Lecture Notes 3: the Heat Exchanger Network for Heat Recovery

chemical plant designs use heat integration to reduce energy costs

Suppose you have a hot process stream that needs cooling. Perhaps you could use a utility stream, such as cooling water. The energy transferred to the cooling water would then be discarded to the plant environment (via cooling tower or river discharge). Suppose, however, that you have another process stream that needs heating. IF the desired enthalpy changes of the two process streams are similar, IF the temperature differences support practical heat transfer rates, IF there is no safety problem routing these streams through a common piece of equipment, IF the two streams are in reasonable proximity, etc., THEN it may pay to recover the energy of the hot stream into the cold one, and thus save utility costs.

saving utility costs is good, but operation becomes more complicated

We have given up flexibility by tying two process streams together; it may be that starting up, shutting down, or performing maintenance will affect larger portions of the plant. Furthermore, operating disturbances that affect one process stream may now propagate into the other. More complicated plant structures motivate more sophisticated process control to cope with these new problems.

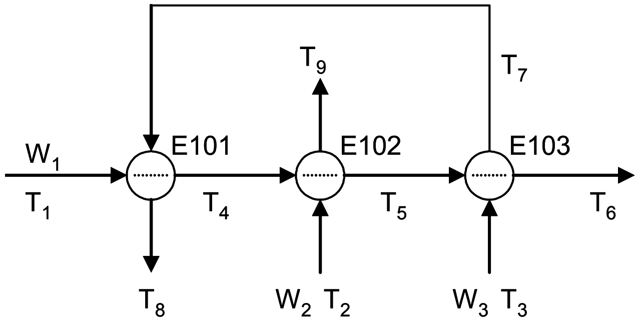
use steady-state process simulators for flowsheet development

Codes such as Aspen Plus, Chemcad, and Hysys are routinely used to develop chemical process flowsheets. They calculate the thermodynamic state of each stream in the flowsheet. These stream properties must, of course, satisfy the steady-state material and energy balances. To complete the calculations requires models for performance of the flowsheet equipment, as well as physical properties.

We will not use these tools in this module, but we will remain mindful of how they would be used. In what follows, therefore, we can anticipate (1) writing material and energy balances (2) describing the performance of equipment (3) using physical property data.

we present an example of heat integration and classify the variables

Consider the following heat recovery network, based on an example presented by McAvoy (2):



To rationalize this example, imagine that we care about all the exit temperatures – this would be the case if we were processing each stream separately in heat exchangers with utility streams. Thus T_6, T_8, and T_9 are classified as controlled variables (system outputs that we wish to maintain at a target value). If the primary purpose of the network is to cool stream 1, then mass flow W_1 and inlet temperature T_1 are regarded as determined by another process; for our purpose they are disturbance variables (system inputs that we must accept as they come). Similarly, T_2 and T_3 are determined by the suppliers of these streams, so they are also disturbance variables.

We cannot exert control over the outlet temperatures unless we have some inputs to use as manipulated variables; these must be W_2 and W_3. Why should W_2 and W_3 be ours to manipulate and W_1 not, if they are all received streams? Perhaps flows W_2 and W_3 are drawn from some larger supply, so that our calling for more or less of these streams at any time does not badly affect either the source or the destination of these streams. (If we should not be manipulating these streams, then we cannot expect to exert the control we desire on the outlet temperatures, and we had better seek another heat integration scheme.)

There remain temperatures T_4, T_5, and T_7, which cannot be directly manipulated, are not input disturbances, but are not to be controlled. We classify these as intermediate variables – they depend on inputs, and they either affect outputs or are themselves uncontrolled outputs of the system. Our shower process in Lecture Notes 1 and 2 was simple enough to have no intermediate variables, but we encounter them in most cases.

we write material and energy balances for steady-state operation

Wait. We have been talking about disturbances and control and changing stream flow rates – how can we be content with a steady-state balance?

At this stage of design, we are simply not ready to quantify transient behavior. Our job is to specify a suitable reference operating condition and size the heat exchangers appropriately, and that requires a steady state description, such as we would have from a steady-state process simulator. We will use the RGA (Relative Gain Array) and DC (Disturbance Cost) to examine the merits of various control loops we might consider, and a steady-state model will be sufficient for that.

The steady state material balances are true, but not informative, so we shall omit them. A steady-state energy balance on the cold stream in a heat exchanger will be

$$W\int_{T_{ref}}^{T_{in}} C_p dT + Q = W\int_{T_{ref}}^{T_{out}} C_p dT \tag{3-1}$$

The enthalpy of the stream is calculated from a thermodynamic reference temperature T_{ref}, and the heat capacity C_p may vary with temperature. Q is the heat duty of the exchanger, supplied to stream W through the heat transfer surface. Combining the inlet and outlet streams,

$$\begin{aligned} Q &= W\int_{T_{in}}^{T_{out}} C_p dT \\ &= W\overline{C_p}\left(T_{out} - T_{in}\right) \end{aligned} \tag{3-2}$$

where the heat capacity has been averaged over the temperature interval. For our purposes (learning about operability screening tools to be used in preliminary-stage design) we will make matters simple and assume that C_p does not vary with temperature.

Thus far we have energy balances and physical properties; we also need a performance model of the heat exchanger. We will use

$$Q = UA\frac{\left(T_{hot,in} - T_{cold,out}\right) - \left(T_{hot,out} - T_{cold,in}\right)}{\ln\dfrac{\left(T_{hot,in} - T_{cold,out}\right)}{\left(T_{hot,out} - T_{cold,in}\right)}} \qquad (3\text{-}3)$$

in which we will neglect how the overall coefficient U varies with operating conditions.

the equations in summary

HX101

$$\begin{aligned} 0 &= -W_1C_{p1}T_1 + W_1C_{p1}T_4 + Q_1 \\ 0 &= W_3C_{p3}T_7 + Q_1 - W_3C_{p3}T_8 \\ \text{where } Q_1 &= U_1A_1\frac{\left(T_1 - T_8\right) - \left(T_4 - T_7\right)}{\ln\dfrac{\left(T_1 - T_8\right)}{\left(T_4 - T_7\right)}} \end{aligned} \qquad (3\text{-}4)$$

HX102

$$\begin{aligned} 0 &= -W_1C_{p1}T_4 + W_1C_{p1}T_5 + Q_2 \\ 0 &= W_2C_{p2}T_2 + Q_2 - W_2C_{p2}T_9 \\ \text{where } Q_2 &= U_2A_2\frac{\left(T_4 - T_9\right) - \left(T_5 - T_2\right)}{\ln\dfrac{\left(T_4 - T_9\right)}{\left(T_5 - T_2\right)}} \end{aligned} \qquad (3\text{-}5)$$

HX103

$$\begin{aligned} 0 &= -W_1C_{p1}T_5 + W_1C_{p1}T_6 + Q_3 \\ 0 &= W_3C_{p3}T_3 + Q_3 - W_3C_{p3}T_7 \\ \text{where } Q_3 &= U_3A_3\frac{\left(T_5 - T_7\right) - \left(T_6 - T_3\right)}{\ln\dfrac{\left(T_5 - T_7\right)}{\left(T_6 - T_3\right)}} \end{aligned} \qquad (3\text{-}6)$$

if we had been using a steady-state process simulator we would not see the equations

Still, they would be something like those we have written. The temperature dependence of physical properties would be accounted for, and perhaps the heat exchanger model would be more elaborate, but they would feature the combination of *the-energy-must-balance* and *how-the-equipment-performs* that we see above.

remember, these are steady-state equations!

In writing balances for the shower process, we set the accumulation terms to zero because that was a reasonable approximation to process behavior. Here, we have done no such thing – we have set the accumulation terms to zero because we are looking only at steady state.

However, steady-state does not mean static! Suppose we define a reference state for all the variables. Of course these must satisfy the equation set. Then we change one of the inputs, say temperature T_1. For some period of time following, our equation set will not apply, because the disturbance in T_1 changes the accumulation of energy in the system and variables are changing in time. Ultimately, however, we expect the system to settle down to a new steady state whose variables are more or less different from the reference values, because of the effect of variable T_1. This steady state would be described by (3-4) through (3-6).

we use the equations for the design case, in which our objective is to choose the areas
We want to find out how large the heat exchangers must be to operate at the reference condition, and what manipulated variable flow rates we should supply. This means we specify system parameters like C_p and U, pin down disturbance variables at their reference values, and specify the output variables to describe the system performance that we want. Then we calculate the required heat exchanger areas and manipulated variables. Notice, however, that two of the intermediate variables must be specified, as well. Notice the subscript "r", which indicates the reference condition, or design case.

design case

specify:	calculate:
C_{p1}, C_{p2}, C_{p3}, U_1, U_2, U_3	A_1, A_2, A_3
W_{1r}, T_{1r}, T_{2r}, T_{3r}	Q_{1r}, Q_{2r}, Q_{3r}
T_{4r}, T_{5r}, T_{6r}, T_{8r}, T_{9r}	W_{2r}, W_{3r}, T_{7r}

we use the equations for the rating case, in which our objective is to calculate outputs
After arriving at a design, we are interested in how it operates under different conditions. Now the heat exchanger areas are fixed; we specify all inputs and calculate intermediate and output variables. In describing the rating case no subscript is used, because the flow and temperature variables can take any value that satisfies the energy balance and performance equations.

rating case

specify:	calculate:
C_{p1}, C_{p2}, C_{p3}, U_1, U_2, U_3	Q_1, Q_2, Q_3
A_1, A_2, A_3	T_4, T_5, T_6
W_1, W_2, W_3, T_1, T_2, T_3	$T_{7,}$ T_8, T_9

we calculate either steady-state case by specifying 15 quantities and calculating 9
Ideally, we would solve the 9 equations for the 9 unknowns. We can quickly eliminate the 3 heat duties, but we are then left with 6 nonlinear equations that are implicit in the temperatures. It is unlikely, therefore, that we will be able to isolate all desired unknown variables. Hence, our calculation method will have to be iterative. We will consider two alternatives:

In the design case, we know most of the temperatures (because we are specifying both inlet and outlet conditions), which makes it straightforward to calculate the areas. Hence we can vary the two manipulated variables and other temperatures to seek a combination of areas that we like. An example is given in "HX network.xls". It is important to represent all three heat exchangers in the spreadsheet, so that all equations are simultaneously satisfied.

In the rating case, we are seeking intermediate and outlet temperatures. We can arrange the 6 nonlinear equations, which are implicit in our 6 unknown temperatures, to equal zero. We can then use a numerical root-finding scheme to calculate the temperatures that satisfy the equations. We will discuss root-finding in further detail below.

In summary, we can calculate steady-state performance as required, so that we can arrive at a design for our network. Next, we must consider if the network will be operable.

before we proceed, recall what we learned from the shower process

Our intention is to use RGA and DC to screen the heat exchanger network for potential controllability problems. Let us compare our plan with our experience from the shower:

<table>
<tr><th>Shower</th><th>HX network</th></tr>
<tr><td colspan="2">Compute reference conditions from steady-state equations</td></tr>
<tr><td>The output variables T and F were explicitly calculated from 1 linear and 1 nonlinear equation.</td><td>The output variables are implicit within 6 nonlinear equations. An iteration scheme is necessary to calculate the reference steady state. The result will be numbers, not a mathematical expression that shows dependencies.</td></tr>
<tr><td colspan="2">Classify the variables</td></tr>
<tr><td>2 CV, 2 DV, 2 MV.
With DV and MV specified, we have 2 equations to calculate CV.</td><td>3 CV (T_6, T_8, T_9), 4 DV (W_1, T_1, T_2, T_3), 2 MV (W_2, W_3), and 3 IV – intermediate variables – (T_4, T_5, T_7). With DV and MV specified, we have 6 equations to eliminate IV and calculate CV.</td></tr>
<tr><td colspan="2">Derive a linear system approximation to the equation set:
$\underline{y}' = \underline{\underline{P}}_m \underline{x}'_m + \underline{\underline{P}}_d \underline{x}'_d$ (or if scaled, $\underline{y}^{*'} = \underline{\underline{P}}^*_m \underline{x}^{*'}_m + \underline{\underline{P}}^*_d \underline{x}^{*'}_d$)</td></tr>
<tr><td>Taylor series used to obtain analytical expressions for the elements in $\underline{\underline{P}}_m$ and $\underline{\underline{P}}_d$.</td><td>We decline to exert the effort required to differentiate the 6 equations and solve for $\underline{y}$’. Instead, we will calculate the elements in $\underline{\underline{P}}_m$ and $\underline{\underline{P}}_d$ by numerical differentiation. However, we will forego the insight that a mathematical expression could convey.</td></tr>
<tr><td colspan="2">Obtain RGA and DC from $\underline{\underline{P}}_m$ and $\underline{\underline{P}}_d$.</td></tr>
<tr><td>RGA and DC were mathematical expressions, from which we could both infer system behavior and calculate specific cases.</td><td>RGA and DC will be numbers. We must calculate various cases and hope to infer system behavior.</td></tr>
</table>

what we gain and lose by these numerical approximations

We avoid all the algebra of writing Taylor series, but we get only a matrix of numbers. With the analytical solution, we package the entire system behavior into a mathematical expression, which we can use to understand its behavior. With the numerical procedure, we must run multiple cases, and then compare tables of numbers, to sense how the system will operate.

we must develop several tools before we can proceed with this plan
If we start to get bogged down in details, it will help to remember the side-by-side comparison presented above. These numerical methods are just more ways to accomplish the same steps we followed in our simpler shower problem.

root-finding
Let us think of our six equations as a nonlinear vector function $\underline{F}$. A valid steady-state is described by

$$\underline{F} = 0 \tag{3-7}$$

Thus the root of $\underline{F}$ is the vector of 3 CV and 3 IV that satisfy the energy balances and performance equations at steady-state (for specified values of DV, MV, and parameters). Numerical root-finding is done by an algorithm that seeks to drive F to zero by varying values of the root.

- guess root value $\underline{x}_{guess}$
- calculate the function $\underline{F}$ at the guessed root $\underline{F}(\underline{x}_{guess})$
- is $\underline{F}$ close enough to zero?
- if so, declare $\underline{x}_{guess}$ the root
- if not, modify $\underline{x}_{guess}$ in *some way that depends on what was learned in calculating $\underline{F}$* and go around again

This algorithm is easiest to visualize for scalar F and x, but applies equally to our vectors. The trick is specifying the italicized part of the last step. Various schemes are used – repeated bisection of the interval of x known to contain the root, aiming at the root by calculating the slope of F at each guessed value of the root, fitting lines or polynomials to the function and finding their roots as improved guesses – and with a little library work, we can find about them in detail. For our purposes, we can use a routine developed by Prof. Beers for 10.34, and made available on the OpenCourseWare site (1).

The end result is that although we cannot solve our equations for T_6, T_8, T_9 in general, we can compute appropriate values for these CV in specific cases.

numerical differentiation
We can also obtain a numerical estimate of a derivative:

$$\frac{\partial y_i}{\partial x_j} \approx \left.\frac{y_i(x_j + \delta x) - y_i(x_j - \delta x)}{2\delta x}\right|_{\text{all } x_k \neq x_j \text{ unchanged}} \tag{3-8}$$

where δx should be small enough to approximate the slope of y_i but not so small that numerical round-off errors dominate. Each determination of y_i in (3-8) is done by numerical root-finding.

linearization

Any one of our controlled variables (call it y_i) is related to the input (manipulated and disturbance) variables $\underline{x}_m$ and $\underline{x}_d$ through the nonlinear equation set $\underline{F}$. We use the Taylor series to approximate this relationship as

$$y_i = y_{ir} + \left.\frac{\partial y_i}{\partial x_{m1}}\right|_r (x_{m1} - x_{m1r}) + \left.\frac{\partial y_i}{\partial x_{m2}}\right|_r (x_{m2} - x_{m2r}) + \ldots + \left.\frac{\partial y_i}{\partial x_{d1}}\right|_r (x_{d1} - x_{d1r}) + \ldots \qquad (3\text{-}9)$$

or in deviation variables

$$y_i' = \left.\frac{\partial y_i}{\partial x_{m1}}\right|_r x_{m1}' + \left.\frac{\partial y_i}{\partial x_{m2}}\right|_r x_{m2}' + \ldots + \left.\frac{\partial y_i}{\partial x_{d1}}\right|_r x_{d1}' + \ldots \qquad (3\text{-}10)$$

When we apply this general form to each controlled variable, we can collect them into the standard linear system form

$$\underline{y}' = \underline{\underline{P}}_m \underline{x}_m' + \underline{\underline{P}}_d \underline{x}_d' \qquad (3\text{-}11)$$

where we see that matrices $\underline{\underline{P}}_m$ and $\underline{\underline{P}}_d$ are ensembles of the Taylor series partial derivatives, evaluated at the reference condition. Writing (3-11) specifically for the heat exchanger network,

$$\begin{bmatrix} T_6' \\ T_8' \\ T_9' \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial T_6}{\partial W_2}\right|_r & \left.\frac{\partial T_6}{\partial W_3}\right|_r \\ \left.\frac{\partial T_8}{\partial W_2}\right|_r & \left.\frac{\partial T_8}{\partial W_3}\right|_r \\ \left.\frac{\partial T_9}{\partial W_2}\right|_r & \left.\frac{\partial T_9}{\partial W_3}\right|_r \end{bmatrix} \begin{bmatrix} W_2' \\ W_3' \end{bmatrix} + \begin{bmatrix} \left.\frac{\partial T_6}{\partial W_1}\right|_r & \left.\frac{\partial T_6}{\partial T_1}\right|_r & \left.\frac{\partial T_6}{\partial T_2}\right|_r & \left.\frac{\partial T_6}{\partial T_3}\right|_r \\ \left.\frac{\partial T_8}{\partial W_1}\right|_r & \left.\frac{\partial T_8}{\partial T_1}\right|_r & \left.\frac{\partial T_8}{\partial T_2}\right|_r & \left.\frac{\partial T_8}{\partial T_3}\right|_r \\ \left.\frac{\partial T_9}{\partial W_1}\right|_r & \left.\frac{\partial T_9}{\partial T_1}\right|_r & \left.\frac{\partial T_9}{\partial T_2}\right|_r & \left.\frac{\partial T_9}{\partial T_3}\right|_r \end{bmatrix} \begin{bmatrix} W_1' \\ T_1' \\ T_2' \\ T_3' \end{bmatrix} \qquad (3\text{-}12)$$

Thus we assemble matrices $\underline{\underline{P}}_m$ and $\underline{\underline{P}}_d$ directly by calculating numerical derivatives with (3-8).

putting it together to calculate the matrices

For example, we calculate the root of nonlinear system $\underline{F}$ at $W_2 = W_{2r} + \delta W_2$, with all other input variables at reference conditions. Then we calculate the root for $W_2 = W_{2r} - \delta W_2$, leaving the other inputs unchanged. From these two steady-state conditions, we obtain the first column of matrix $\underline{\underline{P}}_m$. The other column is obtained similarly by finding roots at $W_3 = W_{3r} \pm \delta W_3$, leaving W_2 and the other inputs at the reference values. $\underline{\underline{P}}_d$ follows in a similar manner.

scaling variables using matrix notation

We express each vector of variables in terms of scaled variables, and then separate the scaling ranges into a separate matrix. For example,

$$\underline{x}_m^{'} = \begin{bmatrix} W_2^{'} \\ W_3^{'} \end{bmatrix} = \begin{bmatrix} W_2^{*'}\Delta W_2 \\ W_3^{*'}\Delta W_3 \end{bmatrix} = \begin{bmatrix} \Delta W_2 & 0 \\ 0 & \Delta W_3 \end{bmatrix} \begin{bmatrix} W_2^{*'} \\ W_3^{*'} \end{bmatrix} = \underline{\underline{S}}_m \underline{x}_m^{*'} \qquad (3\text{-}13)$$

Similar treatments are done for $\underline{y}^{'}$ and $\underline{x}_d^{'}$. Substituting into the linear model, we find

$$\underline{\underline{S}}\,\underline{y}^{*'} = \underline{\underline{P}}_m \underline{\underline{S}}_m \underline{x}_m^{*'} + \underline{\underline{P}}_d \underline{\underline{S}}_d \underline{x}_d^{*'} \qquad (3\text{-}14)$$

where the scaling ranges for individual controlled variables in $\underline{y}$ are collected into $\underline{\underline{S}}$, and so forth. We then isolate y*′; on performing the indicated matrix multiplication, we obtain matrices $\underline{\underline{P}}_m^*$ and $\underline{\underline{P}}_d^*$.

$$\begin{aligned} \underline{y}^{*'} &= \underline{\underline{S}}^{-1} \underline{\underline{P}}_m \underline{\underline{S}}_m \underline{x}_m^{*'} + \underline{\underline{S}}^{-1} \underline{\underline{P}}_d \underline{\underline{S}}_d \underline{x}_d^{*'} \\ &= \underline{\underline{P}}_m^{*} \underline{x}_m^{*'} + \underline{\underline{P}}_d^{*} \underline{x}_d^{*'} \end{aligned} \qquad (3\text{-}15)$$

In Matlab, we should use left division instead of the matrix inverse:

```
Pmstar = S\Pm*Sm;  % = inv(S)*Pm*Sm
Pdstar = S\Pd*Sd;  % = inv(S)*Pd*Sd
```

don’t get lost

If you find yourself feeling bewildered by these various operations, return to the side-by-side comparison above and make sure that you understand the overall procedure for turning a nonlinear model into a linear system approximation. Then review the tools again with the purpose of each step firmly in mind.

we create a Matlab program to perform these calculations

We base our calculations on Prof. Beers’ code reduced_Newton.m, which will calculate the vector of roots of a set of nonlinear algebraic equations. reduced_Newton requires that we write a routine called calc_f.m to contain our six equations; given a vector of CV and IV values, calc_f returns the vector computed from the six equations. If it returns zeroes, we know that the supplied vector was a root. We write the main routine to set the reference conditions, calculate matrices $\underline{\underline{P}}_m^*$ and $\underline{\underline{P}}_d^*$, and compute the RGA and DC.

From our shower work, we recall

$$\mathrm{RGA} = \underline{\underline{P}}_m \otimes \left(\underline{\underline{P}}_m^{-1}\right)^T \qquad \left(= \underline{\underline{P}}_m^{*} \otimes \left(\underline{\underline{P}}_m^{*\,-1}\right)^T\right) \qquad (3\text{-}16)$$

and

$$\mathrm{DC} = \mathrm{norm}\left(-\underline{\underline{P}}_m^{*\,-1} \underline{\underline{P}}_d^{*} \underline{x}_d^{*'}\right) \qquad (3\text{-}17)$$

we do an example to calculate RGA

Let us for convenience presume these are all unspecified organic liquid streams. Let all the heat capacities be simply 2500 J kg^{-1} K^{-1} and all the heat transfer coefficients 250 W m^{-2} K^{-1}. Our design has led us to three identical heat exchangers of 117 m^2. The inlet streams are

W_1: 8.2 kg s^{-1} T_1:250°C
W_2: 9.1 kg s^{-1} T_2:136°C
W_3: 5.6 kg s^{-1} T_3:100°C

We calculate the corresponding intermediate and output variables

T_4: 196.2°C T_6: 129.3°C
T_5: 159.8°C T_8: 223.4°C
T_7: 144.7°C T_9: 168.8°C

Matrices $\underline{\underline{P}}_m$ and $\underline{\underline{P}}_d$ are

```
Pm =
   -0.5998   -3.9815
   -0.2312   -9.3344
   -2.9267   -4.2319


Pd =
    6.0495    0.1114    0.3487    0.5399
    3.3765    0.7906    0.1344    0.0750
    5.7393    0.3142    0.6031    0.0827
```

where the units of the $\underline{\underline{P}}_m$ elements, and the first column of $\underline{\underline{P}}_d$ elements, are K s kg^{-1}. The remaining elements of $\underline{\underline{P}}_d$ are dimensionless.

Actually, our first controllability problem has been visible for some time: there are only two manipulated variables to deploy against three desired controlled variables. Hence we must let one of these remain uncontrolled. We thus calculate RGA from the appropriate submatrix of $\underline{\underline{P}}_m$ for three separate cases:

<u>control T_6 and T_8</u>

```
RGA =
    1.1968   -0.1968
   -0.1968    1.1968
```

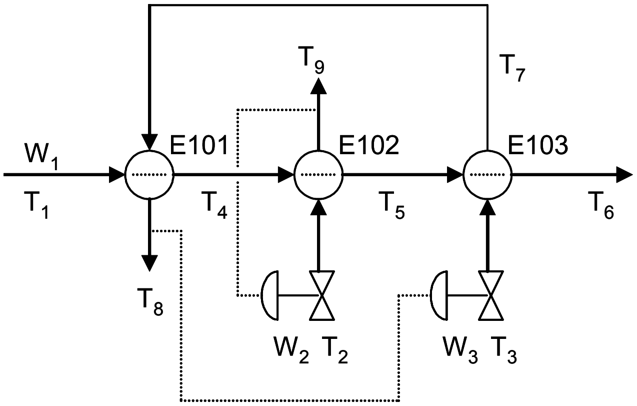
<u>control T_6 and T_9</u>

```
RGA =
   -0.2785    1.2785
    1.2785   -0.2785
```

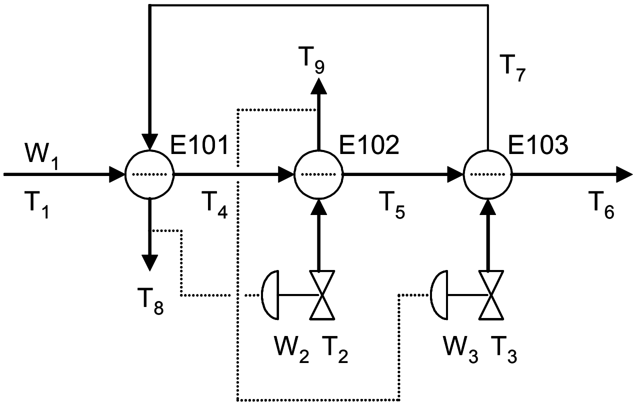
control T_8 and T_9

```
RGA =
   -0.0371    1.0371
    1.0371   -0.0371
```

The selection of which two variables to control may depend on other considerations – safety considerations, impact on the downstream processes, likelihood of disturbances affecting the various CV – but if there are no other compelling grounds, the RGA suggests that we pair T_8 with W_3 and T_9 with W_2. These two loops are predicted to show the least interaction; that is, adjustments in one loop should not strongly disturb the other.



Notice that each RGA shows (by negative elements) one pairing to be unstable. It would require dynamic simulation to explore this quantitatively, but we can see the outlines of the behavior by imagining the effects of a step disturbance in one input. For example, suppose that we pair T_8 with W_2 and T_9 with W_3.



NOT RECOMMENDED

We imagine an initial steady-state at the reference conditions, and we disturb this by a small step increase in temperature T_2. This will increase outlet temperature T_9. The controller tries to counter an increase in T_9 by increasing W_3, so that stream W_1 will approach HX102 cooler. Increasing W_3, however, also tends to reduce T_8. The T_8 controller tries to increase T_8 by heating the W_1 stream approaching HX103. It does this by decreasing W_2. Decreasing W_2, however, tends to increase outlet temperature T_9, thus reinforcing the initial disturbance. The likely outcome is maximum W_3 and minimum W_2, far from the desired reference condition.

we continue the example to compute DC

If we were designing this process, of course, we would find out about other considerations besides the RGA, as well as the information needed to set suitable scaling ranges for all our variables. Because our purpose is to learn screening tools, however, we are entitled to make some simplifying assumptions about these details. For example, we will assume that all temperatures are scaled to ±10ºC around the reference condition and flows are scaled between zero and twice the reference value. With these simple scaling ranges, we calculate the dimensionless matrices $\underline{\underline{P}}_m^*$ and $\underline{\underline{P}}_d^*$

```
Pmstar =
   -0.5458   -2.2296
   -0.2104   -5.2273
   -2.6633   -2.3699
```

```
Pdstar =
    4.9606    0.1114    0.3487    0.5399
    2.7687    0.7906    0.1344    0.0750
    4.7062    0.3142    0.6031    0.0827
```

For further calculations, we omit the first row of each matrix, because we have chosen to control T_8 and T_9, and their dependency is given by rows 2 and 3.

We calculate DC as a function of the disturbance vector $x_d^{*\prime}$. We check cases in which flow W_1 varies from 50% to 150% of its reference value. That is, $W_1^{*\prime}$ takes values of -0.25, 0, and 0.25. Each temperature varies a full ±10ºC, so that each scaled deviation variable takes values of -0.5, 0, and 0.5. This means 81 calculations of DC. We are concerned with the largest values:

```
    DC        W1*′      T1*′       T2*′      T3*′       W2*′       W3*′
  0.4706   -0.2500         0   -0.5000   -0.5000   -0.4510   -0.1343
  0.4706    0.2500         0    0.5000    0.5000    0.4510    0.1343
  0.4784   -0.2500   -0.5000   -0.5000         0   -0.4329   -0.2035
  0.4784    0.2500    0.5000    0.5000         0    0.4329    0.2035
  0.4898   -0.2500   -0.5000   -0.5000   -0.5000   -0.4424   -0.2103
  0.4898    0.2500    0.5000    0.5000    0.5000    0.4424    0.2103
```

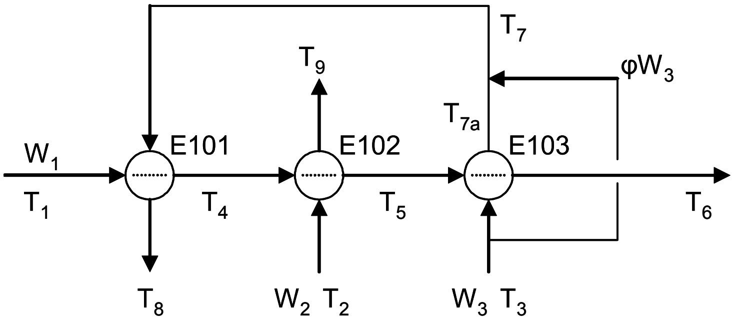
The maximum occurs symmetrically: when the flow and all temperatures decrease, or increase. Notice that both W2*′ and W3*′ are below their extreme values (±0.5) so that we have sufficient MV capacity to overcome the anticipated disturbances.

before we contemplated heat recovery, we would have controlled all three temperatures

We should not give up control too easily. The problem at present is that we have two MV for three CV. Here are two schemes for creating another MV:

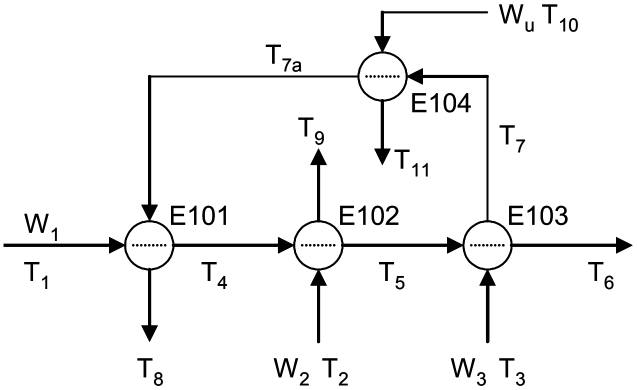
adjustable bypass stream

As in Seider et al (3), we divert some of stream W_3 past HX103. We now can manipulate W_2, W_3, and the bypass fraction φ, so we can control all three variables. However, because we are removing part of W_3 from being heated, we may require larger heat exchangers to produce the desired output temperatures. Thus our new operating capability comes at a capital cost.



trim heat exchanger

We do part of our heating with a new heat exchanger, using a plant utility heating fluid. We now can manipulate W_2, W_3, and the utility flow W_u, so we can control all three variables. Again, our new operating capability comes at a capital cost. In this case, however, we may be able to reduce the other heat exchanger sizes somewhat.



by now, we know the drill for analyzing these alternatives

The first step is to update the process model to reflect the process changes. For the bypass scheme, we must write M&E balances over the mixing junction. For the trim HX scheme, we need energy balances and a performance equation for the new heat exchanger.

We review our variable classification – CV is the same, but each scheme has a new MV and some new IV, as well. For simplicity (we have enough complication) we shall ignore any new DV that would be introduced, such as T_{10}.

We calculate new reference conditions – that is, a new design. Our original input and output specifications for the reference still apply, of course, but we must choose a new MV reference in each case. That specification will affect the other heat exchanger sizes. We will use the DC, calculated after the fact, to evaluate whether our design (bypass fraction, trim heat exchanger size) has the capability to overcome disturbances.

We rewrite f_calc.m and update the main program as needed to calculate the linear system matrices.

We calculate the RGA for each scheme to guide us in pairing MV and CV. We calculate DC, and may have to adapt the design if DC indicates that we cannot compensate for disturbances with the reference bypass flow, or trim HX size, that we chose.

how the story ends

What do you find?

references

(1) K. Beers, 10.34, ocw.mit.edu.
(2) T. McAvoy, Interaction Analysis, Instrument Society of America, 1983.
(3) W. Seider, J. Seader, and D. Lewin, Product and Process Design Principles, 2nd ed, Wylie, 2004.

nomenclature

C_p	liquid specific heat
W	mass flow rate
$\underline{\underline{P}}_d$	matrix of gain coefficients for the disturbance variables
$\underline{\underline{P}}_m$	matrix of gain coefficients for the manipulated variables
Q	heat duty of a heat exchanger
T	temperature
T_{in}	inlet temperature for flowing stream
T_{out}	outlet temperature for flowing stream
T_{ref}	thermodynamic reference temperature for enthalpy
U	overall coefficient of heat transfer for a heat exchanger
$\underline{x}_d$	vector of input variables into the system, the disturbance variables
$\underline{x}_m$	vector of input variables into the system, the manipulated variables
$\underline{y}$	vector of output variables from the system, the controlled variables

abbreviations

CV	"controlled variable", a system output that we wish to maintain at a set point value
DC	"disturbance cost"
DV	"disturbance variable", a system input that we have no influence over
MV	"manipulated variable", a system input that we may adjust for our purposes
RGA	"relative gain array"

subscripts

cold	colder stream in a heat exchanger
hot	hotter stream in a heat exchanger
r	a reference operating condition around which we derive a linear approximation

superscripts

$'$	indicates a deviation variable; i.e., the physical variable minus a reference value
*	indicates a scaled variable; i.e., variable has been divided by its operating range