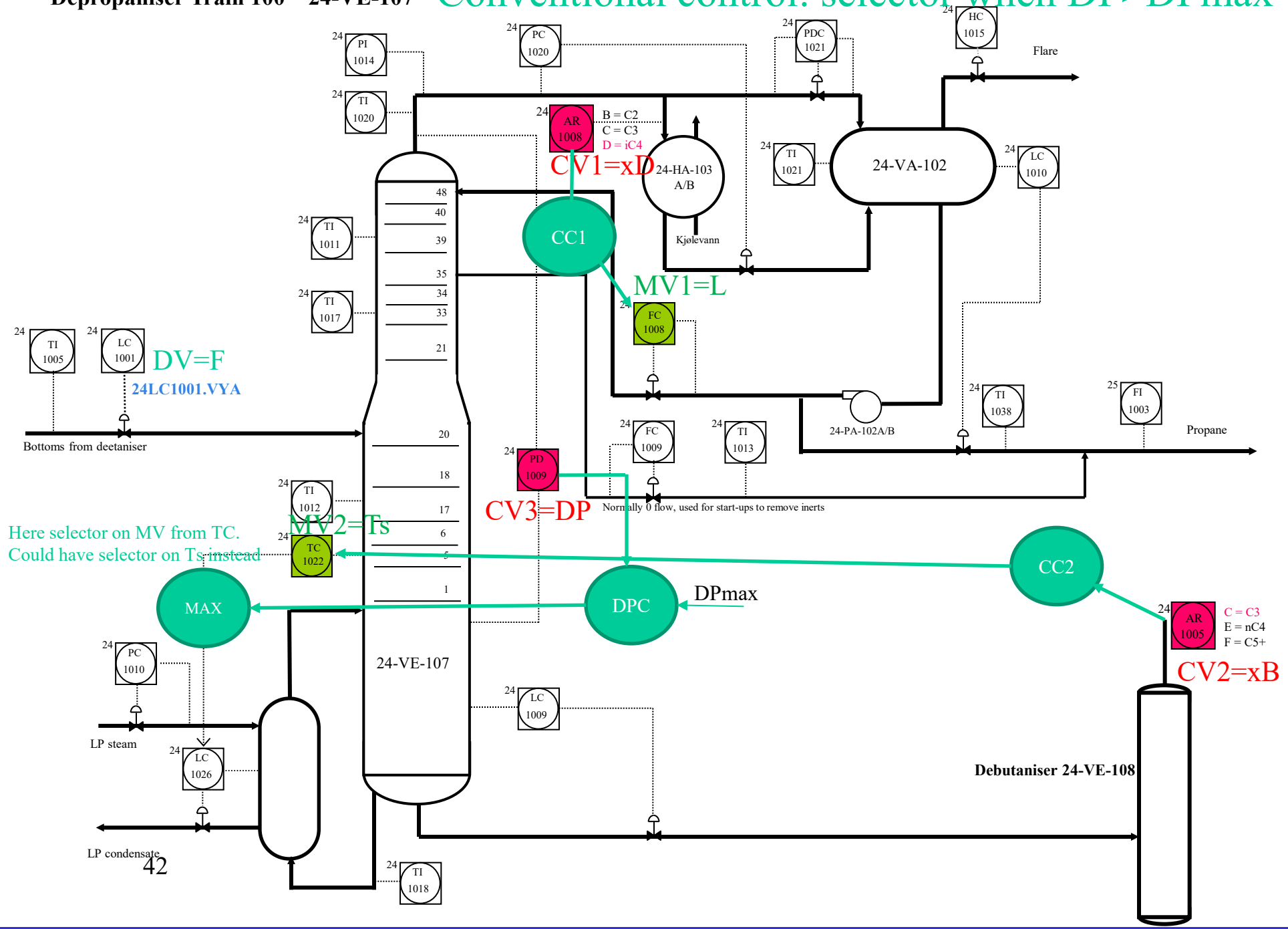
Conventional control: selector when DP>DPmax



41

Depropaniser Train 100 – 24-VE-107

Conventional control: selector when DP>DPmax



42

Conventional control (PID)

Select pairings MV-CV (obvious here: L-xD, Ts-xB)
Selector: Give up xB when we meet DP constraint

Tune 3 PI controllers (CC1, CC2, DPC)

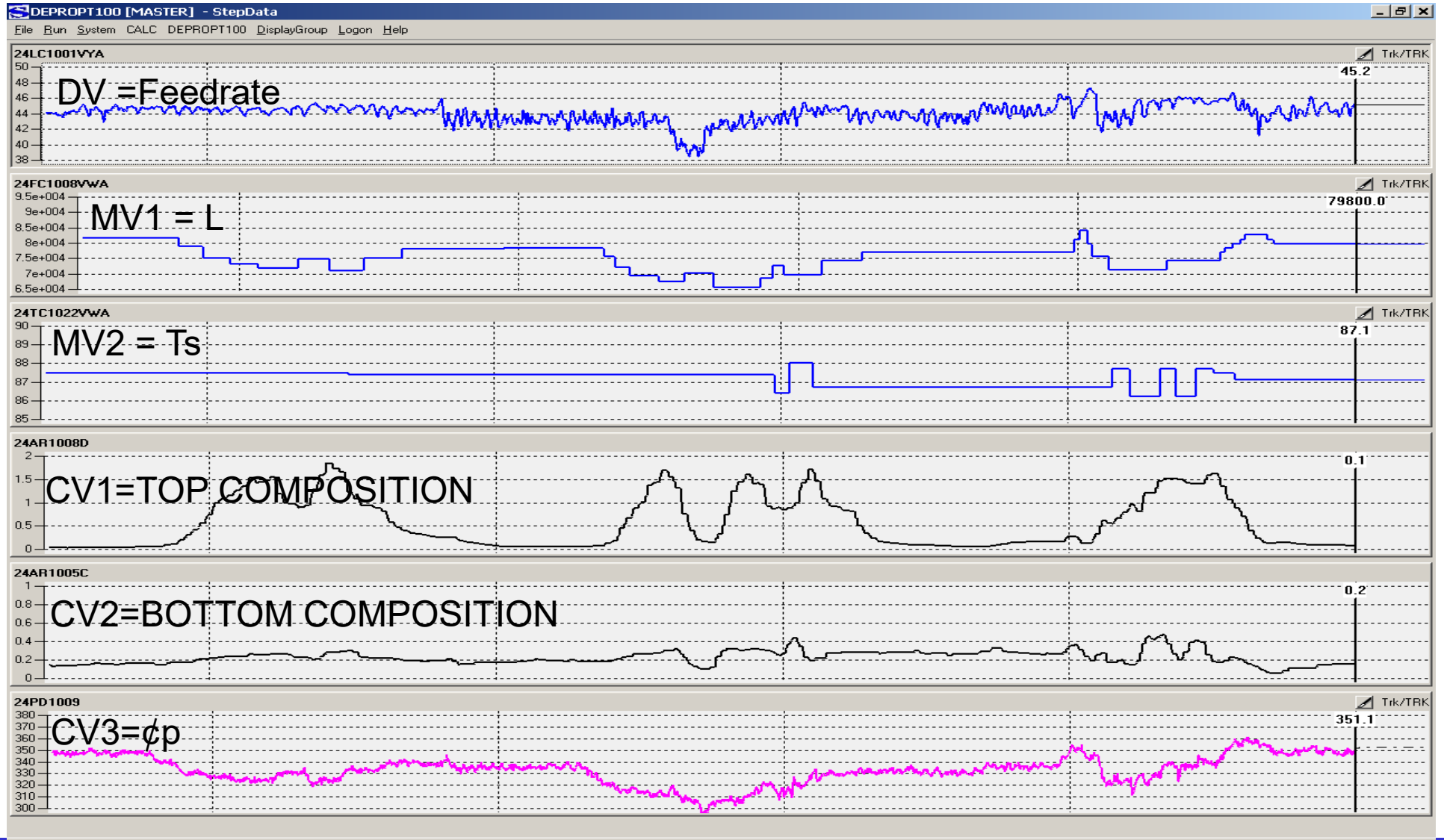
MPC

Don’t need to make pairing choices

But need model
And need to tune MPC controller

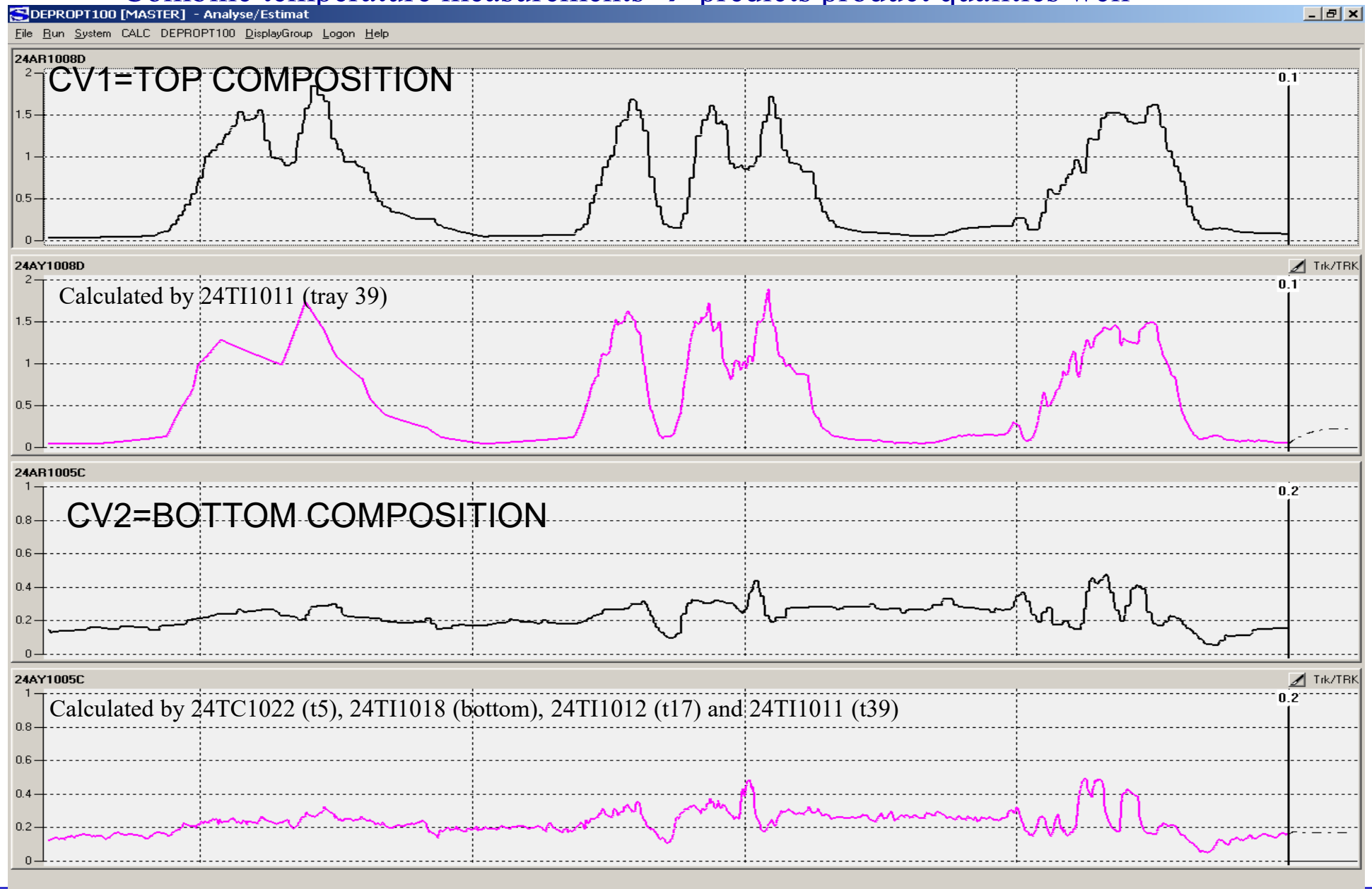
Depropaniser Train100 step testing

- 3 days – normal operation during night
-



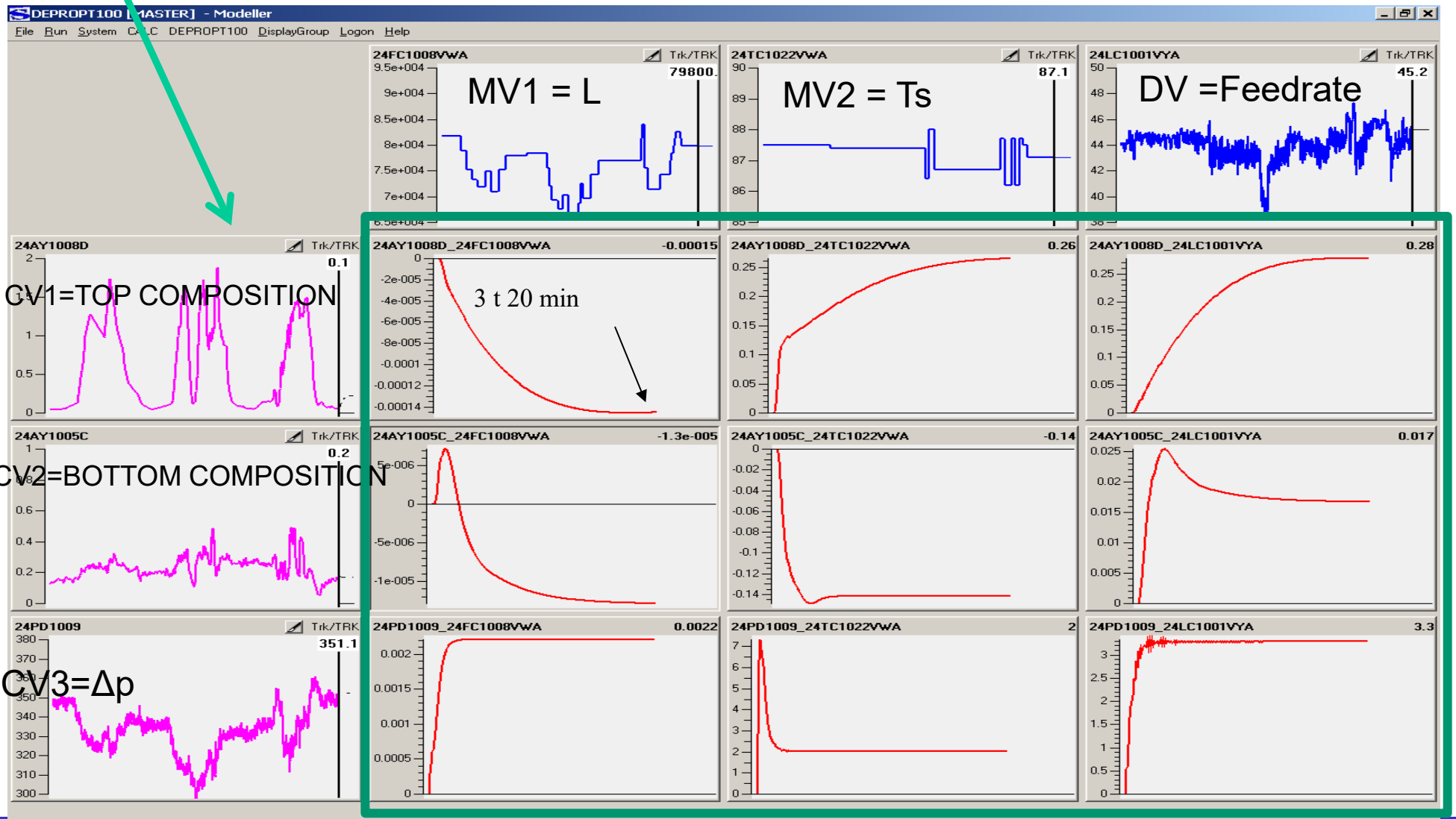
Estimator: inferential models

- Analyser responses are delayed – temperature measurements respond 20 min earlier
- Combine temperature measurements → predicts product qualities well



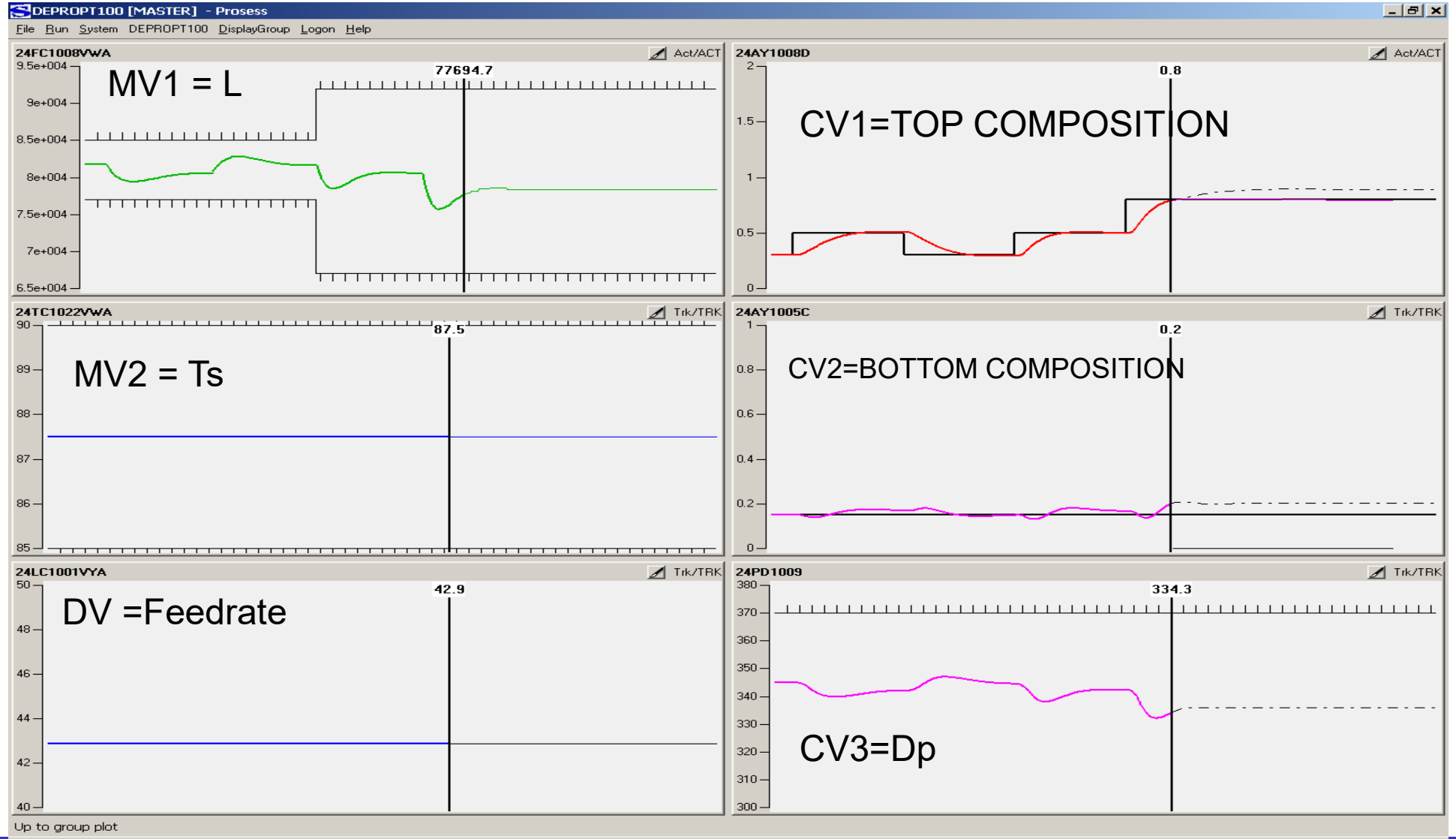
Depropaniser Train100 step testing – Final model

- Step response models:
  - MV1=reflux set point increase of 1 kg/h
  - MV2=temperature set point increase of 1 degree C
  - DV=output increase of 1%.

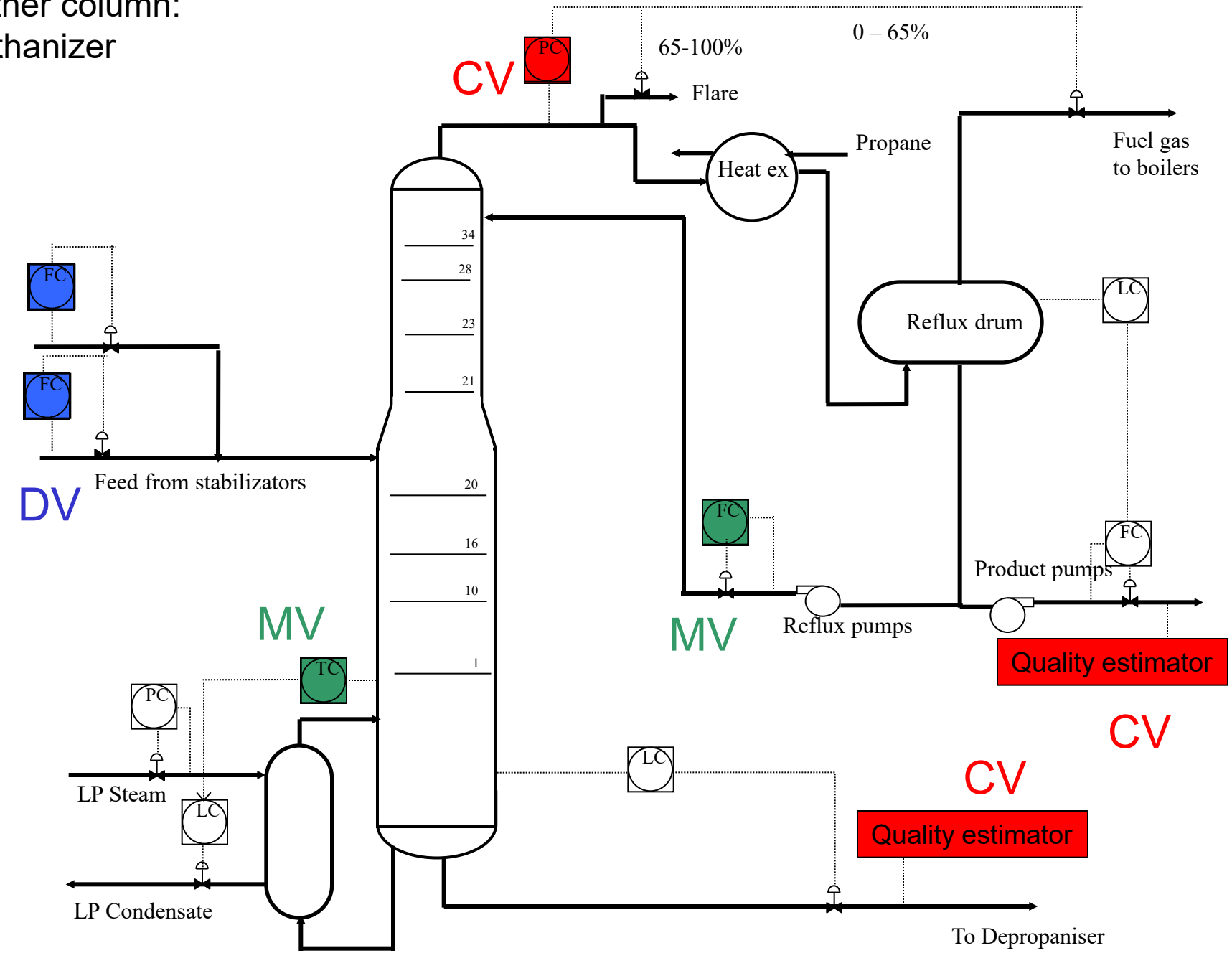


Depropaniser Train100 MPC – controller activation

- Starts with 1 MV and 1 CV – CV set point changes, controller tuning, model verification and corrections
- Shifts to another MV/CV pair, same procedure
- Interactions verified – controls 2x2 system (2 MV + 2 CV)

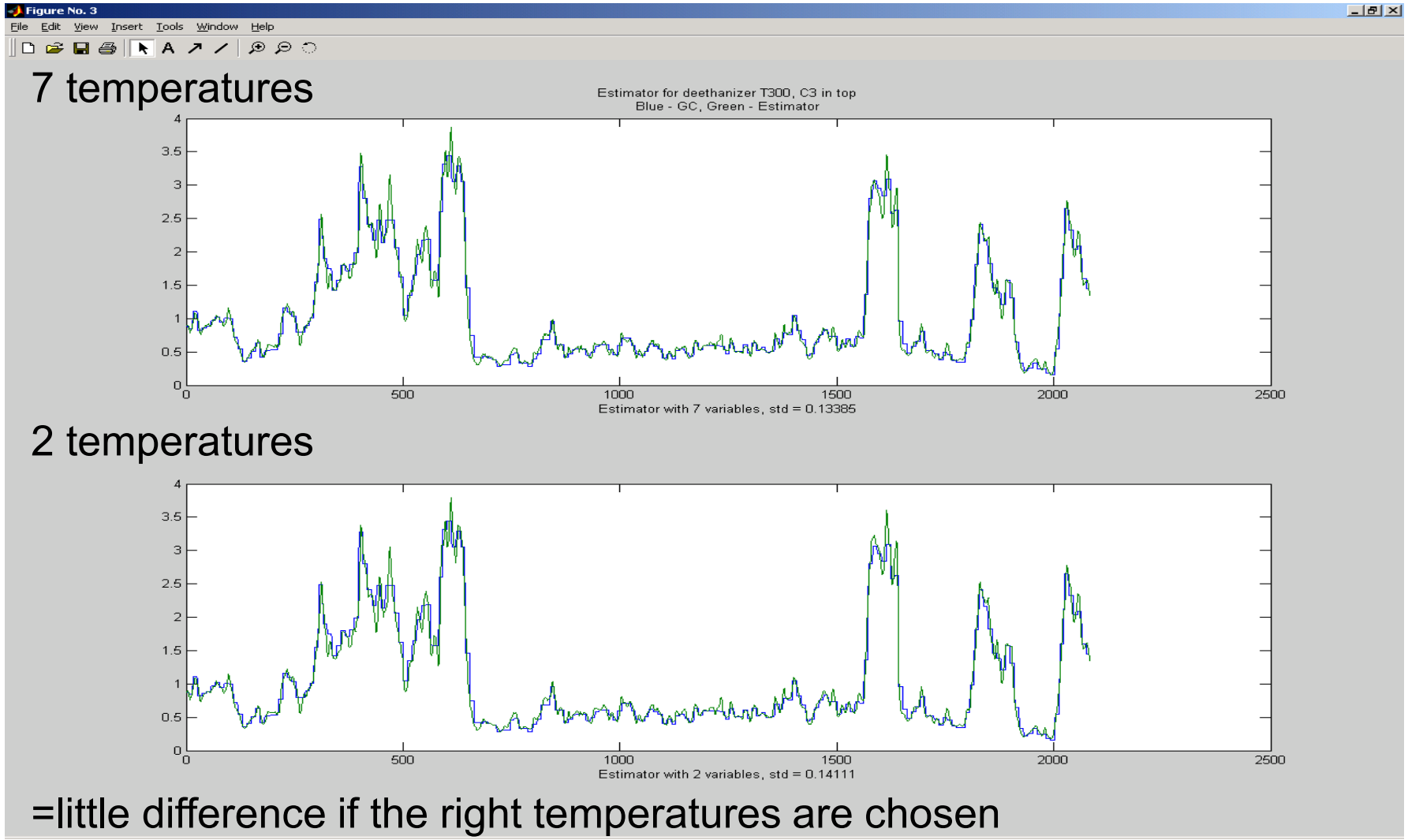


Another column: Deethanizer



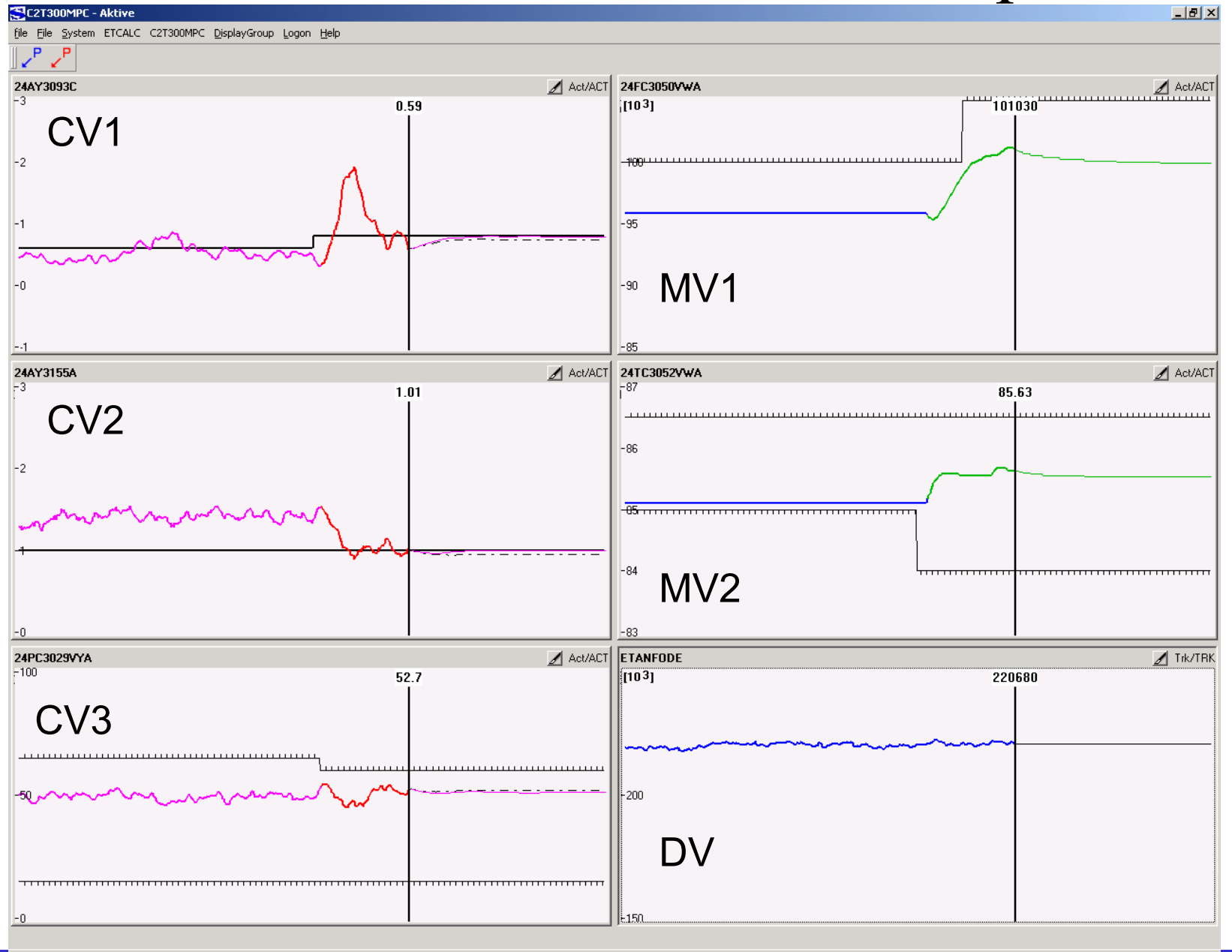
Top: Binary separation in this case
Quality estimator vs. gas chromatograph

(use logarithmic composition to reduce nonlinearity, CV = - ln $x_{impurity}$)



=little difference if the right temperatures are chosen

The final test: MPC in closed-loop



Conclusion MPC

- Sometimes simpler than previous advanced control
- Well accepted by operators
- Equinor: Use of in-house technology and expertise successful