Transformed inputs for linearization, decoupling and feedforward disturbance rejection

Sigurd Skogestad[a], Cristina Zotică[a], Nicholas Alsop[b]

[a]*Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491, Trondheim, Norway*
[b]*Senior Process Control Engineer, Borealis AB, Stenungsund, Sweden*

Abstract

This paper introduces powerful static input transformations which transform the the original system (process) into a transformed system which is easier to control. The transformed inputs v may be implemented in many ways and under many names, for example, as ratio, feedforward and decoupling control, and even as cascade control. All these methods are frequently used in industry, but are often introduced in an *ad-hoc* fashion. The present paper provides a systematic method for deriving such control strategies from a nonlinear process model. For a static model, the ideal transformed input v_0 is simply the right-hand-side f_0 of the model equations. With this choice the transformed system becomes $y = v_0$ at steady state, that is, it is linear, decoupled and independent of disturbances. For implementation of the transformed inputs, the model f_0 need to be inverted, and for this we may use either a model-based or a feedback-based inverse. The latter leads to the use of cascade control. The ideal transformed input derived from a dynamic model is a special case of feedback linearization. However, except for achieving linearization also dynamically, we find that the benefits of feedback linearization compared to using transformed inputs v_0 based on a static model are usually small.

Keywords: nonlinear process control, feedforward control, model-based control, cascade control

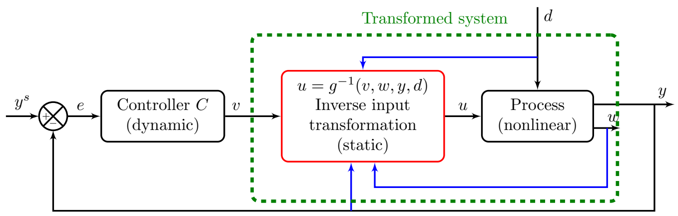


Figure 1: Use of transformed inputs v. For example, the transformed input could be the ratio $v = g(u, d) = \frac{u}{d}$, and the "inverse input transformation" block that inverts this relationship would then be $u = g^{-1}(v, d) = vd$.

1. Introduction

Industry frequently makes use of nonlinear static model-based "calculation blocks", "function blocks", or "ratio stations" to provide feedforward action, decoupling or linearization (adaptive gain), and Shinskey (1981) provides many examples of this. The main motivation for this work is to provide a better theoretical basis for these model-based nonlinear control elements, which we in this paper study in the context of nonlinear input transformations.

Let u denote the original (physical) input and let v denote the transformed input which depends on u and other variables. The main idea is that the controller C (or in some cases the operator) sets the value of the transformed input v rather than the physical input u, see Figure 1. In this paper, we define the transformed input v as a nonlinear static function g of the physical input u and other variables:

$$v = g(u, w, y, d) \tag{1}$$

Note that the specific function g is a design choice for the control engineer. The variables are defined as follows,

$v :=$ transformed inputs

$u :=$ physical inputs

$y :=$ controlled outputs

$d :=$ measured disturbances

$w :=$ other measured dependent variables (states).

perhaps we should emphasis that we do not required measurements of all states of the process? Reply from Sigurd: Isn't this clear from the examples below?

In this paper paper, we do not include dynamic elements in the definition of the transformed input v, although this is frequently done in industrial practice. Nevertheless, even without dynamics, Eq. 1 provides a very generic definition so let us state more clearly the objective of introducing the transformed input. *The transformed input v replaces the physical input u as the manipulated variable for control of the output y, with the aim of simplifying the control task by including elements such as decoupling, linearization and feedforward action.*

Shinskey (1981) (on page 119) writes in relation to selecting input and output variables for control:

> "There is no need to be limited to single measurable or manipulable variables. If a more meaningful variable happens to be a mathematical combination of two or more measurable or manipulable variables, there is no reason why it cannot be used."

Some simple examples of transformed inputs are

$$v = u + d \tag{2a}$$

$$v = \frac{u}{d} \tag{2b}$$

$$v = u_1 - u_2 \tag{2c}$$

$$v = \frac{u_1}{u_2} \tag{2d}$$

$$v = w \tag{2e}$$

Such transformed inputs are often introduced by engineers on simple physical grounds. The transformed input $v = u+d$ in Eq. 2a provides feedforward action from a measured disturbance d. It may used, for example, for a case where u and d represent two feedrates and we want to control the combined flowrate $u + d$. The ratio $v = \frac{u}{d}$ in Eq. 2b may provide feedforward action and linearization. It is typically used when u and d represent two feedrates and we want to control the quality (e.g. composition) of the combined feed. A transformed variable with two inputs may provide decoupling, for example the difference $v = u_1 - u_2$ in Eq. 2c and the ratio $v = \frac{u_1}{u_2}$ in Eq. 2d.

The transformed input $v = w$ in Eq. 2e with $w = F$ is probably the most common of all in process control, as it is used when u is the valve position and w is the corresponding measured flowrate F. This particular transformed variable ($v = F$) is so common that in many cases people consider the flowrate F to be the physical input u, and to simplify the treatment we will sometimes do this in this paper.

However, it is not enough to define the transformed input v, as in (1) and (2), we also need to generate from a given value of $v = g(u, w, y, d)$, the corresponding physical input u. Shinskey (1981) calls this "reversing the process model". There are two main ways of generating this inverse:

A. *Model-based inverse* [1] using the "inverse input transformation" $u = g^{-1}(v, w, y, d)$; see Figure 1.

B. *Feedback-based inverse* using a cascade implementation with a slave controller.

The model-based approach may be used for Eqs. 2a-2d above. For example, for $v = g(u, d) = u + d$ in Eq. 2a the inverse becomes

$$u = g^{-1}(v, d) = v - d \tag{3}$$

However, a model-based inverse is not possible for the transformed input $v = w$ in Eq. 2e because $g(w, y, d) = v$ does not depend explicitly on u. In this case, we must use a cascade implementation with a slave w (flow) controller that

[1] In this paper, the notation g^{-1} means that we invert or reverse the static function g between independent and dependent variables. For example, if the original function is $v = g(u, d)$ where u is the independent variable, then the solution that results from solving $v = g(u, d)$ with respect to u for a given v is written as $u = g^{-1}(v, d)$.

generates the physical input u (valve position) that keeps v at it's setpoint.

In most cases, the selection of transformed inputs v is based on simple static models, for example, from material or energy balances (Shinskey, 1981). However, the treatment of Shinskey is case-study based and in this paper, we aim to show how to select the transformed variables in a systematic manner. Ideally, assuming no model error and that we measure all disturbances, these "ideal transformed variables" gives a transformed system from v to y (see Figure 1) that is linear, decoupled and independent of disturbances.

We also show how this approach can be extended to dynamic models, and this case is closely related to the theory of feedback linearization (Isidori, 1995), which has a strong theoretical basis. For dynamic models, the transformed input v may depend on the controlled variable y, but even in such cases the main feedback from y is through the outer feedback controller C (see Figure 1). The outer feedback controller C has the aim of correcting for uncertainty, including for model error and unknown disturbances.

1.1. Previous academic work

There is hardly any academic literature on the common industrial approach of (Shinskey, 1981) of using static models to derive transformed inputs. On the other hand, as just mentioned, there is a large body of mathematical theory on variable transformations to transform nonlinear differential equations into linear differential equations, which has been applied in the control field. The most well-known approach is feedback linearization based on mathematical concepts from Lie algebra (e.g., Isidori (1995); Khalil (2015), Kravaris & Chung (1987)). As mentioned, this theory is closely related to the input transformations for dynamic systems studied in this paper. However, the theory of feedback linearization, although extensively taught in nonlinear control classes, is hardly ever used in industrial practice, at least within the field of process control. There are several reasons for this. One is that the mathematics are seemingly complicated. Another reason is that, mainly for reasons of mathematical generality and simplicity, Isidori (1995) selects the transformed inputs such that the resulting transformed linear system is integrating, $\frac{dy}{dt} = v$. This means that the transformed system is at the limit to instability, so the transformed inputs v cannot be kept constant. For example, with a fixed v, any unmeasured disturbance will result in an integrating output y. Therefore, Isidori (1995) introduces an outer state feedback controller as part of the solution. However, in many cases it is strongly desirable to be able fix v, at least on an intermediate time scale, and actually the transformation into an integrating system is not necessary. For example, Kravaris & Chung (1987) and (Bastin & Dochain, 1990), who study process control applications, use a formulation that gives a stable linear transformed system on the form $\frac{dy}{dt} = Ay + Bv$ (where the matrices A and B are tuning parameters), and this is the approach taken in this paper. In a personal communication, (Isidori, 2020) emphasizes that $A = 0$ was just chosen as an example, but this message has not made its way to the many potential users of the feedback linearization theory.

For our purposes, the advantage with the large body of literature on feedback linearization, is that this literature provides a mathematical basis for issues related to the invertibility and stability of the proposed transformations.

Throughout the paper we assume that we measure all the parameters that enter the transformations, such as disturbances d and internal variables (states) w. This is often not true, so in practice there are two alternatives. The most common is to keep only parts of the benefit of the input transformation, for example decoupling, and leave the disturbance rejection to the outer feedback controller C. The other approach is to use an estimator or observer (e.g., Kravaris & Chung (1987), (Bastin & Dochain, 1990)) to estimate the non-measured variables d and w. The issue of estimators is not discussed in this paper. It should be noted that the introduction of estimators, and along with it the issue of noise and model uncertainty, makes it very difficult to prove generally the mathematical properties of feedback linearization.

The paper starts with a motivating mixing example in Section 2. Next, in Section 3 we discuss in more detail the two main approaches for implementing the transformed inputs which are “exact” model inversion and inversion by feedback (cascade control). In Section 4, we provide control engineers with model-based tools for selecting "ideal" transformed inputs to provide linearization, decoupling and feedforward control. The model can either be a static model or a low-order dynamic model. In the dynamic case, the theory is closely related to the theory of feedback linearization. In Section 5 we present several case studies. In Section 6, we discuss the results and Section 7 we make our final remarks.

We have attempted to keep the mathematical treatment at a quite low level, so that the paper will be readable also for an industrial audience. One reason is that we strongly believe that the results in this paper can be very useful in industrial practice.

2. Motivating case study: Decoupling of mixing process

The main reason for introducing transformed inputs v is to simplify the control of the outputs y. In section 4, we introduce systematic methods for selecting "ideal" transformed input v that provide linearization, decoupling and disturbance rejection. However, in many cases, engineers use simpler transformed inputs (Shinskey, 1981) that do not provide all these features. In this section, we consider a simple motivating case study.

2.1. Example 1: Mixing process with flowrates as physical inputs

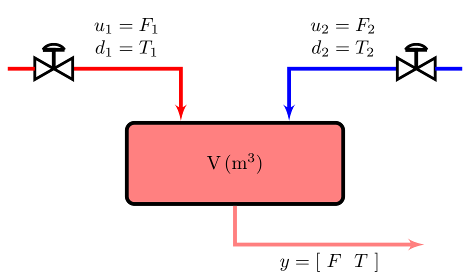


Figure 2: Flowsheet of mixing process

The mixing process in Figure 2 has two inlet streams, and initially we consider for simplicity the two flowrates F_1 and F_2 [kg s^{-1}] as the physical inputs,

(a)) Traditional design with two separate handles ($u_1 = F_1$, $u_2 = F_2$)

(b) Modern one-handle design where decoupling is achieved by the physical design ($v_1 = F_1 + F_2$ and $v_r = \frac{F_1}{F_2}$).

Figure 3: Valve (faucet) designs for mixing hot and cold water for homes.

rather than the valve positions. The inlet flows are mixed to get a given total flow (F) and quality (T), which we want to control. Depending on the application, T could represent temperature or composition. We will consider the mixing of hot (F_1) and cold (F_2) water where T is temperature. The main disturbances are the two inlet temperatures. Thus, we have

$$u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}; \quad y = \begin{bmatrix} F \\ T \end{bmatrix}; \quad d = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$$

A real design of this process using a traditional faucet with two separate handles (valves) is shown in Figure 3a. We know that this process is quite interactive. For example, to increase the temperature $y_2 = T$ while keeping a constant total flow $y_1 = F$, we need to increase the input $u_1 = F_1$ (hot water) while reducing $u_2 = F_2$ by the same amount.

Transformation for decoupling

To eliminate the interactions and make the process decoupled, we may use the alternative one-handle faucet in Figure 3b. Here, one direction of the handle (usually up-down) is used for adjusting the total flow ($F = F_1 + F_2$), and the other direction (usually left-right) is used for adjusting the temperature by changing the ratio $\frac{F_1}{F_2}$ of hot and cold water. This corresponds to using the transformed inputs $v_1 = F_1 + F_2$ and $v_r = \frac{F_1}{F_2}$. The fact that these transformed inputs give decoupling is probably clear on physical grounds, and it can easily be proven by making use of the mass and energy balances as shown later in Example 5 (see Section 5.3).

For the modern one-handle design in Figure 3b, the transformed variables v_1 and v_r are implemented physically. However, to implement decoupling using the traditional two-handle design in Figure 3a, we need to add in the control scheme a decoupling block to compute the physical inputs (u) from the transformed inputs (v). With this digital rather than physical decoupler, we may use the opportunity to replace the flow ratio $v_r = \frac{F_1}{F_2}$ by the alternative ratio $v_2 = \frac{F_1}{F_1+F_2}$. The transformed inputs $v = g(u)$ then become

$$v_1 = F_1 + F_2 = u_1 + u_2 \tag{4a}$$

$$v_2 = \frac{F_1}{F_1 + F_2} = \frac{u_1}{u_1 + u_2} \tag{4b}$$

Both ratios v_r and v_2 give decoupling and they are equivalent in the sense that fixing one keeps the other constant (since $v_2 = \frac{v_r}{v_r+1}$). However, the alternative ratio v_2 in (4b) has some properties that makes it better for implementation. First, it avoids division by zero when $F_2 = 0$. Second, v_2 is always in the range 0 to 1, whereas v_r may vary between 0 and ∞. Third, as shown later in Example 5, the ratio v_2 is a special case of the ideal static transformed input (v_0) and provides linearization. Fourth, the expression for the inverse in the decoupling block becomes very simple with v_2, see (5).

To find the inverse $u = g^{-1}(v)$ (decoupling block), we solve the expression for the transformed input in Eq. 4 with respect to u for a given v, to derive

$$u_1 = F_1 = v_1 v_2 \tag{5a}$$

$$u_2 = F_2 = v_1 - v_1 v_2 \tag{5b}$$

The nonlinear decoupling g^{-1} in Eq. 5 may be implemented in the block "inverse input transformation" in Figure 1 to provide a decoupled response from v to y for the traditional two-handle design. In this simple case, the equations can easily be implemented using standard multiplication and subtraction elements.

In Example 5 (see Section 5.3), we will use a systematic procedure to derive a generalized (ideal) version of the transformed inputs in Eq. 4, where disturbance rejection and linearization are also included.

2.2. Example 2: Mixing process with valve positions as physical inputs

We have so far assumed that the flow rates F_1 and F_2 are the physical inputs, but in practice it is more likely that the valve positions z_1 and z_2 are the physical inputs and that the flowrates are possible extra measured variables w:

$$u = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}; \quad y = \begin{bmatrix} F \\ T \end{bmatrix}; \quad d = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}; \quad w = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

We will now consider three ways of implementing this in order to retain a decoupled transformed system from the transformed input v to the output y.

Alternative A (purely model-based inversion). The first option does not make use of any measured flows (w). It is based on inverting the entire model, including the valve model. The general block diagram is shown in Figure 1, and the corresponding flowsheet for this particular example is shown in Figure 4. A typical valve equation is

$$F = k f_v(z)\sqrt{\Delta P} \tag{6}$$

Here, F is the flow, z is the valve position, k is the valve constant and ΔP is pressure drop over the valve which is assumed to be a measured disturbance. The valve characteristic $f_v(z)$ is assumed to be known. For a linear valve, we have $f_v(z) = z$. Inverting the valve equation gives

$$z = f_v^{-1}\left(\frac{F}{k\sqrt{\Delta P}}\right) \tag{7}$$

The purely mode-based inversion may then be implemented as shown in Fig-

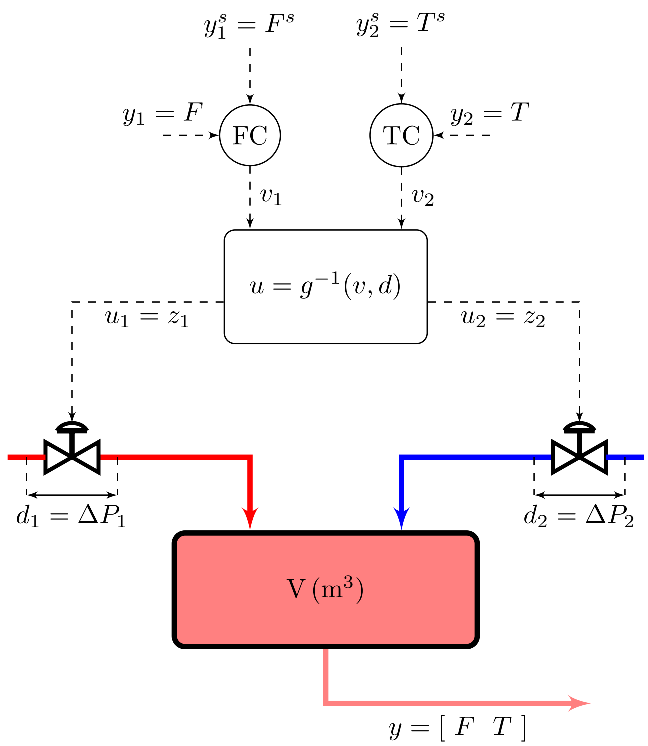


Figure 4: Alternative A for implementation of transformed inputs $v_1 = F_1 + F_2$ and $v_2 = \frac{F_1}{F_1+F_2}$ for the mixing process when the physical inputs u are the valve positions z. The inversion block $u = g^{-1}(v, d)$ is a combination of the inversions in Eq. 5 and Eq. 7.

Write v1=F1+F2 and v2 = F1/F1+F2 on Figure

ure 4, where the inversion block $g^{-1}(v,d)$ computes the valve positions $u = z$ (physical input) by combining the inversions in Eq. 5 and Eq. 7.

However, in practice, this implementation (A) may not work well, mainly because of uncertainty (error) in the valve characteristic $f_v(z)$ and the valve constant k, but also because of incorrect measurements of the pressure drop disturbances (ΔP). Therefore, if the inlet flows (F_1, F_2) can be measured with reasonable accuracy, then an implementation with slave flow controllers (Figure 5) is preferred, both because it is simpler and because the feedback controller accounts for the uncertainty in the model and in measurements. This implementation (C) is discussed next. Note that the naming of the alternatives (A, B, C) is the same as used later in the general treatment in Section 3.

Alternative C (combined model- and feedback-based inversion). The best option for this example is to use the model-based nonlinear decoupling in (5) to compute the desired flowrates (F_1^s and F_2^s) and combine this with two slave flow (w) controllers (FC1 and FC2), as shown in the flowsheet in Figure 5.

Note that the process as seen from the slave flow controllers (FC1 and FC2) is nonlinear. However, usually the valve dynamics from $u = z$ to $w = F$ are fast and with small couplings, so it is possible to design two fast slave flow controllers such we have almost perfect control ($w = w^s$), at least on the slower time scale relevant for the outer controllers (TC and FC). Note that the slave flow controllers, through the action of feedback, indirectly generate the inverse in Eq. 7.

Alternative B (purely feedback-based inversion). The third option (Figure 6) uses only feedback for the inversion. It also makes use of the measured w-variables (flows), but here the slave w-controllers are replaced by slave v-controllers (VC1 and VC2). This avoids the inverse block g^{-1}, so instead the decoupling from the transformed input v to y is taken care of by the v-controller, which controls the total flow (v_1) and the ratio (v_2) to given setpoints.

However, note we are using two single-loop v-controllers (VC1 and VC2) for a strongly coupled nonlinear process (from u to v). One may then question if there is any benefit compared to the simplest scheme with no input transfor-

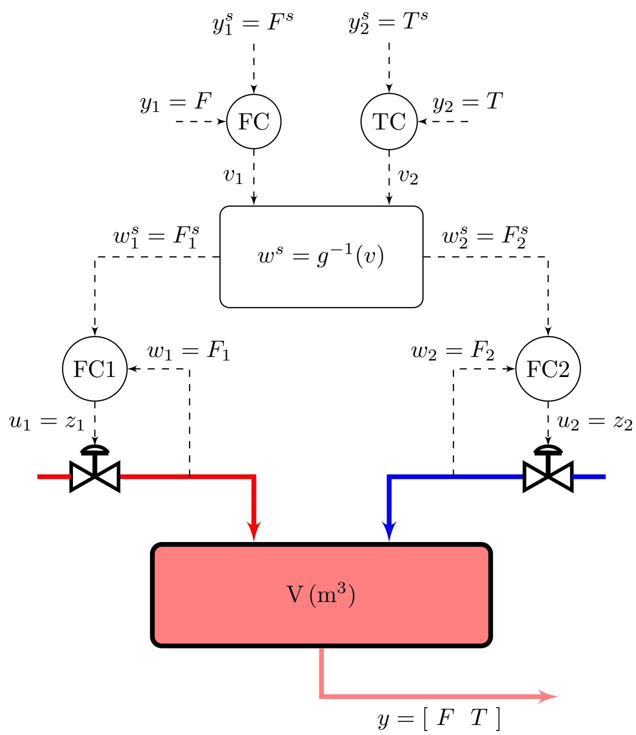


Figure 5: Alternative C for implementation of transformed inputs $v_1 = F_1 + F_2$ and $v_2 = \frac{F_1}{F_1+F_2}$ for the mixing process when the physical inputs u are the valve positions z. The nonlinear decoupling block $g^{-1}(v)$ is given in Eq. 5.

Inputs to white box: v1=F1+F2 and v2 = F1/F1+F2. Outputs from white box: F1s = v1 v2, F2s = v1 -v1v2

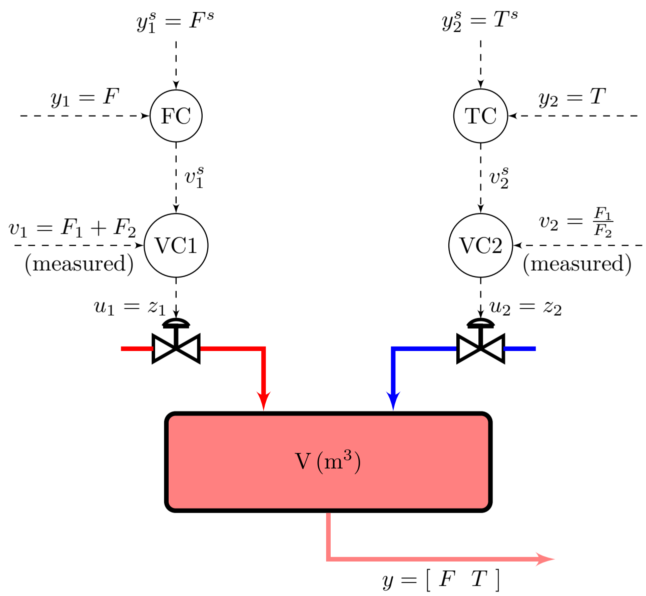


Figure 6: Alternative B (purely feedback inversion) for implementation of transformed inputs $v_1 = F_1 + F_2$ and $v_2 = \frac{F_1}{F_1+F_2}$ for the mixing process when the physical inputs u are the valve positions z.

change to v2 = F1/F1+F2 on Figure

mation (not shown in any Figure), that is, letting the outer controllers (TC and FC) manipulate directly the physical inputs (u), which are the valve positions. The answer is that there can indeed be a significant benefit by using input transformations with v-controllers (Alternative B) if there are effective delays associated with the control of y (e.g., measurement delays for F and T in our case), such that slave v-controllers can be significantly faster than outer controllers (TC and FC).

Nevertheless, it is clear that in this case Alternative C with the w (flow) controllers (Figure 5) is the best option because the interactions are much less than for the v-controllers in Figure 6.

3. Implementation of transformed inputs

This section generalizes the three alternative inverse implementations (A, B, C) from the motivating Example 2.

The transformed input v is defined as a nonlinear static function g that depends on the original (physical) input u and other measured variables:

$$v = g(u, w, y, d) \tag{8}$$

All variables may be vectors. For the multivariable case, we will assume that we have an equal number (n) of inputs u, outputs y and transformed inputs v. Often the function g is independent of y and in many cases we do not have extra measurements w. Note that g may not depend explicitly on u, but it should then depend indirectly on u through the measured variables w.

As mentioned in the introduction, the idea is that the outer controller C or the operator will set the value or the setpoint of the transformed input v. However, to implement v on the real process, we need to generate the corresponding physical input u. There are two main approaches for implementing the physical input u:

A. Model-based implementation, see Figures 1 and 7a. This gives exactly

$v = v^s$ (assuming all variables are measured perfectly and there is no model error).

B. Feedback-based implementation. see Figure 7b. With integral action in the slave controller C_v this gives $v = v^s$ after a dynamic transient.

We also discuss a third implementation (Figure 7c) which is a combination of the two. The three alternatives are the same as the ones presented in Figures 4, 6 and 5 for Example 2, respectively.

3.1. Alternative A: Model-based inversion (Figure 1 and Figure 7a)

The first approach is to invert the input transformation $v = g(u, w, y, d)$ in Eq. 8, by analytically or numerically finding the input u that corresponds to given values of v, w, y and d. We can formally write the solution as

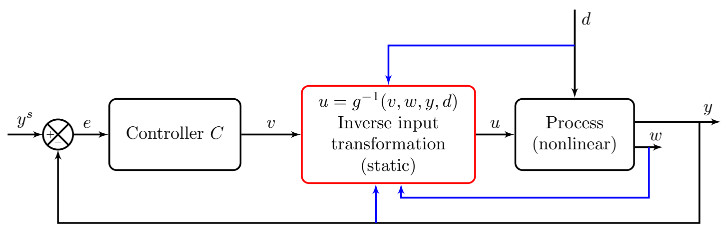
$$u = g^{-1}(v, w, y, d) \tag{9}$$

This gives the exact inverse $g^{-1}(v, w, y, d)$ if the inverse exists, if there is no model uncertainty and if all variables w, y and d are measured perfectly.

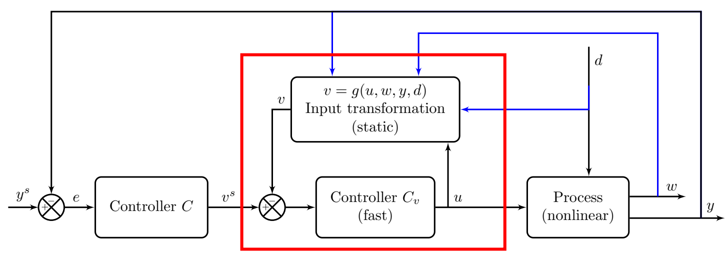
3.2. Alternative B: Feedback inversion with slave v-controller (cascade control) (Figure 7b)

A dynamic approximation of the inverse input transformation may be generated using an inner (slave) feedback controller C_v as shown in Figure 7b. Here, we compute the actual value $v = g(u, w, y, d)$ from measurements of u, w, y and d, and use the inner controller C_v to dynamically generate the input u that makes v approach the desired value v^s. Since $v = v^s$ at steady-state, the nonlinearity in the responses from u to v is effectively removed by the action of the feedback controller C_v.

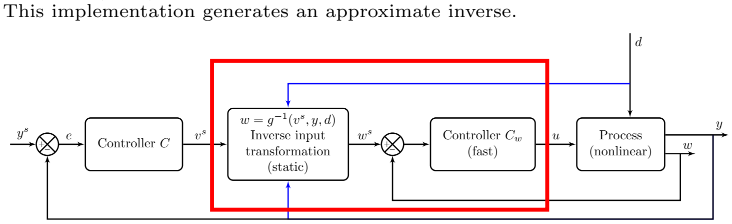
For tuning the controller C_v, it should be noted that the response from u to v usually has a large direct (static) effect, and for static processes pure I-controllers are generally sufficient, even for nonlinear processes. Therefore, the



(a) Model-based implementation A of transformed input $v = g(u, w, y, d)$. The physical input $u = g^{-1}(v, w, y, d)$ is generated by a static (algebraic) calculation block which inverts the transformed input model equations. The model-based implementation generates the exact inverse for the case with no model error.



(b) Feedback implementation B of transformed input $v = g(u, w, y, d)$ using cascaded v-controller. The computed value of v is driven to its setpoint v_s by the inner (slave) feedback controller C_v which generates the physical input u. This implementation generates an approximate inverse.



(c) Combined model-based and feedback implementation C of transformed input $v = g(w, y, d)$ using slave w-controller. Commonly, C_w is a flow controller (w = flowrate) and u is the valve position. This implementation generates an approximate inverse.

Figure 7: Alternative implementations for inverting input transformation $v = g(u, w, y, d)$. C, C_v and C_w are usually single-loop PID controllers

inner controller C_v in Figure 7b is often simply a linear integrating controller

$$u(t) = u_0 + K_I \int_{t_0}^{t} (v^s(t) - v(t))dt \tag{10}$$

where u_0 is the bias and the integral gain K_I is a tuning parameter. The integral action will make $v = v^s$ at steady state (as time goes to infinity) and a larger value of K_I will make $v(t)$ approach v^s faster. More generally, one may tune linear PID-controllers using the SIMC rules (Skogestad, 2003) based on the experimental response from u to v. For the $n \times n$ multivariable case, one usually designs n single-loop linear controllers for C_v, although it is possible to use multivariable control.

The feedback implementation (alternative B) in Figure 7b is required if v does not explicitly depend on u, or if there are unstable zero dynamics (inverse response in the scalar case) in the response from u to v. In other cases, the feedback implementation may be used as a numerical solution "trick" for generating an approximate inverse. In the control literature, this trick is often referred to as "dynamic inversion" (Lee et al., 2016). The reason for using this trick could be to avoid the complexity of deriving the inverse in Eq. 9 (see Example 7) or to avoid problems with singularities (Lee et al., 2016).

To sum up, although the process response from u to v may be nonlinear and interactive, the use of linear single-loop controllers will provide (almost) the desired decoupled and linear response of the transformed system from v to y, provided the inner loop with C_v can be made sufficiently fast.

3.3. Alternative C: Combined feedback- and model-based inversion with slave w-controller (Figure 7c)

This implementation is of particular interest for the case when $v = g(u, w, y, d)$ does not depend explicitly on u and there is only one measured w-variable associated with each input u. In addition to the inner controller C_w for w, we also need a block that inverts the transformation g with respect to w, that is, which computes the setpoint $w^s = g^{-1}(v, y, d)$.

This cascade implementation C (Figure 7c) is less general than the cascade implementation B in Figure 7b, because it assumes that the function $v = g(u, w, y, d)$ can be inverted to generate $w = g^{-1}(v^s, y, d)$. On the other hand, it has the advantage that we can include some model-based inversion, which may contribute to linearization, feedforward and decoupling. It also has the advantage that the inner controller C_w controls a physical measurement w, whereas v in Figure 7b is usually not a physical variable. The inner controller C_w may be tuned in a similar way as C_v, based on an experimental response from u to w.

A very common example is when C_w is a flow controller, that is, when w is a flow (F) and u is the corresponding valve position (z). Another common example is when C_w is a power or temperature controller, that is, when is w is a temperature or power (Q) and u is a valve position (z). In both these cases, we may have a model for the relationship from u to w, which we could have inverted and used in a model-based implementation A (Fig. 7a), but instead we prefer to use feedback control based on a measurement of w to invert the relationship, either because it is simpler or because it is more accurate. Of course, this assumes that we can use relatively high gain in the slave controller C_w such that the time constant for the slave loop is much smaller (typically by a factor 10 or more) than the time constant of the outer loop involving C.

Looking back at the mixing tank process, we concluded that Alternative C was the best. It includes model-based decoupling, which makes it much easier to tune the w (flow) controllers for Alternative C (Figure 5) than the v (sum and ratio) controllers for Alternative B (Figure 6).

3.4. Comparison of the three alternative implementations in Figure 7

The red blocks in the three block diagrams in Figure 7 perform the same task of generating the physical input u from a given value of transformed input v, but there are some important differences. First, the transformed input v is replaced by its desired value v^s (setpoint) in the two feedback implementations (B, C), because the feedback implementations do not give the exact inverse during dynamic transients. Second, and which is less clear from the Figure 7, there are often differences in the variables involved. In particular, the use of measured w-variables in the two feedback implementations may replace some process model equations and disturbance variables (d) in the exact model-based implementation in Figure 7a. Indeed, this was the case in the Motivating mixing example where we introduced flow measurements as w-variables for Alternatives B and C.

4. Derivation of ideal transformed inputs

Input transformations are in common use and as illustrated in the motivating mixing example they may be very useful. However, the main question we want to answer in this paper is:

> *How do we derive good input transformations in a systematic manner?*

Starting from a static or dynamic process model, we show in this section how to derive ideal transformed inputs which ideally achieve linearization, decoupling and disturbance rejection. We assume that we have a $n \times n$ control problem with n inputs u and n outputs y, and we want to use the model equations to find n transformed inputs v. The case with a static model is discussed in Section 4.1 and a dynamic model in Section 4.2. Note that we may combine static and dynamic models as shown in Example 5 in Section 5.3. In Section 4.3, we discuss that it may be convenient in many cases to write the model in terms of extra measured state variables w.

4.1. Obtaining ideal transformed system from a static process model

In the industrial literature, Shinskey (1981) shows by examples how to use static process models to derive nonlinear feedforward and decoupling blocks which are similar to the input transformations derived below. However, and very surprisingly, for the simple and important case of static systems, there seems to be no academic literature on how to do derive static feedforward and decoupling blocks in a systematic manner. Possibly this is because the derivation is almost trivial, as shown in the next few lines.

Consider a static process model with n independent equations written in the following general form

$$0 = f(u, y, d) \tag{11}$$

or even more generally as $n + n_x$ equations in the form

$$0 = f_x(u, x, y, d) \tag{12}$$

where x represents additional internal variables (states). In the more general case in Eq. 12, we assume that we can use the n_x extra equations to eliminate the internal variables x to get a model (at least formally) as given in Eq. 11.

Since the model equations in Eq. 11 are assumed to be independent, they may be solved with respect to y (at least formally) to get the static model on the form $y = f_0(u, d)$. We then have the following general result.

Ideal transformed variable based on static model. *Consider a static nonlinear model in the form*

$$y = f_0(u, d) \tag{13}$$

Define from this the ideal static transformed input

$$v_0 = \underbrace{B_0^{-1} f_0(u, d)}_{g(u,d)} \tag{14}$$

In Eq. 14, the matrix B_0 is free to choose and we usually choose

$$B_0 = I \tag{15}$$

Assume that v_0 can be exactly implemented by solving Eq. 14 with respect to u

to get the ideal input

$$u = g^{-1}(v_0, d) \tag{16}$$

Then, assuming that the real system is static with model $f_0(u,d)$ (no model error) and that we have perfect measurement of d, the transformed system becomes

$$y = B_0 v_0 \tag{17}$$

The transformed system in Eq. 17 is linear and independent of disturbances, and for the multivariable case it is also decoupled if we select B_0 to be a diagonal matrix.

Proof. The proof is trivial. From Eq. 14 we get $f_0(u,d) = B_0 v_0$ and substituting this into Eq. 13 gives $y = B_0 v_0$ in Eq. 17. The assumptions related to (16) are necessary to be able to generate the corresponding ideal input u. □

Note that we use the subscript 0 to show that v_0 is an ideal transformed input derived from a static model.

Note that it may not be necessary to explicitly derive the expression for $f_0(u,d)$ in Eq. 13. Rather, since the objective is to find the ideal input $u = g^{-1}(v_0, d)$ that gives the transformed system $y = B_0 v_0$ in Eq. 17, it may be simpler to stay with the original model equations in Eq. 11 or Eq. 12, and solve these with respect to u for a given value of $y = B_0 v_0$ to obtain $u = g^{-1}(v_0, d)$. This solution can be done either analytically or numerically, but a numerical solution is usually necessary for complicated models, like for the heat exchanger example discussed later and in (Zotică et al., 2020).

4.1.1. Choice of the tuning parameter B_0

The choice of B_0 is not critical, as it can be compensated by changing the gain of the outer controller C. We usually choose $B_0 = I$ such that the ideal transformed input is $v_0 = f_0(u,d)$. Since this gives $y = v_0$ at steady state, it may be tempting to think of the transformed input $v = v_0$ as the setpoint for the output y, but this is misleading because we usually have an outer feedback controller C which has the "true" setpoint y^s as one of its inputs, whereas v_0 is the output from C (see Figure 1). Thus, it is better to think of v_0 as the transformed process input, or possibly as a modified setpoint $y^{s'}$ (Bastin & Dochain, 1990).

4.2. Obtaining ideal transformed input from a dynamic process model

We next examine the case where we have a dynamic process model as given in Eq. 18. The derivation of the resulting ideal transformed input v_A is closely related to the theory of feedback linearization.

Ideal transformed variable based on dynamic model. *Consider a nonlinear dynamic model in the form*

$$\frac{dy}{dt} = f(u, y, d) \tag{18}$$

For the model in Eq. 18, the ideal transformed input is

$$v_A = \underbrace{B^{-1}(f(u, y, d) - Ay)}_{g(u,y,d)} \tag{19}$$

Here, the matrices A and B are tuning parameters. Assume that v can be exactly implemented by solving Eq. 19 with respect to u to get

$$u = g^{-1}(v_A, y, d) \tag{20}$$

Then assuming no uncertainty (no model error for $f(u, y, d)$ and perfect measurements (of d and y) the transformed system becomes

$$\frac{dy}{dt} = Ay + Bv_A \tag{21}$$

The transformed system in Eq. 21 is linear and independent of disturbances, and for the multivariable $(n \times n)$ case, it is also decoupled if we select A and B to be diagonal matrices.

Proof. Substituting the transformed input in Eq. 19 into Eq. 18 gives Eq. 21. Note that we have assumed that we can generate from the transformed input v_A the exact corresponding physical input u. $\square$

Note that we use the subscript A to show that v_A is an ideal transformed input derived based on a dynamic model and with a tuning parameter A.

It may seem that (18) represents a large class of dynamic models, but actually it is quite restrictive since we must assume that the number of differential equations (states) is equal to the number of inputs and outputs in the vectors u and y. In particular, we assume that the input u directly affects the time derivative of y $\left(\frac{dy}{dt}\right)$ of the controlled output y, which means that the relative order of the process system is assumed to be 1. Specifically, for the scalar case $(n = 1)$, we assume that we can write the model for y using only one scalar differential equation (18). Thus, for the scalar case we are restricted to a first-order system. However, if we allow the function f to depend on additional measured states w, then the class of systems is significantly larger. This is discussed in more detail later.

To guarantee invertibility in (20) , it is possible to restrict the class of models to guarantee that we always have a solution, as is done in the literature on exact linearization. In particular, in this literature it is assumed that the model is linear in the input u, that is, that we can write the right-hand side of Eq. 18 as shown in Khalil (2015) (p. 293).

$$f(u, y, d) = f_1(y, d) + f_2(y, d)\ u \tag{22}$$

where the functions f_1 and f_2 must satisfy certain smoothness conditions. Interestingly, many process models are linear in the flows, so if we make use of inner flow controllers then many process models satisfy Eq. 22. Nevertheless, we do not make this assumption in this paper, so the invertibility may need to be studied separately for each application.

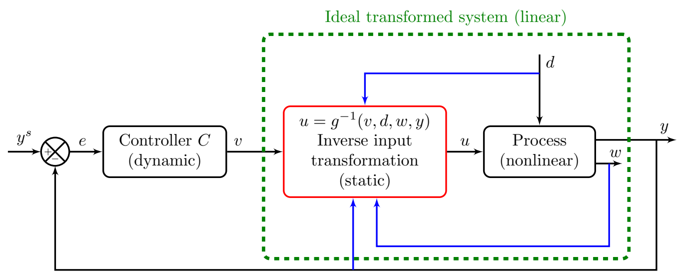


Figure 8: Block diagram where ideally (with no uncertainty) the transformed system is linear, decoupled and independent of disturbances. To achieve this and also get $y = v$ at steady state, use $v = v_0$ in Eq. **??** for a static process or $v = v_A$ in Eq. 24a with A diagonal for a dynamic process.

4.2.1. Choice of tuning parameter B

To get dynamic decoupling in Eq. 21 for the multivariable case, we need to select both matrices B and A to be diagonal. Dynamic decoupling is desirable because the optimal outer controller C is then diagonal (single-loop controllers). Otherwise, the choice of B is not critical as it may be compensated by changing the gain in the feedback controller C. To keep the initial (high-frequency) gain from v_i to y_i equal to that of the original system (from u_i to y_i) one may choose $B = \text{diag}(\tilde{B}) = \text{diag}(\partial f/\partial u)_*$ where the differentiation is performed at the nominal operating point *. However, in most of the examples in this paper we select

$$B = -A \tag{23}$$

because this gives $y = v_A$ at steady state (where $\frac{dy}{dt} = 0$). (Interestingly, since $y = Iv_A$ at steady state where I is the identity matrix, the choice $B = -A$ gives decoupling at steady state even if A and B are not diagonal.) With the choice $B = -A$, the transformed input and corresponding transformed system become

$$v_A = -A^{-1} f(u, y, d) + y \tag{24a}$$

$$\frac{dy}{dt} = A(y - v_A) \tag{24b}$$

Equivalently, one may introduce the time constant matrix of the transformed system,

$$\mathcal{T}_A = -A^{-1} \tag{25}$$

and the transformed input and corresponding transformed system may be written as

$$v_A = \mathcal{T}_A f(u, y, d) + y \tag{26a}$$

$$\mathcal{T}_A \frac{dy}{dt} + y = v_A \tag{26b}$$

4.2.2. Choice of tuning parameter A

The choice of the parameter A (or equivalently of $\mathcal{T}_A = -A^{-1}$) is important as it determines the dynamics of the transformed system. However, the importance should not be overemphasized, since we can change the closed-loop dynamics by design of the outer controller C. Note that we must choose $A < 0$ for the transformed system to be stable. We discuss below three choices for the tuning parameter A.

1. Keep the original dynamics, $A = \tilde{A}$. In most cases we propose selecting

$$A = \tilde{A} \equiv \left(\frac{\partial f}{\partial y}\right)_* \tag{27}$$

where the derivative is evaluated at the nominal point $*$ of operation. This makes the dynamics of the transformed system equal to the linearized dynamics of the original system. This choice also minimizes the effect of the measurements y on the transformed variables v_A (See Appendix). This seems reasonable because the outer controller C in any cases makes use of the measurements y.

2. Make the transformed system faster: $|A| > |\tilde{A}|$. To speed up the response from v to y, one may use larger magnitudes for the elements in A than that resulting from Eq. 27. However, note that the presence of a time delay in the measurement of y (or other dynamics that result in an effective delay) may give instability if we choose the elements in A too large in magnitude. Alternatively, note that it is always possible to select $A = \tilde{A}$ as in Eq. 27 and instead "speed up" the response with the outer controller C, which can be designed based on the experimental response from v_A to y and for which established robust design methods are available, for example, the SIMC PID-rules Skogestad (2003).

3. Make the system integrating: $A = 0$. The choice $A = 0$ is recommended in the standard feedback linearization literature (Isidori, 1995). This results in an integrating transformed system, $\frac{dy}{dt} = Bv$, where usually one selects B=I. However, except for cases where the original system is unstable or close to integrating, the choice $A = 0$ is not recommended. There are two reasons for this. The main reason is that with $A = 0$ the transformed system will not go to steady state, without the outer controller C. In particular, any unmeasured disturbances will cause the output y to change in a ramplike fashion and drift away from its desired steady state. This drifting will only stop when the input u reaches its physical maximum or minimum constraint. This is very undesirable, because usually one wants to be able to operate the transformed system without the outer controller C. The second reason for not selecting $A = 0$ is that there is a performance loss because generally we want to have integral action in the outer controller C to correct for uncertainty. With $A = 0$, the integrator in the transformed system poses performance limitations with a PI-controller, in particular for disturbance rejection (e.g., (Skogestad, 2003)). This performance limitation is not considered in the feedback linearization literature because they assume state feedback, that is, they assume C is a P-controller.

4.3. Model and transformed input in terms of measured state variables w

To derive ideal transformed variables we have assumed that we for the static case have a model $y = f_0(u, d)$ as given in (13) and for the dynamic case have a model $\frac{dy}{dt} = f(u, y, d)$ as given in (18). Importantly, the derived expressions for the ideal transformed inputs (v_0 and v_A) hold also when we include additional measured dependent variables (states, outputs) w in the expressions for f_0 and

f, that is, if we consider static models in the form

$$y = f_{0,w}(u, w, d) \tag{28}$$

and dynamic models in the form

$$\frac{dy}{dt} = f_w(u, w, y, d) \tag{29}$$

This allows us to use simpler models, because we don't need a model for w in (28) or (29). Thus, we are essentially replacing a model equation (for w) by a measurement (of w).

In the dynamic case, we have the additional advantage that the class of dynamic systems we can handle becomes much larger. To see this, note that we in our derivation of ideal transformed variables v_A for the dynamic case, assumed that the model can be written in the form $\frac{dy}{dt} = f(u, y, d)$, which means that the dynamic model order n (no. of differential equations) must be equal to the number of inputs u and outputs y. Specifically, for a scalar system ($n = 1$), we can only have one differential equation. However, in general we may have a high-order dynamic model in the form

$$\frac{dx}{dt} = f_x(u, x, d) \tag{30}$$

where the state vector x consists of the outputs y, the extra measured states w, and the remaining unmeasured states x_u. They key assumption is now that we measure all the states w that enter into the differential equation for y, that is, we assume that we can write the model for the outputs y in the form $\frac{dy}{dt} = f(u, w, y, d)$ as given in Eq. 29 where w, y and d are measured. Note that this allows for a high-order dynamic response from the input u to the output y. For example, we may have $\frac{dw}{dt} = f_w(u, w, d)$ (or the dynamic may be even higher order with additional unmeasured states x_u), but these dynamics for w will not matter for evaluating v_A.

In conclusion, when we extend the class of models to include w-variables, as given in Eq. 28 and Eq. 29, then from the above derivations, the expressions for the ideal transformed inputs remain the same, except that we must add w in the argument list. Thus, the ideal transformed input for the static model Eq. 28 becomes

$$v_0 = \underbrace{B_0^{-1} f_{0,w}(u, w, d)}_{g(u,w,d)} \tag{31}$$

and the ideal transformed input for a dynamic model Eq. 29 becomes

$$v_A = \underbrace{B^{-1}(f_w(u, w, y, d) - Ay)}_{g(u,w,y,d)} \tag{32}$$

Assuming that we are able to generate the ideal inverse g^{-1} and that the inverse is internally stable, the resulting transformed system becomes as before linear and independent of disturbances. Specifically, the transformed system becomes $y = B_0 v_0$ if the model is static and $\frac{dy}{dt} = Ay + Bv_A$ if the model is dynamic. In most cases, we choose $B_0 = I$ and $B = -A$ which give $y = v_0$ and $y = v$ at steady state. Choosing A diagonal also gives decoupling for the dynamic case. More generally, we can have a combination of dynamic and static model equations, as illustrated in Example 5.

Relative order. Note that to achieve the ideal properties of linearity, disturbance rejection and decoupling, we must assume that we can generate the ideal inverse g^{-1}. For the static case, this means that u must have a direct effect on y in Eq. 28, which means that the relative order from u to y is 0. For the dynamic case, this means that u has a direct effect on $\frac{dy}{dt}$ in Eq. 29, which means that the relative order from u to y is 1. If this is not satisfied, then we will have to use an approximate cascade implementation for the inverse (Figures 7b and 7c) and perfect disturbance rejection cannot be achieved for all disturbances. Perfect disturbance rejection will only be approached asymptotically if we can make the slave loop sufficiently fast. However, this may not always be possible because of unstable zero dynamics.

Choice for u. Note that expressions for the transformed variables v_0 and v_A depend on the model equations only and not on what particular variable we select to be the physical input u. The resulting ideal transformed system is also in theory independent of what we select as the input u. However, the expressions for generating the input $u = g^{-1}(v, w, y, d)$ will depend on u. More importantly, if the real process is different from the model, for example, because of constraints for u, then perfect inversion may not be possible and the behavior of the real transformed system will depend on the choice for u.

Dynamics of transformed system with w-variables. When we include w-variables in the ideal static transformed inputs v_0, then the dynamics of the transformed system (from v_0 to y) will no longer be the same as of the original system (from u to y). The reason is the feedback from w. An example is given by the variables $v_{0,w}$ for the heat exchanger in Example 4 (Figure 12) where we see that the the dynamics of the transformed system gets slow. Note that for the static case, we have no parameter to choose the dynamics of the transformed system.

On the other hand, in the dynamic case, that is with ideal dynamic transformed inputs v_A, we can use the matrix A to freely set the dynamics of the transformed system, also when v_A depends on w. However, note that the dynamics of the transformed system (from v_A to y) will not be the same as for the original dynamics (from u to y), no matter how we choose A. One reason is that the number of differential equations describing the transformed dynamic system $\frac{dy}{dt} = Ay + Bv$ is generally lower than that of the original dynamic system in Eq. 30. The choice $A = \text{diag}(\partial f_w/\partial y)_*$ may be a good starting point as it gives little feedback from y, but this choice will not keep the original dynamics, because u also has an indirect (and possible high-order) effect on y through the variable w.

Unstable zero dynamics [2] *and internal instability of the inverse.* Note that

[2]Unstable zero dynamics go by many names. They are the same as RHP-zeros for linear systems, and linear systems with RHP-zeros and/or time delay are also called non-minimum phase systems. In the linear scalar case, RHP-zeros always give inverse response in the time

we are essentially treating w as measured disturbance when deriving the transformed variables v_0 and v_A in Eq. 31 and Eq. 32. Is there any problem in doing this? Yes, if w depends on the input u in a dynamic way, then this will introduce dynamics in the map from u to the transformed input v. If this results in unstable zero dynamics from u to v (which may be v_0 or v_A or any other transformed input), then this will result in internal instability when generating $u = g^{-1}(v, w, y, d)$ using the ideal inverse. This follows because the unstable zeros of the original map become unstable poles of the inverse map. A simple example is given in the discussion section. This means that the implementations with the exact inverse in Figures 7a and 8 may yield internal instability in some cases. Fortunately, as argued in the discussion section, it's not very likely to happen in practice, because unstable zero dynamics require that the indirect dependency of v on u through w is strong.

The internal instability can in any case be avoided if we use the alternative implementation with a slave v-controller in Figure 7b, but the slave controller C_v then needs to be tuned sufficiently slow so that the unfavorable zero dynamics do not cause closed-loop instability. Thus, disturbance rejection will not be perfect in this case.

4.4. Ideal static transformed variable v_0 applied to a dynamic system

Clearly, if we apply v_0 in Eq. 31 with $B_0 = I$ (derived from a static model) as a transformed input to a dynamic system (with the same static model), then we get linearization, decoupling and disturbance rejection at steady state.

But what happens dynamically? We cannot say anything in general, but fortunately, if we apply v_0 to the particular dynamic system in Eq. 29, then we get perfect disturbance rejection also dynamically, if we initially are at steady state. This surprising fact, which is proved in the discussion in Section 6, holds because of the particular simple dynamics assumed in Eq. 29. The assumption

domain. More general, for nonlinear systems the unstable zero dynamics from u to v correspond to the resulting unstable dynamics of the inverse map from v to u (Isidori, 1995).

about being initially at steady state is usually not limiting, because it's desirable that the system remains close to steady state.

Also note that when v_0 is independent of w, i.e., we have $v_0 = f_0(u, d)$, then there is no feedback from the outputs or states, and the transformed dynamic system (from v_0 to y) retains the dynamics of the original system (from u to y) without needing any tuning parameter. These two facts makes it tempting for an engineer to apply v_0 also to dynamic system.

However, there are advantages of instead applying the dynamic transformed input v_A (Eq. 32) to the dynamic system $\frac{dy}{dt} = f(u, w, y, d)$. First, the transformed system from v_A to y is linear dynamically, whereas the transformed system from v_0 to y is linear only at steady state. This is seen in several of the examples, e.g., from (Eq. 42) in Example 3. Second, for the case when the static transformed input v_0 depends on w, the system dynamics change because of the resulting feedback from w to u, but we have no parameter in v_0 to affect it. On the other hand, when we use the ideal dynamic transformed variable v_A, we can choose the dynamics of the linear transformed system through the parameter A.

5. Examples

5.1. Example 3: Simple nonlinear level process

We consider control of level (volume V) in a tank with two inflows and one outflow. Assuming constant liquid density, we derive from the mass balance the following dynamic model

$$\frac{dV}{dt} = F_1 + F_2 - F_3 \tag{33}$$

The level has some self-regulation, as the outflow is assumed to be

$$F_3 = k\sqrt{k_1 V + p_0} \tag{34}$$

where $\Delta P = k_1 V + p_0$ is the pressure drop at the outlet. For simplicity, we assume that the variables p_0 and V are scaled such that $k_1 = k = 1$. We also assume that the inflow $u = F_1$ can be manipulated directly, for example, because we have a fast slave flow controller. The dynamic model then becomes

$$\frac{dy}{dt} = f(u, y, d) = u + d_1 - \sqrt{y + d_2} \tag{35}$$

where the variables are defined as follows

$y := V$ = volume of fluid;

$u := F_1$ = inflow (manipulated)

$d_1 := F_2$ = inflow (disturbance)

$d_2 := p_0$ = pressure difference before and after tank

Transformed variable v_A based on dynamic model. From (19) the ideal transformed variable is $v_A = B^{-1}(f(u, y, d){-}Ay)$. We choose $B = -A$ so that $y = v_A$ for the transformed system at steady state. With this choice we get,

$$v_A = y - A^{-1}f(u, y, d) = \underbrace{y - A^{-1}(u + d_1 - \sqrt{y + d_2})}_{g(u,y,d)} \tag{36}$$

and the resulting transformed dynamic system becomes as expected

$$\frac{dy}{dt} = A(y - v_A) \tag{37}$$

For implementation using Figures 7a or 8, we invert the expression $v_A = g(u, y, d)$ in Eq. 36 to find the corresponding input,

$$u = g^{-1}(v_A, y, d) = A(y - v_A) - d_1 + \sqrt{y + d_2} \tag{38}$$

The constant A is a tuning parameter which determines the dynamics of the transformed system. To eliminate the feedback from the output y to the transformed variable v_A at the nominal point, we may from Eq. 27 choose

$$A = \left(\frac{\partial f}{\partial y}\right)_* = -\frac{1}{2\sqrt{y^* + d_2^*}}$$

where y^* and d_2^* denote the nominal steady-state values.

Transformed variable v_0 based on static model. In some cases it is simpler to formulate a static model than a dynamic model, or the dynamic model is of high order and cannot be written in the form $\frac{dy}{dt} = f(u, y, d)$. Although this is not the case here, let us see what happens if we instead derive the transformed input v_0 based on the static model. Solving $0 = f(u, y, d)$ in Eq. 35 with respect to y gives the static model

$$y = f_0(u, d) = (u + d_1)^2 - d_2 \tag{39}$$

The ideal transformed variable based on this static model (selecting $B_0 = 1$) is

$$v_0 = f_0(u, d) = (u + d_1)^2 - d_2 \tag{40}$$

For a given v_0, we invert this relationship to get the corresponding input

$$u = f_0^{-1}(v_0, d) = \sqrt{v_0 + d_2} - d_1 \tag{41}$$

At steady state this gives the transformed system $y = v_0$, independent of disturbances d_1 and d_2. Also note that the steady-state response from the transformed input v_0 to the output y is linear with gain $B_0 = 1$.

What happens if we apply the ideal static transformed input v_0 to the dynamic system? Substituting Eq. 41 into Eq. 35 gives the following transformed dynamic system:

$$\frac{dy}{dt} = \sqrt{v_0 + d_2} \; - \; \sqrt{y + d_2} \tag{42}$$

We find that the transformed dynamic system with v_0 is independent of d_1 but it depends on d_2. However, for practical purposes we have perfect disturbance rejection also for d_2. To see this, note that $y = v_0$ at steady state. It then follows that if we are initially at steady state and we keep v_0 constant, then from Eq. 42 we have $dy/dt = 0$ for any disturbance d_2. Thus, y will remain at v_0 and we have perfect disturbance rejection for d_2 also dynamically.

Thus, for disturbance rejection, we may as well use the static transformed variable v_0 rather than v_A based on a dynamic model, and this may be proven to hold generally for other systems (see Discussion section). Also, with v_0 we have the advantage that there is no feedback from the output y, so the transformed system will retain the dynamics of the original system without needing to choose the parameter A. The disadvantage is that the dynamic response (42) from v_0 to y is nonlinear, whereas the dynamic response from v_A to y is linear. However, this in itself may not be very important for practical implementation, so it's likely that an engineer may prefer using the ideal static transformed input v_0 rather than v_A.

5.2. Example 4. Heated tank

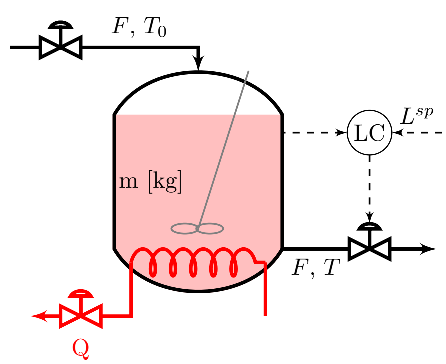


Figure 9: Process flowsheet of tank heated with an electric coil.

Consider the continuous process in Figure 9 with an electric heater. Assuming perfect mixing, constant heat capacity c_P [kJ °C^{-1}] and constant mass holdup m [kg], the energy balance gives the following dynamic model

$$\frac{dT}{dt} = f(u, y, d) = \frac{1}{mc_P}(Fc_P(T_0 - T) + Q) \tag{43}$$

The objective is to control the outlet temperature $y = T$ using the inlet flowrate $u = F$ [kg/s] as the manipulated input (we assume that we have a fast slave flow controller so that we can consider $u = F$ to be the physical input). Q and T_0 (heat input and inlet temperature) are measured disturbances. Setting $dT/dt = 0$. we derive the corresponding static model for the outlet temperature

$$T = f_0(u, d) = T_0 + \frac{Q}{Fc_P} \tag{44}$$

From Eq. 31 and Eq. 32, the ideal transformed inputs v_0 and v_A, based on a static and dynamic model, respectively, become

$$v_0 = f_0(u, d) = T_0 + \frac{Q}{Fc_P} \tag{45a}$$

$$v_A = -A^{-1} f(u, y, d) + y = -A^{-1}\left(\frac{F}{m}(T_0 - T) + \frac{Q}{mc_P}\right) + T \tag{45b}$$

We have here chosen the parameters $B_0 = I$ and $B = -A$ so that we at steady state have $y = v_0$ and $y = v_A$, respectively.

If we apply these two transformed variables to the dynamic system in Eq. 43 then the transformed dynamic system becomes for the two cases

$$\frac{dT}{dt} = \frac{F}{m}(v_0 - T) \tag{46a}$$

$$\frac{dT}{dt} = -A(v_A - T) \tag{46b}$$

For both cases, we find that the transformed system is independent of disturbances (in Q and T_0). If we choose $A = -\left(\frac{\partial f}{\partial T}\right)_* = -\frac{F^*}{m}$, then we see that transformed systems in terms of v_0 and v_A are identical close to the nominal operating point $(*)$, but note that the transformed system in terms of v_0 is nonlinear, whereas the transformed system in terms of v_A is linear.

Note that the expressions for for the transformed variables v_0 and v_A, and also the expressions (46) for the transformed systems, do not depend on what we choose as the input u (as expected). However. in practice the choice for

u it may matter, for example, because we may get singularity in the input transformation or because of input constraints. This problem may occur with $u = F$ as discussed next.

For implementation using the exact inverse in Figures 7a and 8, we need to invert the expressions for v to find the physical input $u = F$. For the case when we use $v = v_0$ based on a static model we get from Eq. 45a,

$$u = F = \frac{Q}{c_p(v_0 - T_0)} \tag{47}$$

We note that there is a singularity at $v_0 = T_0$. This may be a problem, because it may happen that the outer controller C makes a large decrease in v_0 (possibly to speed up the response) so that v_0 drops below T_0. This will cause the input u to jump from a large positive value (in practice, with constraints, from $u = u_{max}$) to a large negative value (in practice, to $u = u_{min} = 0$). We simulated this, and found that it made the system drift away from the desired steady state, and it did not recover.

A similar singularity occurs when we use the dynamic model to derive the ideal transformed input v_A. We find by inverting Eq. 45b,

$$u = F = \frac{Q + Amc_P(v - y)}{\rho c_p(y - T_0)} \tag{48}$$

The singularity at $y = T_0$ may happen in situations with large dynamic variations in T_0 or $y = T$.

Both for v_0 and v_A, there are ways of handling the singularity. One is to set $u = u_{max}$ when the computed value of u is negative. Another way is to use the cascade implementation in Figure 7b with a slave v-controller.

5.3. *Example 5. Ideal transformed inputs for mixing process (Motivating Example continued)*

Consider the mixing process in Figure 2 with the following inputs, outputs and disturbances:

$$u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}; \quad y = \begin{bmatrix} F \\ T \end{bmatrix}; \quad d = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \tag{49}$$

This is the same process as in Example 1, where we to obtain decoupling used engineering insight to propose a sum and a ratio of the flows as transformed inputs, see Eq. 4. In this section, we will derive ideal transformed inputs using systematic methods. Note that we assume that the flows F_1 and F_2 are the physical inputs u.

For simplicity, we assume that the mass m [kg] of the system is constant, which is a reasonable assumption in many cases. The dynamic mass balance $\frac{dm}{dt} = F_1 + F_2 - F$ then gives by setting $\frac{dm}{dt} = 0$, the following static mass balance:

$$F = F_1 + F_2 \tag{50}$$

Assuming perfect mixing, the dynamic energy balance becomes

$$m\frac{dT}{dt} = F_1T_1 + F_2T_2 - FT \tag{51}$$

We have here assumed constant and equal heat capacities so that c_P drops out of the energy balance. Substituting the mass balance Eq. 50 into the energy balance Eq. 51 and using the more general notation in Eq. 49, then gives the following model equations for the mixing process

$$y_1 = \underbrace{u_1 + u_2}_{f_{0,1}(u,d)} \tag{52a}$$

$$\frac{dy_2}{dt} = \underbrace{\frac{1}{m}\left(u_1(d_1 - y_2) + u_2(d_2 - y_2)\right)}_{f_2(u,y,d)} \tag{52b}$$

We see from Eq. 52a and Eq. 52b that this is a coupled (interactive) process, since both inputs (u_1 and u_2) affect both outputs ($y_1 = F$ and $y_2 = T$). This makes single-loop control challenging and control performance may be poor. We therefore want to consider the use of ideal transformed inputs which has the potential of giving a linear and decoupled transformed system, and in addition give perfect feedforward action from the disturbances in T_1 and T_2.

5.3.1. Ideal transformed input v_0 from static model

We first derive the transformed inputs that result for the case when the holdup m can be neglected ($m = 0$) and we have a purely static model. This assumption is reasonable for many practical mixing processes. By setting $m\frac{dy_2}{dt} = 0$ we derive from Eq. 52b the following static equation for the temperature $y_2 = T$:

$$y_2 = \underbrace{\frac{u_1 d_1 + u_2 d_2}{u_1 + u_2}}_{f_{0,2}(u,d)} \tag{53}$$

With the standard choice $B_0 = I$, the ideal static transformed inputs v_0 are simply the right-hand side f_0 of the static model equations.. Thus, the ideal static transformed inputs for the mixing tank are:

$$v_{0,1} = \underbrace{u_1 + u_2}_{g_1(u) = f_{0,1}(u,d)} \tag{54a}$$

$$v_{0,2} = \underbrace{\frac{u_1 d_1 + u_2 d_2}{u_1 + u_2}}_{g_2(u,d) = f_{0,2}(u,d)} \tag{54b}$$

The static model for the transformed system becomes $y = v_0$, or equivalently $y_1 = F = v_{0,1}$ and $y_2 = T = v_{0,2}$. As expected, the transformed system is decoupled, independent of disturbances and linear (with gain equal to the identity matrix, I).

For implementation using the exact inverse, we need to invert the expressions

(54) for v_0 to find the physical inputs (flows) u. We get

$$u_1 = g^{-1}(v_0, d)_1 = \frac{v_{0,1}(v_{0,2} - d_2)}{d_1 - d_2} \tag{55a}$$

$$u_2 = g^{-1}(v_0, d)_2 = \frac{v_{0,1}(d_1 - v_{0,2})}{d_1 - d_2} \tag{55b}$$

Note that there is a singularity in the transformation when the two inlet flows have the same temperature, $d_1 = d_2$. This is not a limitation of the proposed method, because it is then physically impossible to freely set the temperature $y_1 = T$ of the mixed flow.

5.4. Comparison with engineering-based variables from Example 1

Comparing the ideal static transformed inputs in Eq. 54 with the engineering-based variables in Eq. 4, we see that $v_{0,1}$ is the sum $u_1 + u_2$ as before. The second variable $v_{0,2}$ is very similar to the ratio $v_2 = \frac{u_1}{u_1+u_2}$ in (4b), except that $v_{0,2}$ includes feedforward action from disturbances $d_1 = T_1$ and $d_2 = T_2$. Note that $v_{0,2} = v_2(d_1 - d_2) + d_2$. For cases where we do not measure the disturbances, the best option is to select d_1 and d_2 as constants (at their nominal values), and in this case we get that the transformed inputs $v_{0,2}$ and v_2 are equivalent from a control point of view, since $v_{0,2} = c_1 v_2 + c_2$, where $c_1 = d_1 - d_2$ and $c_2 = d_2$ are constants. Equivalent here means that both transformed inputs provide decoupling and nominal linearization. In summary, whereas the engineering-based variables in Eq. 4 give only decoupling, the systematic variables v_0 in Eq. 54 also provide in addition perfect feedforward control and linearization. The engineering-based ratio $v_2 = \frac{u_1}{u_1+u_2}$ provides partial linearization, because we have perfect linearization for nominal values of the disturbances.

5.4.1. Ideal transformed input v_A from dynamic model

We here consider the case where the holdup m cannot be neglected, so the energy balance is dynamic. We still make the assumption that the holdup m is constant, so the mass balance is static. The model is then as given in Eq. 52, which consists of both static and dynamic model equations. This example illustrates nicely that it is possible to derive ideal transformed inputs for systems with combined static and dynamic model equations.

The first model equation is static, so as the first transformed input we use as before the right-hand side $f_{0,1}(u,d)$ of Eq. 52a. That is, we still use the sum of the inputs

$$v_{0,1} = u_1 + u_2 \tag{56}$$

as the first transformed input. To derive the second transformed input, we use the right-hand side $f_2(u,y,d)$ of the dynamic energy balance in Eq. 52b. From Eq. 24a we derive the ideal transformed input

$$v_{A,2} = y_2 - A^{-1} f_2(u,y,d) \tag{57a}$$

$$= \underbrace{y_2 - A^{-1}\frac{1}{m}\left(u_1(d_1 - y_2) + u_2(d_2 - y_2)\right)}_{g_2(u,y,d)} \tag{57b}$$

Note that we have chosen $B = -A$ which gives $y_2 = v_{A,2}$ at steady state. To implement the transformed inputs $v_{0,1}$ and $v_{A,2}$ in practice, we need to compute the physical inputs u (flowrates u_1 and u_2) from the inverse transformation $u = g^{-1}(v,y,d)$, see Figure 7a. From Eq. 56) and Eq. 57b we derive:

$$u_1 = g^{-1}(v,y,d)_1 = \frac{v_{0,1}(y_2 - d_2) - Am(v_{A,2} - y_2)}{d_1 - d_2} \tag{58a}$$

$$u_2 = g^{-1}(v,y,d)_2 = \frac{v_{0,1}(d_1 - y_2) + Am(v_{A,2} - y_2)}{d_1 - d_2} \tag{58b}$$

The transformed system from the ideal transformed inputs $v = [v_{0,1}\ v_{A,2}]$ to the outputs $y = [y_1\ y_2]$ then becomes

$$y_1 = v_{0,1} \tag{59a}$$

$$\frac{dy_2}{dt} = A(y_2 - v_{A,2}) \tag{59b}$$

which is decoupled, independent of disturbances and linear since A is a constant. The constant A is a tuning parameter. To eliminate the feedback from the output $y_2 = T$ to the transformed variable v_2 in Eq. 57b at the nominal operating point, we choose A such that we keep the nominal linearized dynamics of the original system, which from Eq. 27 gives

$$A = \left(\frac{\partial f_2}{\partial y_2}\right)_* = -\frac{F^*}{m} \tag{60}$$

where $F^* = u_1^* + u_2^* = v_{0,1}^*$ is the nominal total flowrate.

In Figure 7a, we have also included the outer feedback controller C. Note that since the transformed system is decoupled, it is optimal to use two single-loop controllers $C = \mathrm{diag}(C_1, C_2)$. There will be one flow controller (C_1) that computes $v_{0,1}$ and one temperature controller (C_2) that computes $v_{0,2}$. Also note that since the transformed process is linear, linear controllers will suffice. For the flow controller, we recommend an I-controller since the process is static. For the temperature controller, a PI-controller is recommended.

One question that arises is if the feedback C controller is really necessary, since we have $y_1 = v_{0,1}$ and $\frac{dy_2}{dt} = A(y_2 - v_{A,2})$ for the transformed system. Thus, in theory, we could eliminate the controller C by directly setting $v_{0,1}$ and v_2 equal to the setpoint y^s. However, in reality, there will be model error, imperfect measurements of the disturbances that enter the transformation (T_1 and T_2) and additional unknown disturbances (for example, heat loss). The outer feedback controller C is needed to correct for these unavoidable sources of uncertainty. We may also want to use C speed up or slow down the response for y.

5.4.2. Applying the ideal static transformed input v_0 to the dynamic system

What happens if we apply the static transformed input $v_{0,2}$ to the dynamic system in Eq. 53? Substituting u_1 and u_2 from Eq. 55a and Eq. 55b into Eq. 52b gives after a little algebra the following transformed dynamic system

$$y_1 = v_{0,1} \tag{61a}$$

$$\frac{dy_2}{dt} = \frac{v_{0,1}}{m}(v_{0,2} - y_2) \tag{61b}$$

We see that both disturbances d_1 and d_2 drop out, so the transformed system in Eq. 61 based on $v_{0,2}$ is independent of disturbances, also dynamically [3]. We note that the transformed system in Eq. 61 is not truly decoupled, because we see from Eq. 61b that $v_{0,1}$ also affects output y_2. However, for practical purposes, we have decoupling, because if we start from a steady-state operating point, where we have $y_2 = v_{0,2}$, then Eq. 61b tells that a change in $v_{0,1}$ will *not* affect y_2.

Note that the expression for the transformed system in Eq. 61b in terms of $v_{0,2}$ is very similar to Eq. 59b in terms of $v_{A,2}$ if we choose $A = -\frac{v^*_{0,1}}{m}$ as given in Eq. 60. The main difference is that the transformed dynamic system Eq. 59b for v_2 is linear, whereas the transformed dynamic system Eq. 61b for $v_{0,2}$ is nonlinear because of the multiplication with the term $v_{0,1}$ which is time-varying.

In summary, the only advantage of using the more complex variables v_A rather than v_0 derived from a static model, is that the transformed system is more linear. This benefit is probably not sufficient to justify the added complexity, so it is likely that the engineer will prefer to use the static variables v_0. If the disturbances are not measured or do not change frequently, then this is equivalent to the simple sum and ratio proposed in (4) in the Motivating example.

5.4.3. Dynamic simulations

We now illustrate by simulations that the ideal transformed variables indeed give the expected perfect responses, when we assume no model error and perfect measurements of the disturbances. All simulations use the implementation with the model-based inverse in Figure 7a, but without the outer controller C, that is, we have $C = 0$. The inverse transformation $u = g^{-1}(v, \dots)$ in Figure 7a is

[3] Generally, when we apply static transformed inputs v_0 to a dynamic system of the form $\frac{dy}{dt} = f(u, y, d)$, we need to make the assumption that the system is initially at steady state to get perfect dynamic disturbances rejection. However, this assumption is not necessary for this particular case since the disturbances drop out completely in the transformed system.

given by Eq. 55 when we use the ideal transformed inputs $v_{0,1}$ and $v_{0,2}$ based on a static model for temperature (y_2), and by Eq. 58 when we use the ideal transformed inputs $v_{0,1}$ and $v_{A,2}$. All simulations use the nonlinear process model in Eq. 52, that is, the model for $y_2 = T$ is dynamic.

For the ideal dynamic transformed input $v_{A,2}$, we select as mentioned above $A = \frac{\partial f_2}{\partial y_2} = \frac{v^*_{0,1}}{m}$. With this choice for A, the ideal static and dynamic transformed inputs variables ($v_{0,2}$ and $v_{A,2}$) give very similar dynamic responses, see Eq. 61b and Eq. 59b, and this is confirmed in the simulations.

Process data. Table 1 shows the nominal operating conditions for the mixing process. At the nominal operating point the two inputs are equal ($F_1 = F_2$), which makes the process highly coupled and difficult to control using conventional single-loop PID-controllers.

Table 1: Nominal operating conditions for Example 5 (mixing process).

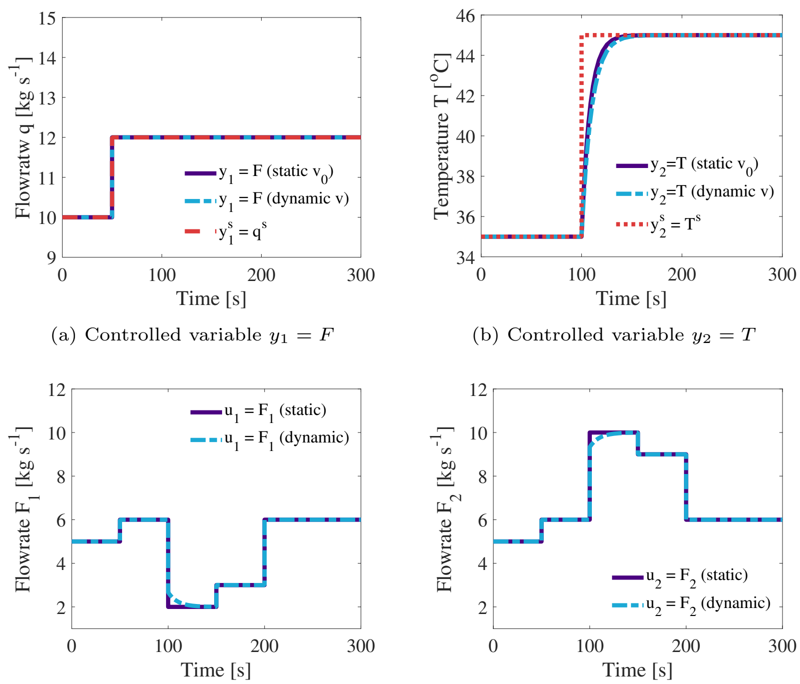
Variable	F_1	F_2	F	T_1	T_2	T	m
Value	5	5	10	20	50	35	100
Unit	$\mathrm{kg\,s^{-1}}$	$\mathrm{kg\,s^{-1}}$	$\mathrm{kg\,s^{-1}}$	°C	°C	°C	kg

With no model error and perfect disturbance measurement, the simulations show that both outputs $y_1 = q$ (Figure 10a) and $y_2 = T$ (Figure 10b) are independent of the two disturbances) and follow the original system dynamics for setpoint changes at time $t = 200\,\mathrm{s}$ and at time $t = 150\,\mathrm{s}$ respectively. The holds for both ideal static transformed variables v_0 and the ideal dynamic variables v_A. The inputs u_1 in Figure 10c and u_2 in Figure 10c change in a step-wise manner because we use a static algebraic block to compute them.

The simulation results are not very exciting or surprising, and simply confirm what is expected from the transformed system models in Eq. 61b and Eq. 59b. The responses for the ideal static and dynamics transformed inputs are identical, except for the dynamic transients when we have a setpoint change for y_2 (at $t = 100$s). This is because $v_{0,1}$ is at 12 $\mathrm{kg\,s^{-1}}$, rather than at it's nominal value of 10 $\mathrm{kg\,s^{-1}}$, which results in a slightly faster response for y_2 for the static case

(v_0). We also see that the inputs u_1 and u_2 make a larger initial change at $t = 100$s for the static case.

The benefit of using the dynamic transformed input v_A rather than the static transformed input v_0 is mainly that we get a linear transformed system for designing the outer controller C, but this benefit is not seen in these simulations since we have used $C = 0$.



(a) Controlled variable $y_1 = F$

(b) Controlled variable $y_2 = T$

(c) Physical input $u_1 = F_1$

(d) Physical input $u_2 = F_2$

Figure 10: Simulation response for the mixing process in Example 5 using using both ideal static (v_0) and dynamic (v_A) transformed inputs and the exact implementation of the inverse (Figure 7a).
The simulations are for the following four step disturbances: 2 °C increase in disturbance $d_1 = T_1$ at time $t = 50$ s. 5 °C increase in disturbance $d_2 = T_2$ at time $t = 100$ s. 1 °C increase in setpoint $y_2^s = T^s$ at time $t = 150$ s. 1 kg s^{-1} increase in setpoint $y_1^s = q^s$ at time $t = 200$ s.

Write on all subfigures: (static v0) and (dynamic vA) .... not just (dynamic v)

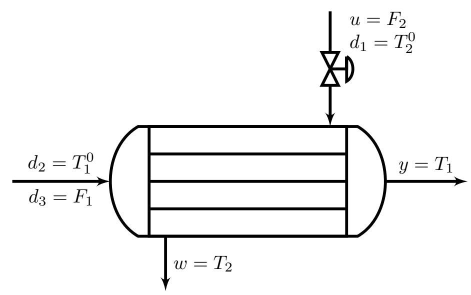


Figure 11: Heat exchanger where the objective is to control the outlet temperature of stream 1 (process side) by exchanging heat with stream 2 (utility side).

5.5. Example 6: Heat exchanger

Temperature control using heat exchangers may benefit from the use of input transformations, both to reduce nonlinearity and to introduce feedforward control. Consider the process in Figure 11 where the objective is to control the outlet temperature of stream 1 (which may be the process side) by exchanging heat with stream 2 (which may be the utility side). We assume that the input (manipulated variable) is the utility flowrate, $u = F_2$. Actually, the true manipulated variable is the valve position z_2, but we will use a slave v-controller for implementation and use the measured flow $w = F_2$ when computing the transformed variable v. This means that a separate flow controller will not be needed. In summary, we have for this example

$$u = F_2,\ y = T_1$$

Measured disturbances are the inlet temperatures and the flowrate of stream 1,

$$d = [T_1^0\ T_2^0\ F_1]$$

In the simulations, we will also consider an unmeasured disturbance in the UA-value, for example, caused by fouling or gas bubbles in the streams,

$$d_{\text{unmeasured}} = UA$$

A possible extra measurement (in addition to F_2) that depends on the input u is the utility outlet temperature

$$w = T_2$$

The dynamic and steady-state behaviors of heat exchangers are highly nonlinear. For example, for small values of $u = F_2$ (relative to F_1), the process gain $k = \frac{dy}{du}$ is large and relatively constant, but for large values of $u = F_2$, the gain k approaches 0 and makes it difficult to control $y = T_1$. This happens because we for large F_2 get a pinch for T_1 (constant value) with $y = T_1$ approaching the inlet temperature T_2^0.

An ideal countercurrent heat exchanger is modelled by partial differential equations, but we use a cell model with $n = 100$ well-mixed cells on each side; see (Reyes-Lúa et al., 2018) for model equations. In total, this gives 200 differential equations to represent the temperature dynamics, so this model clearly cannot be written in the form $dy/dt = f(u, y, d)$ in (18) which allows for only one differential equation. This leads us to consider transformed inputs based on a static model of the heat exchanger. We will consider two transformed inputs

$$v_0 = f_0(u, d) \tag{62}$$

$$v_{0,w} = f_{0,w}(u, w, d) \tag{63}$$

The first is the ideal transformed input v_0 that follows from the detailed static model $y = f_0(u, d)$. This model has the input $u = F_2$ and the three disturbances d as independent variables. Note that this model does not depend on the measured state variable $w = T_2$ and and use of the transformed variable v_0 will therefore retain the dynamics of the original system (the heat exchanger).

The second transformed variable, $v_{0,w}$, is inspired by an actual industrial implementation, where we make use of the measured variable $w = T_2$. This allows us to use a much simpler model, based on just energy balances, without using the detailed model of the heat exchanger. For example, whereas v_0 depends on the UA-value, $v_{0,w}$ does not use this information.

Both transformed inputs are based on steady-state expressions for $y = T_1$ and give $y = v^s$ at steady state. Thus, both transformed inputs will provide perfect disturbance rejection and linearity at steady state. However, this assumes that the model parameters do not change and we will find that that v_0 gives an offset if we change the value of UA, whereas $v_{0,w}$ gives no offset because it uses the measurement $w = T_2$ instead of the model. On the other hand, as we will see from the simulations, there are disadvantages with the transformed input $v_{0,w}$ when it comes to the dynamic response.

5.5.1. Ideal transformed input v_0 based on full static model

We assume that the fluids do not change phase and have constant heat capacity c_{p1}, c_{p2}). Assuming ideal countercurrent flow, the steady-state behavior is then given by the following three equations for the heat transfer Q from stream 1 to stream 2:

$$Q = F_1 c_{p1}(T_1^0 – T_1) \tag{64a}$$

$$Q = F_2 c_{p2}(T_2 – T_2^0) \tag{64b}$$

$$Q = UA \frac{\left(T_1^0 - T_2\right) - \left(T_1 - T_2^0\right)}{\ln\left(\frac{T_1^0 - T_2}{T_1 - T_2^0}\right)} \tag{64c}$$

This gives 3 equations in 3 unknowns (Q, T_1, T_2) which can be solved analytically to find the following expression for T_1 as a function of the input and the disturbances (e.g., Soave & Barolo (2021))

$$y = T_1 = \underbrace{T_1^0 + \epsilon(T_2^0 - T_1^0)}_{f_0(u,d)} \tag{65}$$

where

$$\epsilon = \frac{1{-}E}{C - E}$$
$$C = \frac{F_1 c_{p1}}{F_2 c_{p2}}$$
$$E = \exp\left(UA\left(\frac{1}{F_1 c_{p1}}{-}\frac{1}{F_2 c_{p2}}\right)\right)$$

From (65) the corresponding ideal static transformed input becomes

$$v_0 = f_0(u, d) = T_1^0 + \epsilon(T_2^0 - T_1^0) \tag{66}$$

5.5.2. Transformed input $v_{0,w}$ based on parts of static model

The second transformed variable, $v_{0,w}$, follows by using only the two first expressions for Q in Eqs. 64a and 64b). We first use Eq. 64a to find

$$T_1 = T_1^0 + \frac{Q}{F_1 c_{p1}}$$

and then we substitute Q using Eq. 64b to get

$$y = T_1 = \underbrace{T_1^0 + \frac{F_2 c_{p2}}{F_1 c_{p1}}(T_2^0 - T_2)}_{f_{0,w}(u,w,d)} \tag{67}$$

From (67) the corresponding ideal static transformed input becomes

$$v_{0,w} = f_{0,w}(u, w, d) = T_1^0 + \frac{F_2 c_{p2}}{F_1 c_{p1}}(T_2^0 - T_2) \tag{68}$$

The second transformed input $v_{0,w}$ does not use the model equation for Q in Eq. 64c involving the heat transfer, but instead makes use of the extra measurement $w = T_2$.

5.5.3. Implementation

For both transformed inputs, we will use the pure feedback-based implementation in Figure 7b with a slave v-controller. An alternative would be to use a model-based inverse to compute $u = F_2$ plus a slave flow controller to implement F_2 (Figure 7c). This requires analytically or numerically inverting the model equations (f_0 or $f_{0,w}$). The use of the slave v-controller avoids this, which has two advantages. First, for v_0 we avoid implementing a numerical solution to generate $u = f_0^{-1}(v_0, d)$, and instead we generate the inverse by the slave controller C_v, which is a fast I-controller. Second, for $v_{0,w}$ we avoid the potential internal instability by using the exact inverse. Also, even if have no problem with instability, the dynamic input u generated by the exact inverse may be excessive.

The transformed input $v_{0,w}$ depends on the measured variable $w = T_2$. This feedback will change the dynamics, such that the dynamics of the transformed system will be different to that of the original system. On the other hand, the use of v_0 has no feedback from any output or state, and the dynamics will not change (except for the dynamics of the slave loop, which are negligible in this case because the slave loop for v_0 is fast).

For tuning the slave v_0-controller, we note that the process from $u = F_2$ to v_0 is static (and also the dynamics from the valve position (z_2) to v_0 are generally fast because of fast valve dynamics), and a pure I-controller is recommended (Skogestad, 2003). However, for the $v_{0,w}$-controller, the dependency of $v_{0,w}$ on $w = T_2$ results in a significant overshoot due to stable (LHP) zero in the response from $u = F_2$ to $v_{0,w}$, which may make tuning more difficult. For simplicity, we use I-controllers for both v_0 and $v_{0,w}$, tuned based on the initial gain, and with the same closed-loop time constant ($\tau_C = 10$ s); see Table 2.

Table 2: Tunings for slave v-controller heat exchanger example

Transformed input	K_I	τ_C [s]
v_0	-0.125	10
$v_{0,w}$	-0.01	10

5.5.4. Simulations

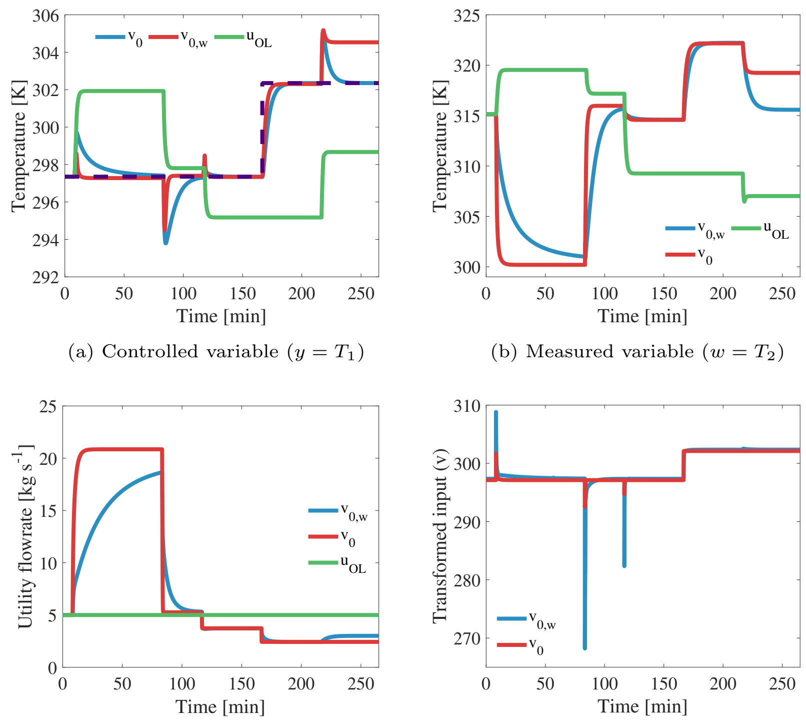
The simulations in Figure 12 compare the two alternative transformed inputs (v_0 or $v_{0,w}$) with the open-loop response with no input transformation (that is, when $u = F_2$ is constant). The simulations show responses to step disturbances in F_1, T_1^0 and T_2^0. The setpoint of the transformed input v^s is initially at 297 K and changes to 302 K at time $t = 167$ min.

From the response for the controlled variable ($y = T_1$) in Figure 12a, we clearly see that there is a benefit of using transformed inputs. Both transformed inputs give in theory perfect control at steady state ($y = v$) for measured disturbances and this is confirmed by the simulations. For the unmeasured disturbance in UA (towards the end of the simulation in Figure 12a), we see as expected that we get an offset for $y = T_1$ when we use v_0 as the transformed input, but not when we use $v_{0,w}$.

Dynamically, we find that the responses are best (faster) when we use v_0 as the transformed input (red curve). The disturbance rejection with v_0 is not perfect dynamically because the process dynamics are quite complex and not described by a first-order model. For v_0, the dynamics are as expected similar to the quite fast dynamics of the uncontrolled heat exchanger (green curves).

On the other hand, when we use $v_{0,w}$ (blue curves), which contains an indirect feedback from $w = T_2$, the dynamics for the return to the steady state are much slower. There is not much we can do about this, as there is no tuning parameter in $v_{0,w}$. The slave controller can be used to make the inversion faster, but it will not help in this case. Even with a perfect inverse, the dynamics caused by the feedback from $w = T_2$ will be present. It may be possible to use the outer C controller to speed drive up the response for y, but this could give stability because of measurement delays for y.

In summary, for this example, the responses are best when we use the transformed input v_0 based on the full static model. The exception is for disturbances in the heat exchanger model parameters, including the UA-values, but these can be taken care of by the outer controller C. On the other hand, the implementation of v_0 based on the full static model is complex, so it is nevertheless possible that the simpler implementation using $v_{0,w}$ may be chosen in practice.



(a) Controlled variable ($y = T_1$)

(b) Measured variable ($w = T_2$)

(c) Physical input ($u = F_2$)

(d) Computed transformed input (v).

Figure 12: Dynamic simulation of heat exchanger (Example 4) using the cascade feedback implementation B in Figure 7b. Two choices of the transformed input, v_0 and $v_{0,w}$, are compared with the open-loop (OL) case with no transformation.
The simulations are for the following step disturbances: F_1 from 3 to 4 $\mathrm{kg\,s^{-1}}$ at $t = 8\,\mathrm{min}$, T_2^0 from 293 to 288 K at $t = 80\,\mathrm{min}$, T_1^0 from 343 to 328 K at $t = 117\,\mathrm{min}$, setpoint v^s from 297 to 302 K at $t = 167\,\mathrm{min}$ and U from 150 to 100 $\mathrm{W\,m^{-2}\,K^{-1}}$ at $t = 217\,\mathrm{min}$.

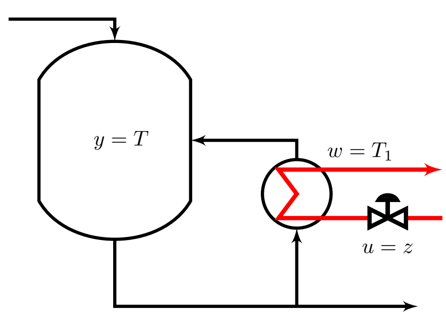


Figure 13: Flowsheet of a tank with heat exchanger (Example 7)

need to change. w=T2, u=z2.

Table 3: Nominal operating conditions for the heat exchanger example from (Skogestad, 2008)

Variable	Value	Unit
F_1	5	$\mathrm{kg\,s^{-1}}$
1_2	3	$\mathrm{kg\,s^{-1}}$
T_2^s	297.2	°C
T_1^0	293	°C
T_2^0	343	°C
U	150	$\mathrm{W\,m^{-2}\,°C^{-1}}$
A	90	$\mathrm{m^2}$
V	0.45	$\mathrm{m^3}$
c_{p1}	1500	$\mathrm{J\,kg^{-1}\,K}$
c_{p2}	1200	$\mathrm{J\,kg^{-1}\,K}$
ρ_1	890	$\mathrm{kg\,m^{-3}}$
ρ_2	980	$\mathrm{kg\,m^{-3}}$

5.6. Example 7. Tank with heat exchanger

In the previous example, the idea of using the measured temperature $w = T_2$ to derive a simpler transformed input v_{0w} was based on a successful industrial implementation. However, we found that it gave a slow dynamic response (Figure 12). However, in the actual industrial implementation, the objective was not to control the exit temperature on the process side (T_1), but rather to control the temperature inside a large tank, $y = T$ (see Figure 13). This has two important implications. First, with a large tank, the dynamics of the heat exchanger are not important compared to the much slower dynamics of the tank. Second, since the industrial objective was to control the tank temperature, the derivation of the transformed input is different.

The process in Figure 13 is a combination of the heated tank in Example 4 and the heat exchanger in Example 6. From (44) the static energy balance gives

$$T = f_0(u, d) = T_0 + \frac{Q}{Fc_P} \tag{69}$$

Here Q is the heat from the utility as given by the three equations in Eq. 64. Note that Q depends on the physical input (the valve position $u = z_2$) as well as on several disturbances (T_2^0, F_1, UA, etc.). From Eq. 31 the ideal static transformed input is (with the choice $B_0 = I$)

$$v_0 = f_0(u, d) = T_0 + \frac{Q}{Fc_P} \tag{70}$$

Assume now that the main disturbances for controlling the temperature $y = T$ come from the term Q. That is, we assume that the disturbances in the feed (F and T^0) can be handled by the outer feedback controller C. Then, based on the expression for v_0, we may suggest using the following simplified transformed input

$$v = Q$$

Next, the question is which expression to use for Q. The standard approach would be to combine the three expressions in (64) to eliminate variables and write Q as a function of $y = T$, u and d. However, this gets quite complicated and a simpler (and maybe better) approach is use only one of the three expressions for Q in (64) by making use of measured dependent variables w. Since the physical input is the valve position $u = z_2$ on the utility side (2), the obvious choice is to use the expression for Q_2 and make use of the dependent measurement $w = T_2$. From (64b) we then derive the following transformed input for control of the tank temperature $y = T$:

$$v = Q_2 = F_2 c_{p2}(T_2 – T_2^0) \tag{71}$$

This transformed input is similar to $v_{0,w}$ in (68), but $v_{0,w}$, which was derived to keep a constant temperature out of the heat exchanger, will not work well for disturbances in F_1. Actually, $v = Q_2$ in (71) is the transformed input that was used in the successful industrial implementation and it can be easily implemented using a slave v (power) controller. To compute v, we must measure measure the flow F_2 and the temperatures (T_2, T_2^0) on the utility (2) side. The slave controller can be tuned to be fast since there is a direct effect from the valve position $u = z_2$ to the power $v = Q_2$.

Note from (69) that the transformed system from $v = Q_2$ to $y = T$ is linear and also independent of disturbances in the utility inlet temperature (T_2^0) and utility pressure (which affects the flow F_2 through the valve equation). If the disturbances in T_2^0 are minor, then one might think from (71) that one may simply the transformed input further by choosing $v = F_2$. The v-controller would then be a flow controller which would counteract the pressure (flow) disturbances and also linearize the system (by avoiding the nonlinearity from $u = z_2$ to F_2 in the valve equation). However, this thinking is not correct, because $v = Q_2$, indirectly through the measurement of $w = T_2$, also counteracts disturbances originating on the process (1) side, that is, it counteracts disturbances in F_1 and T_1^0, and also counteracts disturbances in the heat transfer, for example in the UA-value.

If the physical input was on the process side, $u = z_1$, then it would probably be better to choose $v = Q_1 = F_1 c_{p1}(T_1^0 - T_1)$ as the transformed input for control of the tank temperature.

In summary, the last tank example shows that we in practice may not implement the ideal transformed input (v_0 or v_A), but rather use it as a starting point for suggesting alternative simpler transformed inputs v, which may require fewer disturbance measurements or less modelling effort. The use of the simplified transformed input v may still give significant improvements in the control of y, because v may provide partial disturbance rejection. linearization and decoupling.

We need simulations here to confirm what I claim, although we may not have enough space in this paper. Could be good for a conference paper or at least for the thesis

6. Discussion

6.1. Internal instability of model-based inversion

For the exact implementation of the transformed input v, we must for a given value of v, y, w and d, invert the static map $v = g(u, w, y, d)$ to generate (analytically or numerically) the corresponding value of u,

$$u = g^{-1}(v, y, w, d) \tag{72}$$

Note that Eq. 72 is a purely static expression and therefore by itself does not contain any instabilities. However, when generating the inverse in (72) we are in effect treating w and y as measured disturbances, whereas they in reality depend on u. We will here focus on the dependency of g on w. This feedback dependency may generate internal instability when $u = g^{-1}(v, y, w, d)$ in (72) is applied to the real dynamic system. We use the term *internal* instability because the map from the transformed input v to the output y may appear to be stable, but this may not be true if we consider the input u, and this "hidden" internal instability will eventually appear also in y, either because of model error or because infinite inputs u are not physically realizable.

We have the following general result: *Let $g_u(u, d)$ represent the dynamic map from u to v when the dependency of w and d on u is included. Then we have internal instability when we apply the exact inverse $u = g^{-1}(v, y, w, d)$ in (72) if the map $g_u(u, d)$ contains unstable zero dynamics (RHP-zeros in the linear case) which gives internal instability when we implement the exact desired value for u.* This follows trivially because the inverse will have unstable poles at the unstable poles.

6.1.1. Example: ideal transformed input derived from static model

As a simple example, consider the static system

$$y = u + w + d \tag{73}$$

for which we propose to use as the ideal transformed input the right-hand-side of Eq. 73.

$$v = g(u, w, d) = u + w + d \tag{74}$$

Note that this transformed input follows if we apply the systematic method in Eq. 14 to the static system in Eq. 73 and choose $B_0 = I$. For implementation, Eq. 74 may be solved with respect to u to get the "inverse input transformation"

$$u = g^{-1}(v, w, d) = v - w - d \tag{75}$$

This may be implemented as in Figure 7a and it gives the (ideal) transformed system

$$y = v \tag{76}$$

which has no dynamics and therefore appears to be stable, So far we have not said anything about how w depends on u. In effect, we have treated w as a measured disturbance and we have counteracted the effect of w on v by use of the "feedforward controller" (inverse) in Eq. 75. However, there is a potential "hidden" instability because of the dynamic response from u to w. As an example, assume that it is first-order with a steady state gain of -2,

$$w = \frac{-2u}{4s+1} \quad \text{or} \quad \frac{dw}{dt} = -0.25(2u + w) \tag{77}$$

Note from Eq. 74 that the direct static effect of u on v has a gain of 1, whereas from Eq. 74 the indirect dynamic effect of u on v (through w) has a steady-state gain of -2. The combined effect causes an unstable (RHP) zero from u to v. To see this, eliminate w from Eq. 74 using Eq. 77 to get

$$v = \frac{4s-1}{4s+1} u + d \tag{78}$$

which has an unstable RHP zero at $z = 1/4$. This gives internal instability if we use the exact inverse in 74. To see this, solve (78 with respect to u to get

$$u = \frac{4s+1}{4s-1}(v - d) \tag{79}$$

which as expected is unstable due to the unstable (RHP) pole at $p = 1/4$. The response from v to w is also unstable

$$w = \frac{-2}{4s-1}(v-d) \tag{80}$$

The two instabilities in Eqs. 79 and 80 cancel each other in Eq. 73 to give $y = v$. The system from v to y therefore appears to be stable, but this is not true if we consider the input u.

6.2. Time scale separation for feedback implementations (alternatives B and C)

The feedback implementation in Figure 7b (alternative B) generates only an approximation of the exact inverse in Eq. 9, but the error can be neglected if the inner loop is sufficiently fast. By "sufficiently fast" we mean that the *time scale separation* $\tau_c/\tau_{c,v}$ is sufficiently large. Here, $\tau_{c,v}$ denotes the closed-loop time constant of the inner loop involving the slave controller C_v, and τ_c denotes the response time for the outer loop involving the controller C and the output y. Note that the slave controller C_v generates the inverse by iteration, so reaching complete convergence (steady state) will take infinite time. Assuming a linear first-order response, the approach to convergence (or steady state) within the desired overall response time τ_c is $(1 - e^{\tau_c/\tau_{c,v}})$. Thus, the approach to convergence increases from 63% to 95.0% to 99.3 % as $\tau_c/\tau_{c,v}$ increases from 1 to 3 to 5. Since, convergence (or steady state) for practical purposes is reached at 99.3%, this gives the rule of thumb of requiring a time scale separation between the control layers of at least 5 (Skogestad & Postlethwaite, 2005).

If the time scale separation gets too small, typically 3 or less, the layers will start interacting and we may experience undesired oscillatory behavior or even instability (Baldea & Daoutidis, 2007). A larger value (larger than 5) allows for robustness to process gain variations which will affect the closed-loop time constants of either of the control loops. Therefore, a time scale separation of 10 or larger is usually recommended in most cases. The limiting case of infinite time scale separation corresponds to $\epsilon = \frac{\tau_{c_i}}{t} \to 0$, which is the singular perturbation condition in the mathematical literature.

The main fundamental problem in using a large gain K_I (and thus achieving a large time scale separation) is a possible effective delay (including inverse response) in the response from u to v, especially for cases when v depends on the output variable w. This may be caused by unstable zero dynamics or a measurement delay for w. However, note that unstable zero dynamics are not really a problem of the feedback implementation in Figure 7b, but rather it is a fundamental limitation in the definition of the transformed input v. Specifically, if there are unstable zero dynamics from u to v, then using the exact inverse in Figure 7a will cause internal instability, with an unstable input u. In such cases, the feedback implementation in Figure 7b must be used. Although it does not give the ideal inverse, it may be tuned to be stable.

To avoid excessive changes (spikes) in the value of u send to the process, in particular for the multivariable case, one may insert a filter for the signal u that goes to the process (but not on the signal u that goes to the block that computes v in Figure 7b). For example, a first-order filter may be used, $F = \frac{1}{\tau_F s+1}$ where τ_F is about 5 times larger than the closed-loop time constant $\tau_{c,v}$ for the slave loop involving C_v.

6.3. Decoupling and disturbance rejection when ideal static transformed input v_0 is applied to the dynamic system in (Eq. 18)

In many cases the process is dynamic, but nevertheless we may want to apply the ideal input transformation $v_0 = f_0(u, w, d)$ in Eq. 13, which is derived based on a static process model. Note here that we have chosen $B_0 = I$ so that we have $y = v_0$ at steady state.

When the static transformation in Eq. 13 is applied to a dynamic process, we have that the transformed system is linear, decoupled and independent of disturbances at steady state. However, dynamically we generally do not know what happens; the response may be nonlinear, coupled and dependent on disturbances. However, if we apply the static transformation to the particular dynamic system

$$\frac{dy}{dt} = f(u, w, y, d)$$

in Eq. 18 then *we get perfect disturbance rejection and in many cases decoupling, if we make the reasonable assumption that we are initially at steady state.* This is an important result, but note that the dynamic system $\frac{dy}{dt} = f(u, w, y, d)$ in Eq. 18 is somewhat limited as it only includes low-order dynamic models with as many differential equations as inputs and outputs.

To prove that we retain disturbance rejection, consider the dynamic system $\frac{dy}{dt} = f(u, w, y, d)$. At steady-state, where $f(u, w, d, y) = 0$, we have the static relationship $y = f_0(u, w, d)$. Assume now that we apply the ideal static transformed input $v_0 = f_0(u, w, d)$ to this dynamic system. Then $u = f_0^{-1}(v_0, w, d)$ and the transformed dynamic system becomes

$$\frac{dy}{dt} = f_t(v_0, w, y, d) \tag{81}$$

An example of such a transformed system is given by Eq. 42 for the mixing process in Example 5. At steady-state we have $dy/dt = 0$, which implies that $f_t = 0$ at steady state, independent of the values of d (and w). Since we have $y = v_0$ at steady state, we then know that $f_t = 0$ when $y = v_0$, independent of the values of d (and w). Assume that we are initially at steady-state, so that we have $y = v_0$ and $f_t = 0$. First, consider a disturbance d and assume that we keep the transformed input v_0 constant. Then, since we start from $y = v_0$ and we keep v_0 constant, we have that f_t remains at 0 and from Eq. 82 we have that $\frac{dy}{dt} = 0$, and the system will remain at steady state with $y = v_0$. In conclusion, we have perfect dynamic disturbance rejection for the dynamic model $dy/dt = f(u, w, y, d)$ in Eq. 18 with the use of ideal transformed variables v_0 based on a static model.

However, if we are not initially at steady state then we will not have perfect dynamic disturbance rejection with v_0. For example, if we are in a transition between steady states, due to a change in the transformed input v_0 (made by the outer controller), then an output y_i which is not at steady-state, will not be dynamically independent of disturbances. However, if the changes for v_0 are infrequent or on a slow time scale, then for practical purposes we will have perfect disturbance rejection when applying v_0 to the dynamic system (18).

To prove that we in many cases have decoupling, note the equations (81) is a set of equations of the form

$$\frac{dy_i}{dt} = f_{t,i}(v_0, w, y, d) \tag{82}$$

We make the *additional assumption* that for each equation, $f_{t,i} = 0$ when $y_i = v_{0,i}$. This may not always hold, but it is satisfied for the models studied in this paper. Next, consider a change in a single transformed input $v_{0,i}$ with all the other transformed inputs $v_{0,j} (j \neq i)$ constant. Since we start from a steady state with $y_j = v_{0,j}$ and we keep $v_{0,j}$ constant, we then have that $f_{t,j} = 0$ and from Eq. 82 we have that $dy_j/dt = 0 (j \neq i)$. This means that we have a decoupled response where only y_i changes in response to the change in $v_{0,i}$.

In summary, if we initially are at steady state, then v_0 achieves disturbance rejection and in many cases decoupling. Thus, the main advantage of using v_A (based on the dynamic model Eq. 18; similar to feedback linearization) rather than v_0 (based on a static model) when applied to the dynamic system Eq. 82 is that v_A linearizes the transformed system, also dynamically. Since v_A depends on y, this gives justification for referring to this approach as "feedback linearization".

6.4. Comparison with feedback linearization

The use of transformed variables v_A based on a dynamic model is a special case of feedback linearization to systems of relative order 1 from u to y. For the scalar case (with one input u and one output v) it only allows for one nonlinear differential equation, $\frac{dy}{dt} = f(u, y, d)$ and it transforms it into a first-order linear system, $\frac{dy}{dt} = Ay + Bv_A$. Nevertheless, we have shown in this paper that this may be very useful for practical applications in process control.

Compared to the traditional feedback linearization literature, we have put the main emphasis on the nice properties related to feedforward control and decoupling. In the feedback linearization literature, the main emphasis is usually on the linearization effect. In any case, an important advantage of the feedback linearization literature is that it provides a rich theoretical basis for introducing the transformed variables v_A.

Feedback linearization allows for considering higher-order system with m nonlinear differential equations (m state variables x), and it transforms it into a m'th order chain of m first-order systems from v to y. This sounds very nice, but for process control. it is usually not very helpful, and there are hardly any reports of it having being used. First, the feedback linearization theory assumes that all the states x are measured, which is often not satisfied in process control applications. In any case, even if we can measure all the states, the resulting implementation tends to become complicated, and may not be worth the effort. An alternative approach, which is simpler but less general than feedback linearization, is to a chain of input transformations based on models for the measured states w (Skogestad et al., 2022). Compared to the present paper, this allows for using an exact model-based inverse rather than an inverse generated by the slave v-controller and may improve the disturbance rejection in some cases. However, also this implementation gets rather complex, and it seems that only in rare cases will this benefit be worth the extra complication and effort.

6.5. Simplicity and alternative approaches

We have stressed the need to keep things simple. This is usually not an objective in academic papers, but in practice simplicity is important for many reasons. First, it makes it possible to build a control system of smaller parts (blocks) which may designed and tuned independently. Second, it is easier to understand and modify by engineers and operators, and it reduces errors in the implementation.

As just mentioned, it is possible to generalize the transformed inputs presented in this paper to higher-order dynamic models using the theory of feedback linearization. However, for more general cases, there are other control approaches that may be more suitable than feedback linearization, for example, nonlinear model predictive control which allows for taking into account much more general control objectives, including input and output constraints.

7. Summary and conclusion

In this paper we use the concept of transformed inputs $v = g(u, w, y, d)$ to provide a systematic approach to derive model-based nonlinear calculation blocks and cascade control schemes which are frequently used for industrial processes.

The starting point is often a nonlinear static model, $y = f_0(u, d)$. The ideal static transformed input is then simply the right-hand side of the model, $v_0 = f_0$, that is, we have $g = f_0$ where we note that g in the static case does not depend on the outputs y. For the ideal case, where all disturbances d that enter the model are measured and there is no model error, this gives at steady state the transformed system $y = v_0$, which is linear, decoupled and independent of disturbances.

In practical cases, the ideal transformed inputs v_0 may be used as a starting point to suggest simpler transformed inputs v. In such cases, some of the ideal properties are lost, but the transformed input v may still be very useful and greatly simplify the design of the outer controller C, which in any case is needed to handle model uncertainty and unknown or uncertain disturbances.

For implementation, we need to invert the transformation g to generate the physical input $u = g^{-1}(v_0, \dots)$. In some cases, we may use the exact model-based inverse in Figure 7a, but if the equations are complex, we may use feedback control as a "trick" to solve the equations by using the cascade implementation in Figure 7b with a slave v-controller.

The model may often be written in a simpler form, $y = f_{0w}(u, w, d)$, by allowing for the use of measured dependent variables (states) w as parameters in the model equations. In such cases, the cascade implementation in Figure 7b is usually preferred. First, it may happen that the function f_{0w} does not depend explicitly on the input u and then it's not possible to use a model-based inverse. Second, there is a potential problem with internal instability if we use the model-based inverse when g depends on w. Internal instability may occur if the indirect (dynamic) effect of u on v through w is large compared to the direct (static) effect of u on v.

It is also possible to derive ideal transformed inputs, v_A, based on a dynamic model, $\frac{dy}{dt} = f(u, y, d)$; see Eq. 32. This approach is closely related to the theory of feedback linearization. At first sight, this seems to be a much more powerful approach than the static variables v_0, as it gives a transformed system $\frac{dy}{dt} = Ay + Bv_A$ which is linear, decoupled and independent of disturbances, also dynamically. However, the benefit is usually small. First, the class of dynamic systems described by $\frac{dy}{dt} = f(u, y, d)$ is rather limited. For example, for a single-input single-output processes, it allows for only one differential equation with no direct effect from the input u to the output y (that is, no zeros are allowed). Second, we have found that the ideal static transformed input v_0 performs almost as well as v_A for this class of systems. In particular, it maintains perfect dynamic disturbance rejection if the system initially is at steady state. Third, a disadvantage with v_A is that it is more complex, and in particular that it requires choosing a reasonable value for the tuning parameter A. Simply setting $A = 0$, as is normally recommended in feedback linearization, is normally not a good choice as the resulting transformed system is drifting for unknown disturbances. The main advantage with v_A compared to v_0 is that it linearizes the system dynamically. This may simplify the design of the outer controller C in some cases. A second advantage for cases where the model depends on measured states w is that the parameter A may be used to specify the dynamics of the transformed system.

When we use the static transformed input v_0, there is no parameter to affect the dynamics of the transformed system. For cases where v_0 is independent of w, this means that the transformed input is purely feedforward and the transformed system will have the same dynamics as the original system, which usually is a good choice. However, when we introduce w-measurements in v_0 , the resulting feedback gives changes in the dynamics. For the heat exchanger in Example 6, the response became worse (slower), but there may exist cases where the dynamics in w make the response faster than the original system. Nevertheless, we generally recommend that the engineer starts with static models when deriving transformed inputs.

The use of transformed inputs v may in theory provide no offset at steady state, but this is based on feedforward control and assumes an exact model and perfect measurements of the disturbances. Therefore, we generally need to add an outer controller C which manipulates v to control the output y. Single-loop PID-controllers are usually sufficient because the response from v to y is linear and decoupled, at least in the ideal case. The objective of the outer controller is to correct for errors in the model and measurements and to reject unmeasured or unmodelled disturbances. The outer controller should include integral action to get offset-free control at steady state.

8. Appendix

8.1. Appendix 1. Tuning parameter A for ideal dynamic transformed inputs v_A.

To prove that the choice $A = \mathrm{diag}(\tilde{A}) = \mathrm{diag}(\frac{\partial f}{\partial y})_*$ in (85) minimizes the effect of y on v_A, consider the "original" nonlinear model $\frac{dy}{dt} = f(u, y, d)$, which can be linearized to get

$$\frac{dy}{dt} = df = \tilde{A}dy + \tilde{B}du + \tilde{B}_d dd \tag{83}$$

where the $\sim$ variables correspond to the linearized dynamics of the original system. We have that $\tilde{A} = (\partial f/\partial y)_*$, $\tilde{B} = (\partial f/\partial u)_*$ and $\tilde{B}_d = (\partial f/\partial d)_*$. where the evaluation of the derivatives is performed at the nominal point of operation, denoted by $*$. Recall from (24b) that the dynamics of the transformed system are given by $dy/dt = Ay + Bv_A$. Thus, if we choose $A = \tilde{A}$ then the transformed system will locally (close to the nominal operating point $*$) have the same dynamics as the original system in (83). Furthermore, from from Eq. 19 the linearized transformed input becomes

$$dv_A = B^{-1}(df - Ady) = B^{-1}(\tilde{B}du + \tilde{B}_d dd) \tag{84}$$

and we find that dv_A is independent of dy. Thus, with the choice for A in (27), there is no feedback from y on the transformed input v_A at the nomonal point $*$. For the multivariable case, to get a decoupled response, we may choose A equal to the diagonal elements of the A-matrix of the original system,

$$A = \mathrm{diag}(\tilde{A}) = \mathrm{diag}\left(\frac{\partial f}{\partial y}\right)_* \tag{85}$$

For the multivariable case, this will not exactly keep the original dynamics and there will be some feedback from y to v at the nominal point. However, it provides a good comprise been decoupling and minimizing the feedback from y. In any case, the exact value for A should not be overemphasized, since we can change the closed-loop dynamics by design of the outer controller C.

8.2. NEW PAPER: Ratios and sums of flows as transformed inputs

from Nick: better places before the motivating example. Sigurd: Yes, I agree, but I think that it is even better to put it in a separate paper.

The two most common transformed inputs in process control are sums/differences of flows and ratios of flows. Indeed, we have seen that they appear in most of the examples if we derive ideal transformed variables v_0 or v_A, although usually in more complicated forms. An example of decoupling involving ratios and sums of flows was given in the motivating mixing example in Section 2.

Ratios of flows (e.g., $v = \frac{F_1}{F_2}$) is probably the most common transformed input in process control, and it is frequently introduced based on engineering intuition. Feedforward action results when the ratio involves an input and a disturbance ($v = u/d$) and decoupling when two flows are used ($v = \frac{u_1}{u_2}$). We will here show that ratios can be used for systems that satisfy the *"scaling property"* without actually needing a detailed model of the property y (e.g., composition, temperature, color or viscosity) that we want to control.

from Nick: viscosity is not a good example because it's not a linear mixing property, but it varies log/log. Answer from Sigurd: Actually, we need not assume that it mixes linearly; we only need to assume that we get the same value of the intensive variable (here viscosity) when all flows are increased by the same ratio.

The scaling property simply says that if we increase all the flows in the process by the same factor then the property variable y will remain constant at steady state. The scaling property applies to mixing processes and more generally to processes in thermodynamic equilibrium. For example, for a mixer with three feed flows, keeping two ratios constant will keep all the intensive variables y in the mixer constant at steady state.

Mathematically, the function f for the dependence of the property (intensive) variables y on the flow (extensive) variables F_i is assumed to be homogeneous to the degree 0. For example, consider a intensive variable y (output) which depends on two intensive variables (c_1 and c_2) and two extensive variables (F_1 and F_2). At steady state, we then have for systems that satisfy *the scaling property*

$$y = f(c_1, c_2, kF_1, kF_2) = k^0 f(c_1, c_2, F_1, F_2) \quad (86)$$

Note here that $k^0 = 1$ (because the homogeneity degree is 0), which means that y remains constant when all the flows F_i (extensive variables) are all increased by the same factor k. In general, if we have n independent flow variables F_i (extensive variables), then keeping $n-1$ ratios constant, will keep the dependent intensive variables y constant, provided that the independent intensive variables (c_1 and c_2) are constant.

This scaling property applies to the static operation of many equilibrium processes, for example, an equilibrium distillation column with constant stage efficiency or an equilibrium reactor. A distillation column operating at fixed pressure has three independent flows at steady state (including the through-put), and fixing any two flow ratios in the column (e.g., reflux ratio L/D and boilup-to-feed ratio V/F) will keep all intensive variables (e.g., compositions and temperatures) in the column constant at steady state independent of the feed rate to the column. This assumes that the feed composition and quality is constant.

The scaling property only holds if *all* the ratios between flows (extensive variables) are kept constant. For example, for a distillation column, if we have constant heat input (boilup) V, then to keep constant product composition y in the top of the column, the change in reflux L to a change in feedrate F will be less than that given by keeping a constant ratio L/F. Thus, in this case ratio control should not be used, unless the boilup V is also increased proportionally to F at steady state.

The scaling property does not apply to a heat exchanger, because here the heat exchange area A (which is an extensive variable) is constant, so when all flows are increased by the same factor, the heat transfer becomes less effective and the exit temperatures y will change. It also does not apply to a non-equilibrium reactor, because here the conversion depends on the reactor volume which is not varied in proportion to the feedrate.

For systems that satisfy the scaling property, we know that constant flow ratios give a constant value for the property variable y at steady state, without needing a model for the the property y we want to control. This can be very useful for practical applications. For example, y could be viscosity, for which we may not have a model. However, if we want to achieve feedforward action to independent intensive variables (y_1 and y_2, e.g. feed compositions), then we need a more detailed model. This will result in the ideal transformed inputs v_0 and v_A as derived in this paper, which may often be interpreted as generalized flow ratios. Nevertheless, in practice, we may not have measurements of y_1 and y_2 or we may lack a model, and then choosing ratios as transformed inputs (e.g. $v = \frac{F_1}{F_2}$) based on the simpler scaling property, may provide disturbance rejection with respect to flow disturbances or contribute to decoupling.

Note that the scaling property applies to the steady-state behavior, and to achieve better dynamic behavior it is common to introduce some dynamic elements. For example, if we in a distillation column use the flow ratio V/F, then we may delay the measurement of the feedrate F (disturbance) because it takes some time for a change in F to reach the bottom of the column where we want to control the composition y and where the boilup V enters.

Next, consider sums and differences of flows (e.g. $v = F_1 + F_2 - F_3$). They appear when we want to achieve feedforward action or decoupling when the controlled variable (y) is level (liquid holdup) or pressure (gas holdup). This follows directly from the material balance for total holdup ($y = m$), for example, for the case with two inflows and one outflow,

$$\frac{dm}{dt} = F_1 + F_2 - F_3$$

If the process is integrating, that is, if F_1, F_2 and F_3 are independent of m, then it is reasonable to choose $A = 0$ to get an integrating transformed system and with $B = 1$ the ideal transformed input is $v_A = F_1 + F_2 - F_3$.

9. Transformed outputs

It is also possible to define transformed outputs

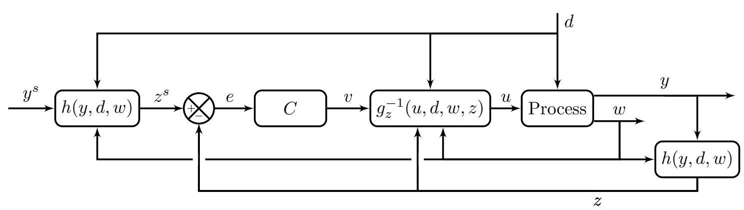
$$z = h(y, w, d) \tag{87}$$

where y are the outputs that we want to control at a given setpoint y_S and h is a static function of our choice. However, we have already shown that we, by use of transformed inputs v alone, can make the transformed system from v to y linear, decoupled and independent of disturbances. How can the transformed system be any simpler by introducing also transformed outputs? To justify the introduction of transformed outputs z, we therefore also include simplicity of the implementation as a secondary objective. So then we get:

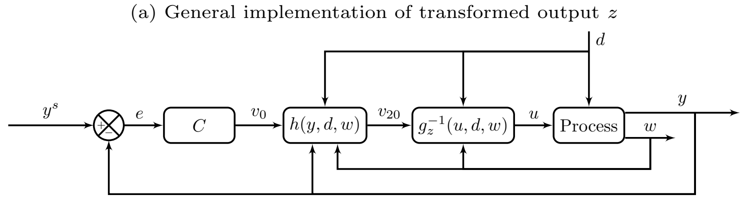
The objective for introducing transformed inputs v is to simplify the control task as seen from the outer controller C, while transformed outputs z may be *introduced to simplify the implementation of the transformed inputs.* This means that when we introduce transformed outputs z, then the transformed inputs will be in terms of these variables, that is,

$$v_z = g_z(u, w, z, d) \tag{88}$$

The implementation of combined transformed inputs and outputs is shown in Figure 14a where we note the controller C is controlling the transformed outputs z rather than the (physical) outputs y for which we have a setpoint y^s. However, since we send both y and y^s through the same static transformation h, we will achieve $y = y^s$ at steady state. Also note from Figure 14a that the input transformation g_z needs to be inverted (or approximately inverted using one of the three options in Figure 7a), whereas inversion is not necessary for the output transformation h.



(a) General implementation of transformed output z



(b) Alternative implementation of transformed output when the ideal transformed input is based on a static model.

Figure 14: System with both input and output transformation

Misprint. 1. u should be replaced by v and vz0 in argument lists for gz-1 (two places). 2. In (b) there should be no arrow from y into the block h. The block h in (b) should be h(y=v0,d,w)

Ideal transformed inputs and outputs from dynamic model. The idea is that it is easier to write the dynamic model in terms of the transformed outputs z rather than in terms of the outputs y. Assume that the dynamic model for the process can be written in the form,

$$\frac{dz}{dt} = f_z(u, w, z, d) \tag{89}$$

where $z = h(y, w, d)$ is the transformed output. From (32) the ideal transformed input is $v_{zA} = B^{-1}(f_z - Az)$, and the transformed system as seen from the controller C becomes

$$\frac{dz}{dt} = Az + Bv_{zA} \tag{90}$$

which is decoupled, linear and independent of disturbances. This simplifies the design of the outer controller C. However, note from Figure 14a that disturbances that effect the transformed outputs $z = h(y, d)$ will only be counteracted if the feedback controller C is implemented.

Ideal transformed inputs and outputs from static model. For the static case, we have assumed that we may write the model (at least formally) in the form $y = f_0(u, w, d)$ with only y on the right hand side. However, in some cases it is much simpler to write the static model in the more general form

$$\underbrace{h(y, w, d)}_{z} = \underbrace{g_z(u, w, d)}_{v_{z0}} \tag{91}$$

where we note that the outputs y are collected on the left-hand side and the inputs u are collected on the right-hand side. The main idea is that the function g_z on the right-hand side is easy to invert. Assume now that we want to derive a transformed system from v_0 to y which at steady state satisfies $y = v_0$. This becomes very easy if we select the left-hand side of the static model Eq. 91 as the the transformed output, $z = h(y, w, d)$, and the right hand side as the transformed input, $v_{z0} = g_z(u, w, d)$ (corresponding to the common choice $B_0 = I$). Note that the task is to derive an input u that gives the desired output $y = v_0$. For a given value of $y = v_0$, the corresponding transformed output is $v_{z0} = h(v_0, w, d)$ which is the left-hand side of Eq. 91. The value of u that corresponds to this left-hand side is obtained by inverting the right-hand side to get $u = g_z^{-1}(v_{z0}, w, d)$, which may be implemented as in Figure 14b, and gives the desired transformed system $y = v_0$.

Note that for static case (v_0), one should implement the transformed output as shown in Figure 14b. Compared to the implementation in Figure 14a, it has the advantage that disturbances that effect the output transformation $z = h(y, w, d)$ are counteracted also without the outer controller C.

To Cristina from Sigurd: You use the other implementation in the DYCOPS paper. I think it would be a little better with this simpler implementation, the offset issue would probably go away. But otherwise probably minor changes. —¿¿ will try

Note that the two implementations in Figure 14 are not equivalent, even for the case when the process is static.

9.1. Example 8. Heater with complicated thermodynamics

A common choice for the transformed output is the specific enthalpy $z = H$ [kJ/kg], which is used for cases where we want to control the temperature, $y = T$. The reason is that the energy balance is easily written in terms of enthalpy.

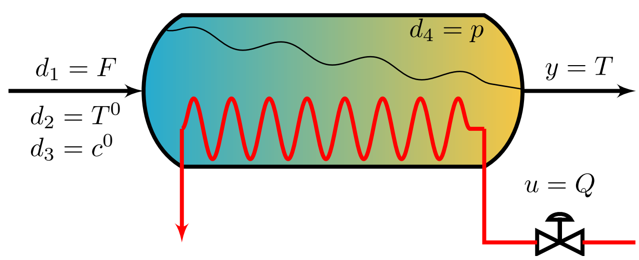


Figure 15: Flowsheet of an electric heater. A multicomponent liquid stream is heated to two-phase product with desired temperature ($y = T$) by manipulating the heat ($u = Q$).

Consider an electric heater (Figure 15 where a multicomponent liquid mixture is partly vaporized to obtain a two-phase product. The physical input for control is the heat input, $u = Q\,$[kW], and the output is the outflow temperature, $y = T$. Disturbances are in the feed (flowrate $F\,\mathrm{kg\,s^{-1}}$, temperature T_0

and composition c_0) and in the operating pressure p.

$$d = [F \; T_0 \; c_0 \; p]$$

It is easy to formulate the energy balance in terms of the specific enthalpy $H \,\mathrm{kJ\,kg^{-1}}$. By assuming constant mass holdup $m \,\mathrm{kg}$ and perfect mixing inside the heater, the dynamic energy balance becomes,

$$m\frac{dH}{dt} = F(H - H^0) + Q \tag{92}$$

The corresponding static energy balance becomes

$$H = H^0 + Q/F$$

The enthalpy depends mainly on temperature, but also on composition and pressure.

$$H = h(T, p, c); \quad H^0 = h(T^0, p, c^0)$$

This function h is given by thermodynamics and is in most cases available only numerically or as a look-up table. Note that we have used the symbol h to denote this function, because it is the same as the function h that we will use to define the transformed output z.

Transformed input in terms of $y = T$. It is complicated to find the ideal transformed input in terms of $y = T$, both for v_A in the dynamic case and for v_0 in the static case. For the dynamic case, we need to derive a model in terms of $\frac{dy}{dt} = \frac{dT}{dt}$, so we would need to differentiate the right hand side of the energy balance to get

$$\frac{dH}{dt} = \frac{\partial h}{\partial T}\frac{dT}{dt} + \Sigma_i \frac{\partial h}{\partial d_i}\frac{dd_i}{dt}$$

where d_i represents the disturbances. There are many problems here. First, performing the differentiation, $\frac{\partial h}{\partial T}$ is difficult, even with an analytic expression for the function h, and even more difficult if h is represented by a table that requires interpolation. Second, we need to find information about the time derivatives of the disturbances. Third, even if we could do all this and find an expression for $\frac{dy}{dt} = f(u, y, d)$ and get $v_A = B^{-1}(f - Ay)$ we would still in the end need to invert this expression to find u. Of course, it could be done numerically, but still it would be a major effort which few would want to do, and which is very likely to contain errors. Fortunately, we show below that everything becomes very simple if we introduce the transformed output $z = H$.

For static case, finding an expression for the ideal transformed variable v_0 is complicated. It will involve inverting the function (or table) h to derive an expression for $y = T$ as function of H, p and c; $T = h^{-1}(H, p, c) = h^{-1}(H^0 + Q/F, p, c)$. This will be difficult if the function h is complicated or just a table. The next step is to invert this relationship to find the corresponding $u = Q$ for a given value of $y = T$. (Comment to myself: But if we invert h-1 then we are back to h; so not so complicated afer all?). However, in the static case, it is not necessary to explicitly find an expression for v_0, because what we really need to find is an expression for $u = Q$ for a given value of $y = T$. This is quite simple in this case if we introduce the transformed output $z = H$, as shown below.

Transformed input in terms of $z = H$. It is very simple to derive the transformed input in terms of the transformed output $z = H = h(T, p, c)$. For the dynamic case, the energy balance (92) becomes

$$\frac{dz}{dt} = f_z(u, d, z) = \frac{1}{m}(F(z - H_0) + u)$$

where we have introduced $z = H$ and $u = Q$. From (19) we derive the ideal transformed input $v_{zA} = B^{-1}(f_z - Az)$, which is easily inverted to find the physical input

$$u = g_z^{-1}(v_{zA}, z, d) = m\,(Bv_{zA} + Az) - F(z - H_0)$$

where A and B are free to choose. This can be implemented as in Figure 14a where v_{zA} is the output from the controller.

Similar, for the static case, we define the transformed output as $z = H = h(T, p, c)$. The static energy balance then becomes $z = H_0 + Q/F$, so the transformed input is $v_{z0} = H_0 + Q/F$ which may be inverted to give

$$u = g_z^{-1}(v_{z0}, d) = F(v_{z0} - H_0)$$

which can be implemented as shown in Figure 14b.

10. Unused

10.1. Comparison with other approaches

Balchen's END. (Probably not worth mentioning as it does seem to be used). Balchen has presented an approach which, at least for simpler cases, is very similar to the feedback linearization and also to the approach in this paper. What he calls $\dot{z}$ is very similar to our v_A (but he selects $A = 0$) and his D-matrix is for simpler cases the same as our B-matrix. However, he allows for any state space model and then D is a fat matrix which somehow gets contribution from all states. However, in practice it seems he only picks out the equations involving the outputs y, so it becomes similar to what we do, when we consider the other states as w-variables (estimated states). But he also adds a small term in D to avoid some problems. This is distillation application in JPC in 1995 where he has a small d2,23 term which he adjusts by trial and error (see the very end of the paper). Otherwise, he only picks out the differential equations involving y.

10.2. Example 6: Stability issues for units in series with recycle

10.3. Example 7: choice of the tuning parameter A. Effect of unmeasured disturbances (Example 3 continued)

10.4. Example 8: Input saturation (Example 3 continued)

10.5. Example 2X. Implementation of input transformations using valve for level control

The objective of this example is to compare the three alternative implementations in Figures 7a to 7c on a very simple example process. We consider using the outflow to control the level $y = H$ in a tank. The true physical input u is the outlet valve position z, that is, $u = z$. From a mass balance for a tank with constant cross-sectional area and constant density, the model can be written as

$$\frac{dH}{dt} = \frac{1}{A_t}\left(q_{\mathrm{in}} - q(u)\right) \tag{93}$$

Here H [m] is the level, A_t [m^2] is the tank area, q_{in} [$\mathrm{m}^3\,\mathrm{s}^{-1}$] is the inflow and $q = q_{\mathrm{out}}$ [$\mathrm{m}^3\,\mathrm{s}^{-1}$] is the outflow. We have knowledge about how the outflow q [$\mathrm{m}^3\,\mathrm{s}^{-1}$] depends on z by the following valve equation:

$$q(u) = F(u) k_V \sqrt{\frac{p_1 - p_2}{\rho}} \tag{94}$$

Here, $F(u)$ is the valve characteristic, k_V [m^2] is the valve constant, ρ [$\mathrm{kg\,m}^{-3}$] is the liquid density, and $p_1 - p_2$ [$\mathrm{N\,m}^{-2}$] (disturbance d) is the pressure drop over the valve. We normalize the valve position u to be in the range 0 (closed) to 1 (fully open) and then $F(u)$ increases from 0 to 1 as u increases from 0 to 1. For a linear valve, $F(u) = u$. We assume that we can measure the level ($y = H$), but this measurement has some delay and is a bit noisy. We also have a measurement of the outflow ($w = q$) which we may use if desired.

10.5.1. No input transformation

The simplest solution is to not make use of the extra measurement $w = q$ or of the model Eq. 94. We directly control the level using a controller C,

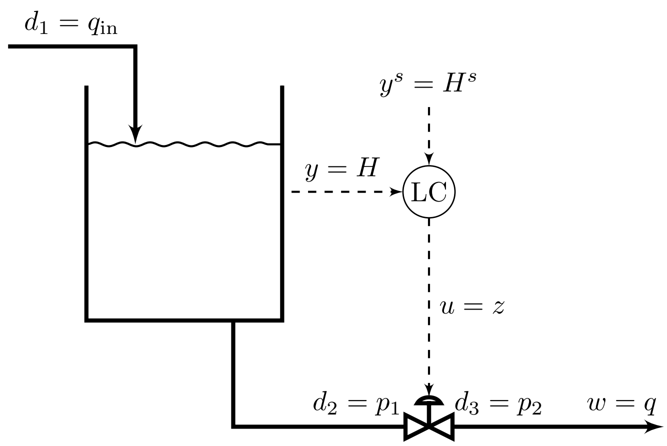


Figure 16: Level control with no input transformation.

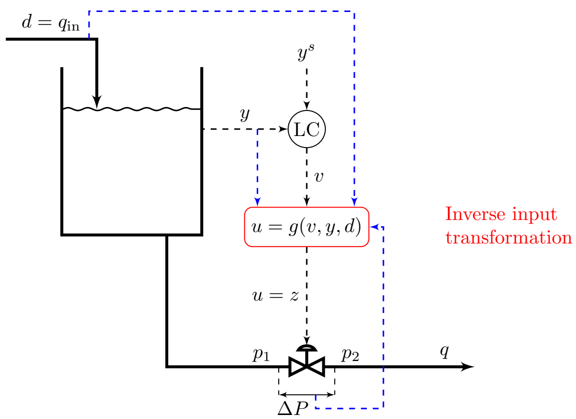
for example a PI-controller, which adjusts the valve position u. This solution is shown in Figure 16. This solution is simple, but it may not result in tight level control, especially if there is delay and noise in the measurement of $y = H$. Thus, the level may vary if there are disturbances in q_{in}, p_1 and p_2. Furthermore, nonlinearity in the valve characteristic $F(u)$ may give a low process gain and thus slow control when $F(u)$ is in a "flat" region, that is, when q is insensitive to changes in u. Typically, this will be when the valve approaches fully open or fully closed.

10.5.2. With input transformation

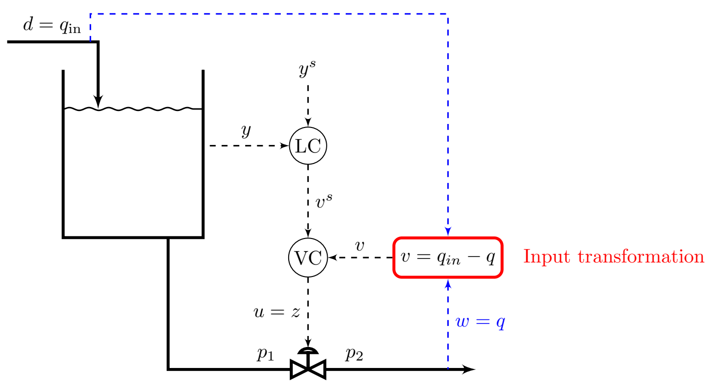
Based on the dynamic model in Eq. 93, the most obvious transformed input is the right-hand side of Eq. 93 multiplied with a constant $\frac{1}{B}$ (where B is a parameter that we introduce to generalize the method and that we can choose),

$$v = \frac{1}{A_t} \left(q_{\text{in}} - q(u)\right) \frac{1}{B} \tag{95}$$

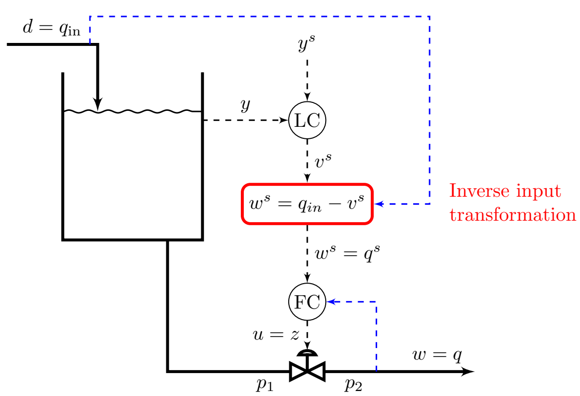
In terms of the transformed input the model simply becomes an integrator



(a) Exact inverse transformation (inverting model).



(b) Inner controller for $v = q_{in} - q$.



(c) With inner controller for $w = q_{\rm out}$ (flow controller).

Figure 17: Three alternative implementations for level control. The input transformation provides feedforward control from q_{in} thus linearization of the valve. The transformation also provides disturbance rejection from p_1 and p_2 (by feedforward in *(a)* and through feedback in *(b)* and *(c)*.

(as in the method of feedback linearization (Isidori, 1995),

$$\frac{dH}{dt} = Bv \tag{96}$$

and "magically" the disturbances and nonlinearity have disappeared as seen from outer controller C, which uses v to control $y = H$. Comparing the original process in Eq. 93 and Eq. 94 with u as the input, with the transformed system in Eq. 96 with v as the transformed input, we see that there are two main advantages. First, the effect of disturbances in p_1 and p_2 on $y = H$ are eliminated. Second, the possible nonlinearity in the valve characteristic $F(u)$ is eliminated. More generally, v in Eq. 95 follows from the systematic method in Section 4 by selecting the parameter $A = 0$. In the following, we select the parameter $B = \frac{1}{At}$ such that the transformed variable simply becomes

$$v = q_{\text{in}} - q(u) \tag{97}$$

To implement the transformed input v in practice, we need to generate from a given value of v the corresponding physical input u. As described above, there are two main options, model-based and measurement-based (cascade).

10.5.2.1. Exact implementation: Inverting the valve model. We here use the implementation in Figure 7a. Inserting the valve equation Eq. 94 into Eq. 97 gives

$$v = \underbrace{q_{\text{in}} - F(u)k_V\sqrt{\frac{p_1 - p_2}{\rho}})}_{g(u,d)} \tag{98}$$

Eq. 98 is on the form $v = g(u, d)$ with $d = [q_{\text{in}}, p_1, p_2]$. Solving Eq. 98 with respect to $u = z$ gives

$$u = \underbrace{\frac{F(u)^{-1}(q_{\text{in}} - v)}{k_V\sqrt{\frac{p_1-p_2}{\rho}}}}_{g^{-1}(v,d)} \tag{99}$$

where $F(u)^{-1}$ denotes the inverse of the valve characteristic $F(u)$. Note that

$F(u)$ does not need to monotonically increasing (Lee et al., 2016). The solution is shown in the flowsheet in Figure 17a. However, the inverse transformation in Eq. 99 requires a good model and it also requires measurements of the disturbances p_1 and p_2. Therefore, rather than inverting the valve equation, the by far more common option is to measure the flow $w = q$ and use an inner flow controller.

10.5.2.2. Alternative cascade implementations: Using the flow measurement (cascade control). We here make use of the extra measurement $w = q$. The transformed input is then

$$v = \underbrace{q_{\text{in}} - w}_{g(w,d)} \tag{100}$$

Here, v does not depend explicitly on u, so we need to use one of the two cascade implementations in Figures 7b and 7c. The specific implementations for our level control problem are shown in the flowsheets in Figures 17b and 17c. For the cascade control of v, the controller C_v is actually a flow controller because v is the difference between two flows. For the cascade control of w, C_w is of course a flow controller since $w = q_{\text{out}}$ is a flow. For cascade control of w, we need to invert Eq. 100 with respect to $w = q$ which gives the "inverse static transformation"

$$w^s = \underbrace{q_{\text{in}} - v^s}_{g_w^{-1}(v,d)} \tag{101}$$

Note that we use superscript s on w and v, because w^s and v^s are the setpoints for w and v, respectively. The main advantages with the two cascade implementations based on measuring $w = q_{\text{out}}$ compared to the exact implementation in Eq. 99, is that we do not need to measure the two disturbances p_1 and p_2 and also to do not need to invert the valve equation $F(u)$ which may be highly uncertain. However, to achieve the desired disturbance rejection and linearization, we must for the cascade implementations assume that the inner flow controller (C_v or C_w) can be made fast compared to the expected process dynamics for $y = H$ and compared to the outer controller C. This is most likely possible, since the valve response from u to $w = q_{\text{out}}$ is usually very fast, that is, the process is essentially static with a time constant (τ) close to zero. From the SIMC PID rules (Skogestad, 2003), a pure I-controller may then be a good choice for the flow controller.

10.6. Cascade implementation

Alternatively to implementing Eq. **??**, we may use the cascade implementation (Figure 7b).

Because this is a *two-input two-outputs* system with both physical inputs u_1 and u_2 appearing in the expressions for the transformed inputs v_1 and v_2, we need to decide on the pairing of the two inner loops $u - v$. The two options are:

Pairing a $u_1 - v_1$ and $u_2 - v_2$ or the reverse

Pairing b $u_1 - v_2$ and $u_2 - v_1$.

Without constraints, the response from v to y is linear regardless of which $u - v$ pairing we choose. However, the pairing decision becomes critical if one of the inputs reaches its lower or upper physical saturation constraints. In this case, we loose control of one of the CVs. The solution here is to pair the input that is less likely to saturate with the transformed input corresponding to the output that is more important to control. This is in accordance to the input saturation rule (Reyes-Lúa & Skogestad, 2020). In the simulations in this work, we do not impose physical saturation limits for the input, but this is an important practical consideration, and one of the potential advantages of using a cascade imnlementation. END CHANGE

Figure 18 shows only the simulations for *Pairing 1*, and the the reverse *Pairing 2* behave similarly. The inner loop controller C_v (Figure 7a) are integral controllers tuned with a closed-loop time constant $\tau_C = 1\,\text{s}$ using the SIMC-tuning rules (Skogestad, 2003).

Compared to the response for exact implementation in Figure 10, the responses for cascade implementation in Figure 18 are no longer independent of disturbances and show perfect response to setpoint changes dynamically. The reason is that the inner controller setting v cannot be made infinitely fast.

HM.... the cascade responses do NOT look good... There is a lot of spiking here. How does this depend on the tuning? The responses also look strange. How can y1=F be spiking at t=100 (from 12 to more than 14) when F1 goes down a little and F2 a little up, so it seems F1+F2 changes by at most 1. Anyway, this is not so important, the responses are still poor. To reduce the coupling we may consider tuning the controller for v1 fast (flow control, tauc=0.1) and the controller for v2 slower(ratio control for temperature, tauc=1).

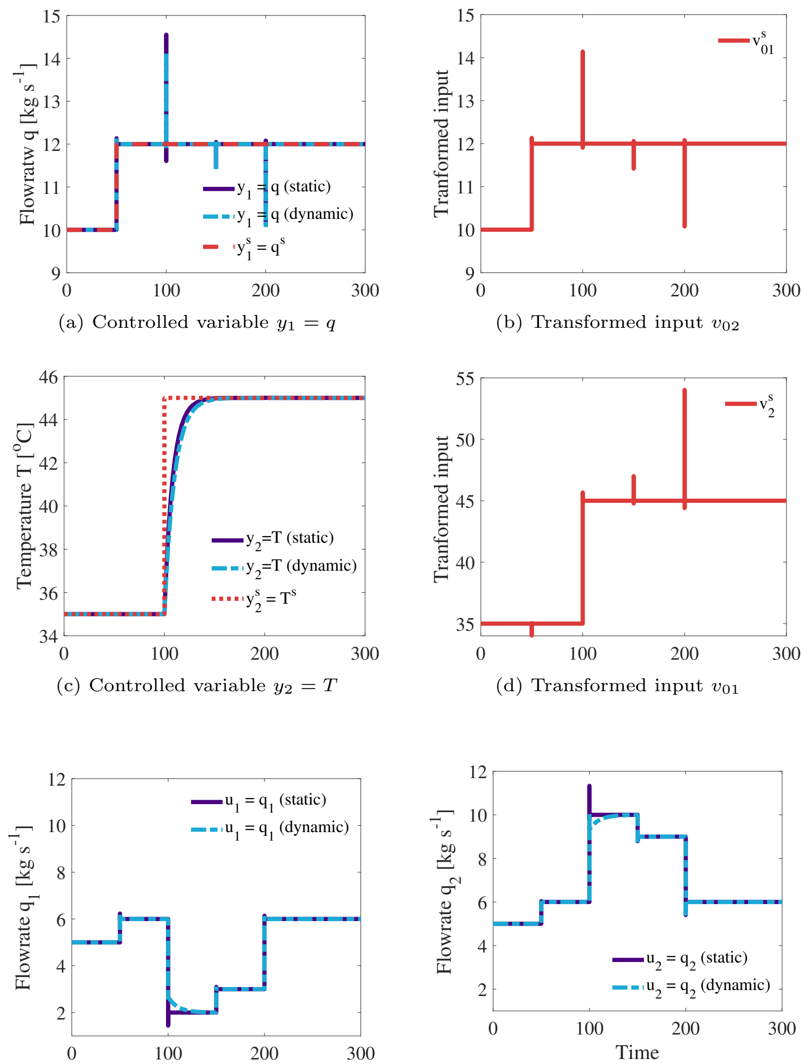
- Example 1: Decoupling of mixing process
- Example 2: Mixing process with valve positions as physical inputs
- Example 3: Nonlinear level
- Example 4: Heated tank
- Example 5: Mixing (Example 1 revisited)
- Example 6: Heat exchanger
- Example 7: Tank with Heat exchanger (w implemented using only feedback)

Acknowledgements

This work is partly funded by HighEFF (Centre for an Energy Efficient and Competitive Industry for the Future). The authors gratefully acknowledge the financial support from the Research Council of Norway and user partners of HighEFF, an 8 year Research Centre under the FME-scheme (Centre for Environment-friendly Energy Research, 257632/20).

References

Baldea, M., & Daoutidis, P. (2007). Control of integrated process networks—a multi-time scale perspective. *Computers and Chemical Engineering*,



(a) Controlled variable $y_1 = q$

(b) Transformed input v_{02}

(c) Controlled variable $y_2 = T$

(d) Transformed input v_{01}

(e) Physical input $u_1 = F_1$

(f) Physical input $u_2 = F_2$

Figure 18: Simulation response for the mixing process in Example 3 to the four step changes in Section **??** using the static transformation and cascade implementation (Figure 7a) with the *Pairing 1:* $u_1 - v_1$ and $u_2 - v_2$.

31, 426–444. URL: https://www.sciencedirect.com/science/article/pii/S009813540600130X. doi:https://doi.org/10.1016/j.compchemeng.2006.05.017. ESCAPE-15.

Bastin, G., & Dochain, D. (1990). *On-line Estimation and Adaptive Control of Bioreactors*. Elsevier Science Publishers B.V.

Isidori, A. (1995). Nonlinear Control Systems.

Isidori, A. (2020). personal communication.

Khalil, H. K. (2015). *Nonlinear control*. (2nd ed.). Boston: Pearson.

Kravaris, C., & Chung, C.-b. (1987). Nonlinear State Feedback Synthesis by Global Input / Output Linearization. *AIChE Journal*, *33*.

Lee, J., Mukherjee, R., & Khalil, H. K. (2016). Output feedback performance recovery in the presence of uncertainties. *Systems and Control Letters*, *90*, 31–37.

Reyes-Lúa, A., & Skogestad, S. (2020). Systematic Design of Active Constraint Switching Using Classical Advanced Control Structures. *Ind. Eng. Chem. Res.*, *59*, 2229–2241. doi:10.1021/acs.iecr.9b04511.

Reyes-Lúa, A., Zotică, C., Das, T., Krishnamoorthy, D., & Skogestad, S. (2018). Changing between Active Constraint Regions for Optimal Operation: Classical Advanced Control versus Model Predictive Control. *Computer-aided chemical engineering*, *43*, 1015–1020.

Shinskey, F. G. (1981). *Controlling multivariable processes*. Instrument Society of America.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, *13*, 291–309. doi:doi:10.1016/S0959-1524(02)00062-8.

Skogestad, S. (2008). *Chemical and Energy Process Engineering*. CRC Press.

Skogestad, S., & Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design*. (2nd ed.). Wiley.

Skogestad, S., Zotică, C., & Alsop, N. (2022). Nonlinear input transformation for linearization, decoupling and disturbance rejection - part 2.

Soave, N., & Barolo, M. (2021). On the Effectiveness of Heat-Exchanger Bypass Control. *Processes*, *9*. URL: https://www.mdpi.com/2227-9717/9/2/244. doi:10.3390/pr9020244.

Zotică, C., Alsop, N., & Skogestad, S. (2020). Transformed Manipulated Variables for Linearization, Decoupling and Perfect Disturbance Rejection. *IFAC-PapersOnLine*, .