



## Review

## Transformed inputs for linearization, decoupling and feedforward control

Sigurd Skogestad<sup>a,\*</sup>, Cristina Zotică<sup>a</sup>, Nicholas Alsop<sup>b</sup><sup>a</sup> Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), 7491, Trondheim, Norway<sup>b</sup> Senior Process Control Engineer, Borealis AB, Stenungsund, Sweden

## ARTICLE INFO

## Article history:

Received 21 June 2022

Received in revised form 17 November 2022

Accepted 20 December 2022

Available online xxxx

## Keywords:

Nonlinear process control

Feedforward control

Model-based control

Cascade control

PID control

## ABSTRACT

This paper introduces powerful static input transformations which transform the original system (process) into a transformed system which is easier to control. The transformed inputs (controller outputs) may be implemented in many ways and under many names, for example, as ratio, feedforward and decoupling control, or more generally as nonlinear computation blocks. 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 is simply the right-hand-side of the model equations. The resulting transformed system is linear, decoupled and independent of disturbances. In some cases, use of extra measurements simplify the input transformation by replacing model equations. It is also possible to derive ideal transformed inputs from a dynamic model, which turns out to be a special case of a nonlinear control approach called feedback linearization. However, except for achieving linearization also dynamically, the benefits of using feedback linearization are small compared to using transformed inputs based on a static model. For implementation we need to invert the input transformation, and for this we may use an exact model-based inverse or an approximate feedback-based inverse. The latter leads to the use of cascade control.

© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## Contents

1. Introduction.....	114
2. Motivating case study: Decoupling of mixing process.....	115
2.1. Example 1: Mixing process with flowrates as physical inputs.....	116
2.2. Example 2: Mixing process with valve positions as physical inputs.....	116
3. Implementation of transformed inputs.....	118
3.1. Alternative A: Model-based inversion (Figs. 1 and 8(a)).....	118
3.2. Alternative B: Feedback inversion with slave $v$ -controller (cascade control) (Fig. 8(b)).....	118
3.3. Alternative C: Combined feedback- and model-based inversion with slave $w$ -controller (Fig. 8(c)).....	118
4. Derivation of ideal transformed inputs.....	118
4.1. Obtaining an ideal transformed system from a static process model.....	119
4.1.1. Choice of the tuning parameter $B_0$ .....	120
4.2. Obtaining ideal transformed input from a dynamic process model.....	120
4.2.1. Choice of tuning parameter $B$ .....	121
4.2.2. Choice of tuning parameter $A$ .....	121
4.3. Transformed input in terms of measured state variables $w$ .....	121
4.3.1. Dynamics of transformed system with measured state ( $w$ ) variables.....	122
4.3.2. Possible unstable zero dynamics and internal instability.....	122
4.4. Ideal static transformed variable $v_0$ applied to a dynamic system.....	122
5. Examples.....	122
5.1. Example 3. Ideal transformed inputs for mixing process (Motivating Example 1, continued).....	122
5.1.1. Ideal transformed input $v_0$ from static model.....	123

\* Corresponding author.

E-mail address: [Sigurd.Skogestad@ntnu.no](mailto:Sigurd.Skogestad@ntnu.no) (S. Skogestad).

5.1.2.	Comparison with engineering-based variables from Motivating Example 1.....	123
5.1.3.	Applying the ideal static transformed input $v_0$ to the dynamic system.....	123
5.1.4.	Ideal transformed input $v_A$ from dynamic model.....	123
5.2.	Example 4. Heated tank.....	124
5.3.	Example 5: Simple nonlinear level process.....	124
5.4.	Example 6: Heat exchanger.....	125
5.4.1.	Ideal transformed input $v_0$ based on full static model.....	126
5.4.2.	Transformed input $v_{0,w}$ based on parts of static model and measured state $w = T_2$ .....	126
5.4.3.	Implementation.....	126
5.4.4.	Simulations.....	127
6.	Discussion.....	127
6.1.	Outer controller $C$ for case with ideal transformed inputs.....	127
6.2.	Time scale separation for feedback implementation (alternative B).....	127
6.3.	Decoupling and disturbance rejection when ideal static transformed input $v_0$ is applied to the dynamic system in (18).....	128
6.4.	Chain of input transformations.....	129
6.5.	Comparison with feedback linearization.....	129
6.6.	Comparison with nonlinear MPC.....	130
6.7.	Transformed outputs.....	130
6.8.	Internal instability with exact inversion.....	130
6.8.1.	Example: Simple linear system with unstable zeros.....	131
6.8.2.	Linear stability analysis of transformed system.....	131
6.8.3.	Relationship between internal stability and stable zero dynamics for output $y$ (for systematic transformed inputs $v_0$ and $v_A$ ).....	132
7.	Summary and conclusion.....	132
	Declaration of competing interest.....	133
	Data availability.....	133
	Acknowledgments.....	133
	Appendix. Tuning parameter $A$ for ideal dynamic transformed inputs $v_A$ .....	133
	References.....	133

## 1. Introduction

Industry frequently makes use of nonlinear static model-based “calculation blocks”, “function blocks”, or “ratio elements” to provide feedforward action, decoupling or linearization (adaptive gain), and Shinskey [1] 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 study in this paper in the context of nonlinear input transformations.

Let  $u$  denote the original (physical) input and let  $v$  denote the transformed input (controller output) 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 Fig. 1. In this paper, we define the transformed input  $v$  as a nonlinear static function  $g$  of the physical input  $u$  and other variables

$$v = g(u, w, y, d) \quad (1)$$

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

- $v$  = transformed inputs (controller outputs)
- $u$  = original (physical) process inputs
- $d$  = measured disturbances
- $y$  = controlled process outputs (measured)
- $w$  = other measured dependent process variables (states).

In this paper, we do not include dynamic elements in the definition of the transformed input  $v$ , although dynamic elements are frequently used in industrial practice (e.g., [2]).

Nevertheless, even without dynamics, (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 [1] (on page 119) writes in relation to selecting input and output variables for the controller:

“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 (controller outputs) are

$$v = u + d \quad (2a)$$

$$v = \frac{u}{d} \quad (2b)$$

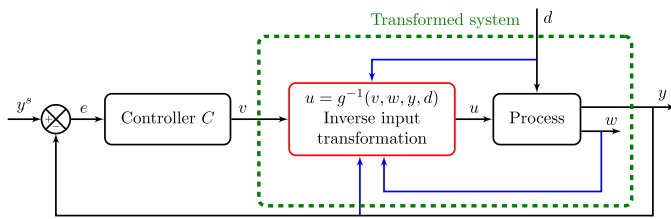
$$v = u_1 - u_2 \quad (2c)$$

$$v = \frac{u_1}{u_2} \quad (2d)$$

$$v = w \quad (2e)$$

Such transformed inputs are often introduced by engineers on simple physical grounds. The transformed input  $v = u + d$  in (2a) may provide feedforward action from a measured disturbance  $d$ . It may be 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 (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 (2c) and the ratio  $v = \frac{u_1}{u_2}$  in (2d).

The transformed input  $v = w$  in (2e) with  $w = F$  (measured flowrate) is probably the most common of all in process control. Its implementation results in a slave flow controller, where the physical input  $u$  is the valve position and the transformed input  $v$  is the flowrate  $F$  (or more precisely,  $v = F_s$ , the setpoint for the flowrate). This particular transformed input ( $v = F$ ) and the use of slave flow control is so common that in many cases



**Fig. 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 which inverts this relationship would then be  $u = g^{-1}(v, d) = vd$ . The outer feedback controller  $C$  has the aim of correcting for uncertainty, including for model error and unknown disturbances.

people consider the flowrate  $F$  to be the (physical) input  $u$ , and to simplify the expressions we will sometimes do this in this paper.

However, it is not enough to define the transformed input  $v$ , as in (1) and (2). For implementation we need to invert the transformation, that is, to generate from a given value of the controller output  $v = g(u, w, y, d)$ , the corresponding physical input  $u$ . Shinskey [1,3] calls this “reversing the process model”. There are two main ways of generating this inverse:

- A. *Model-based inverse*<sup>1</sup> using the “exact inverse input transformation”  $u = g^{-1}(v, w, y, d)$ ; see Fig. 1.
- B. *Feedback-based inverse* using a cascade implementation with a slave controller for  $v$  or  $w$ .

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

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

However, a model-based inverse is sometimes not possible, because  $g(u, w, y, d)$  may not depend explicitly on  $u$ . For example, for the transformed input  $v = F$  (flowrate), we have  $g(u, w, y, d) = w = F$  which is independent of the physical input  $u$ . In this case, the inverse must be generated by feedback using a slave flow controller.

In other cases, a model-based inverse may exist, but nevertheless we may use a feedback controller as a “trick” to generate an approximate inverse. In the control literature, this trick is often referred to as “dynamic inversion” [4].

In most cases, the selection of transformed inputs  $v$  is based on simple static models, for example, from material or energy balances [1,3]. However, the treatment of Shinskey is case-study based and in this paper, we aim to show how to derive the transformed variables in a systematic manner. Ideally, assuming no model error and that we measure all disturbances, these “ideal transformed variables” result in a transformed system from  $v$  to  $y$  (see Fig. 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 [5], which has a strong theoretical basis.

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 simplify the expression for the ideal transformed variable, by

<sup>1</sup> 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 and  $v$  the dependent variable, then the solution that results from solving  $v = g(u, d)$  with respect to  $u$  for a given  $v$  is written as  $u = g^{-1}(v, d)$ .

keeping 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 a soft sensor, estimator or observer (e.g., [6,7]) to estimate the non-measured variables  $d$  and  $w$ .

Surprisingly, there does not seem to be any academic literature on the common industrial approach of Shinskey [1,3] of using static models to derive transformed inputs. On the other hand, for dynamic models, 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., [5,6,8,9]). 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 rarely 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 and there are restrictions in terms of the class of systems and the assumption of full state feedback. Another reason is that, mainly for reasons of mathematical generality and simplicity, Isidori [5] 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 of 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 [5] introduces an outer state feedback controller as part of the solution. However, in many cases it is strongly desirable to be able to fix  $v$ , at least on an intermediate time scale, and actually the transformation into an integrating system is not necessary. For example, [6,7], 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 [10] emphasizes that the decision to make the transformed system integrating was just chosen as an example, but this message has not made its way to many of the potential users of feedback linearization. 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.

The paper starts in Section 2 with a motivating mixing example on the use of input transformations for decoupling. 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 show how to derive “ideal” model-based transformed inputs that 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 study the potential problem of internal instability. We also discuss the use of transformed outputs. In Section 7 we make our final conclusions and 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

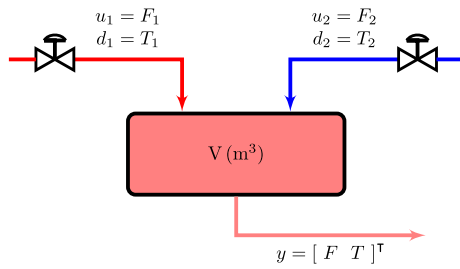


Fig. 2. Flowsheet of mixing process.

systematic methods for selecting “ideal” transformed input  $v$  that provide linearization, decoupling and disturbance rejection. However, in many cases, engineers use simpler transformed inputs [1] 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

The mixing process in Fig. 2 has two inlet streams, and initially we consider for simplicity the two flowrates  $F_1$  and  $F_2$  [kg/s] as the physical inputs, 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 Fig. 3(a). 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  $u_2 = F_2$  (cold water) by the same amount.

#### Input transformation for decoupling

To eliminate the interactions and make the process decoupled, we may use the alternative one-handle faucet in Fig. 3(b). 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 3 (Section 5.1).

For the modern one-handle design in Fig. 3(b), the transformed variables  $v_1$  and  $v_r$  are implemented physically (mechanically). However, to implement decoupling using the traditional two-handle design in Fig. 3(a), 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 normalized ratio  $v_2 = \frac{F_1}{F_1 + F_2}$ . The transformed inputs  $v = g(u)$  then become

$$v_1 = F_1 + F_2 = u_1 + u_2 \quad (4a)$$

$$v_2 = \frac{F_1}{F_1 + F_2} = \frac{u_1}{u_1 + u_2} \quad (4b)$$

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 normalized ratio  $v_2$  in (4b) has some properties that makes it better for implementation. First,  $v_2$  is always in the range 0 to 1, whereas  $v_r$  may vary between 0 and  $\infty$ . Second, as shown later in Example 3, the ratio  $v_2$  is a special case of the ideal static transformed input ( $v_0$ ) and provides linearization. Third, the expression for the exact inverse in the decoupling block becomes very simple with  $v_2$ , see (5). Fourth,  $v_2$  avoids division by zero when  $F_2 = 0$ , which may be an advantage if we obtain the inverse by feedback control (Alternative B in Fig. 6).

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

$$u_1 = F_1 = v_1 v_2 \quad (5a)$$

$$u_2 = F_2 = v_1 - v_1 v_2 \quad (5b)$$

The nonlinear decoupling  $g^{-1}$  in (5) may be implemented in the block “inverse input transformation” in Fig. 1 to provide a decoupled response from  $v$  to  $y$  for the traditional two-handle design.

Note that the inverse transformation contains no divisions, which is important to avoid division by zero. Furthermore, in this simple case, the equations can easily be implemented using standard multiplication and subtraction elements, for example, as shown in Fig. 4.

In Section 5.1 we will use a systematic procedure to derive a generalized (ideal) version of the transformed inputs in (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 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  $v$  in (4) 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. A general block diagram for this option is shown in Fig. 1, and the corresponding flowsheet for this particular example is shown in Fig. 5. A typical valve equation is

$$F = k_v f_v(z) \sqrt{\Delta P} \quad (6)$$

Here,  $F$  is the flow,  $z$  is the valve position,  $k_v$  is the valve constant and  $\Delta P$  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 Eq. (6) gives

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

For a linear valve, we get  $z = \frac{F}{k_v \sqrt{\Delta P}}$ . The resulting model-based inversion may be implemented as shown in Fig. 5, where the exact inverse block  $u = g^{-1}(v, d)$  computes the valve positions  $u = z$  (physical input) by combining the inversions in (5) and (7).



(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  $v_2 = \frac{F_1}{F_2}$ ).

Fig. 3. Valve (faucet) designs for mixing hot and cold water.

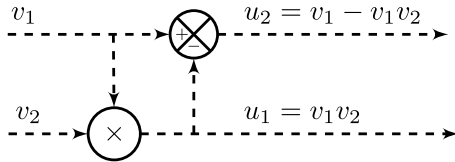


Fig. 4. Implementation of inverse transformation in (5) using standard multiplication and subtraction elements. The two transformed inputs are the sum ( $v_1$ ) and the normalized ratio ( $v_2$ ) of the two inputs ( $u_1$  and  $u_2$ ).

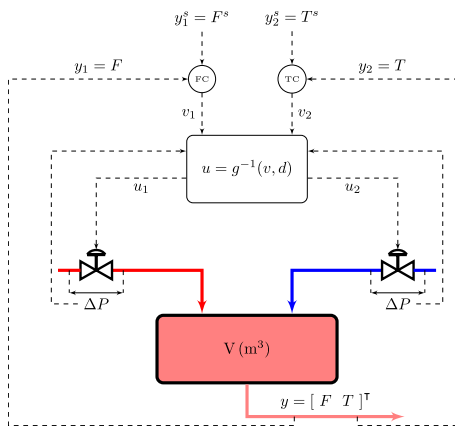


Fig. 5. Model-based inversion (Alternative A) for implementation of transformed inputs  $v_1 = F_1 + F_2$  and  $v_2 = \frac{F_1}{F_1+F_2}$  for the mixing process when the physical inputs  $u$  are the valve positions  $z$ . The inversion block  $u = g^{-1}(v, d)$  is a combination of the inversions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

Specifically, (5) is used to calculate the flowrates  $F_1$  and  $F_2$ , and (7) is used to calculate the corresponding valve positions  $z_1$  and  $z_2$ .

However, in practice, this model-based implementation may not work well, mainly because of uncertainty (error) in the valve characteristic  $f_v(z)$  and the valve constant  $k_v$ , but also because of incorrect measurements of the pressure drop disturbance ( $\Delta P$ ). Therefore, if the inlet flows ( $F_1, F_2$ ) can be measured with reasonable accuracy, it is better to use a feedback-based implementation where the feedback corrects for model uncertainty. Two such implementations are discussed next.

**Alternative B (purely feedback-based inversion).** The first option (Fig. 6) uses only feedback. The measured  $w$ -variables (flows  $F_1$  and  $F_2$ ) are used to compute the actual value of the transformed input  $v$  ( $v_1 = F_1 + F_2$  and  $v_2 = \frac{F_1}{F_1+F_2}$ ), and two slave  $v$ -controllers (VC1 and VC2) manipulate the physical inputs  $u$  (valve positions) to make  $v$  equal to the desired setpoints  $v^s$  which are computed by the two master controllers (FC and TC).

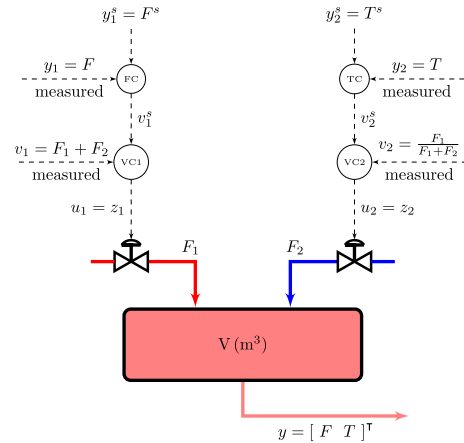


Fig. 6. Feedback inversion (Alternative B) for implementation of transformed inputs  $v_1 = F_1 + F_2$  and  $v_2 = \frac{F_1}{F_1+F_2}$  for the mixing process when the physical inputs  $u$  are the valve positions  $z$ . This solution makes use of slave  $v$ -controllers (VC1 and VC2).

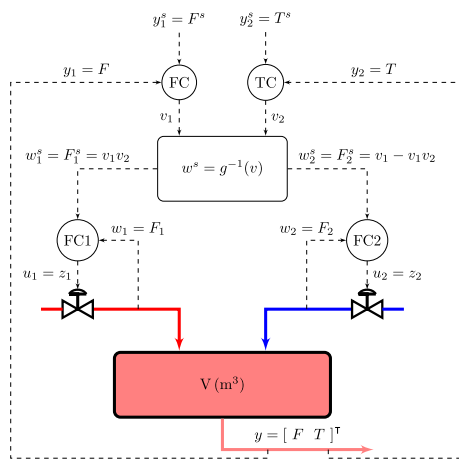
This alternative avoids the inverse block  $g^{-1}$ , so instead the decoupling from  $v$  to  $y$  is taken care of by the two slave  $v$ -controllers (VC1 and VC2). With integral action in the two  $v$ -controllers, decoupling will be exact at steady state.

However, decoupling will not be perfect dynamically, both because the slave  $v$ -controllers are not infinitely fast and because of interactions between the loops. To reduce interactions, we need to choose good variable pairings for VC1 and VC2. In Fig. 6, we have paired the total flow ( $v_1 = F_1 + F_2$ ) with  $z_1$  (which affects  $F_1$ ), and the ratio ( $v_2$ ) with  $z_2$  (which affects  $F_2$ ). This can be shown to be the preferred pairing if  $F_1$  the larger flow.

Since interactions result in dynamic couplings from  $v$  to  $y$  with the purely feedback-based implementation (Alternative B), one may question if there is any benefit compared to the simplest scheme, which is to have no input transformation (not shown in any figure), that is, letting the outer controller  $C$  (which are the two controllers TC and FC in this case) manipulate directly the flows ( $F_1$  and  $F_2$ ) using two slave flow controllers.

The answer is that there can indeed be a significant decoupling benefit when we use input transformations with feedback control of  $v$  (Alternative B) if there are effective delays associated with the control of  $y$  (e.g., measurement delays for  $F$  or  $T$  in our case), such that two slave  $v$ -controllers (VC1 and VC2) can be significantly faster than outer controllers in  $C$  (TC and FC).

**Alternative C (combined model- and feedback-based inversion).** The best option for this example is most likely to use the exact model-based inverse in (5) to compute the desired flowrates ( $F_1^s$  and  $F_2^s$ ) and combine this with two slave flow ( $w = F$ ) controllers (FC1 and FC2) that generate by feedback the corresponding valve positions ( $u_1 = z_1$  and  $u_2 = z_2$ ). This alternative



**Fig. 7.** Combined model and feedback inversion (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  $z$ . This solution combines the exact inverse  $g^{-1}(v)$  in (5) for computing the flow setpoints  $F_1^s$  and  $F_2^s$  with two slave flow( $w$ )-controllers (FC1 and FC2) that generate the corresponding physical inputs  $u_1 = z_1$  and  $u_2 = z_2$ .

is shown in the flowsheet in Fig. 7. It combines the decoupling of the model-based inversion in (5) with the linearization and disturbance rejection of the flow controllers. The two slave flow controllers, through the action of fast feedback, indirectly generate the inverse in (7), and makes the process (almost) linear and independent of disturbances in  $\Delta p_1$  and  $\Delta p_2$ , at least as seen from slower time scale of the outer controllers (TC and FC).

The next section generalizes the three alternative ways (A, B and C) of generating the inverse.

### 3. Implementation of transformed inputs

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 sets the desired value (setpoint)  $v^s$  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 Figs. 1 and 8(a). With an exact inverse  $g^{-1}$ , this gives exactly  $v = v^s$  (assuming all variables are measured perfectly and there is no model error).
- B. Feedback-based implementation, see Fig. 8(b). With integral action in the slave controller  $C_v$  this gives  $v = v^s$  after a dynamic transient.

We also discuss a third implementation (Fig. 8(c)) which is a combination of the two. The three alternatives are the same as the ones presented in Figs. 5–7 for Example 2, respectively.

#### 3.1. Alternative A: Model-based inversion (Figs. 1 and 8(a))

The first approach is to invert the input transformation  $v = g(u, w, y, d)$  in (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) (Fig. 8(b))

A dynamic approximation of the inverse input transformation may be generated using an inner (slave) feedback controller  $C_v$  as shown in Fig. 8(b). 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  $v^s$ . The inner controller  $C_v$  may be tuned based on an experimental response from  $u$  to  $v$ . Usually, a PI-controller, or even a pure I-controller, is sufficient. 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, for example, to reduce dynamic interactions.

#### 3.3. Alternative C: Combined feedback- and model-based inversion with slave $w$ -controller (Fig. 8(c))

This combined implementation with an inner  $w$ -controller is of particular interest for the case when  $v = g(w, y, d)$  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  $w^s = g^{-1}(v, y, d)$ .

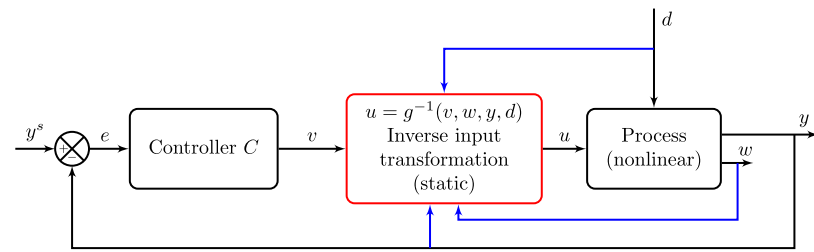
Feedback implementation C (Fig. 8(c)) has the advantage that we can include some model-based inversion, which may contribute to linearization, feedforward and decoupling. It also has the advantage that control of  $w$  (Alt. C) is usually less interactive than control of  $v$  (Alt. B), which is a significant advantage for faster convergence with single-loop control. A feedback implementation (alternatives B or C) is required if  $v$  does not explicitly depend on  $u$ . In other cases, a feedback implementation may be used as a “trick” for numerically generating an approximate inverse. In the control literature, this trick is often referred to as “dynamic inversion” [4]. The reason for using this trick could be to avoid the complexity of deriving the inverse in (9) (see heat exchanger example) or to avoid problems with singularities [4].

### 4. Derivation of ideal transformed inputs

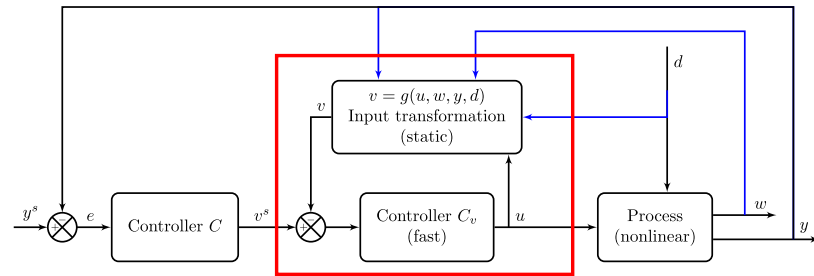
Input transformations are in common use and as illustrated in the motivating mixing example they may be very useful. However, the main question we want to answer in this paper is:

*How do we derive good transformed inputs in a systematic manner?*

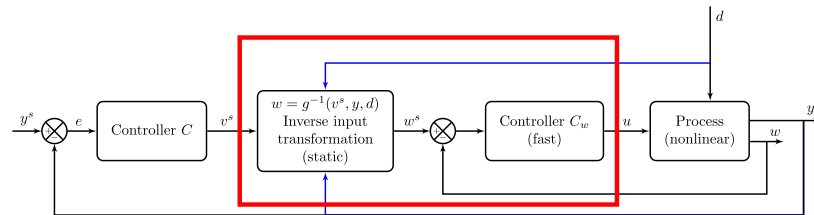
Starting from a static or dynamic process model, we show in this section how to derive ideal transformed inputs (denoted  $v_0$  and  $v_A$ , respectively) which ideally achieve linearization, decoupling and disturbance rejection, see Fig. 9. We assume that we have an  $n \times n$  control problem with  $n$  inputs  $u$  and  $n$  outputs  $y$ ,



(a) Alternative A. Model-based implementation 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) Alternative B. Feedback implementation of transformed input  $v = g(u, w, y, d)$  using cascade control with a slave  $v$ -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.



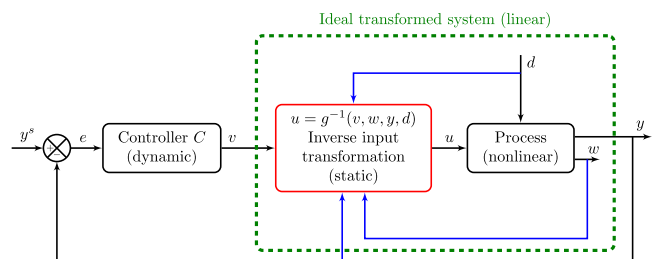
(c) Alternative C. Combined model-based and feedback implementation of transformed input  $v = g(u, w, y, d)$  using slave  $w$ -controller. Commonly,  $C_w$  is a flow controller ( $w = \text{flowrate}$ ) and  $u$  is the valve position. This implementation generates an approximate inverse.

**Fig. 8.** Alternative implementations (red blocks) for inverting the input transformation  $v = g(u, w, y, d)$ , that is, of generating the physical input ( $u$ ) from a given value of the transformed input ( $v$  or  $v^s$ ). Controllers  $C$ ,  $C_v$  and  $C_w$  are usually single-loop PID controllers.

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 the mixing example in Section 5.1. In Section 4.3, we discuss that it is simpler in many cases to write the model in terms of extra measured state variables  $w$ , because we can then replace model equations by measured variables.

#### 4.1. Obtaining an ideal transformed system from a static process model

In the industrial literature, Shinskey [1] 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, he provides no systematic approach, 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. We will first consider the case with no measured states  $w$ , and we will then generalize to include  $w$ -variables.



**Fig. 9.** With ideal transformed inputs the transformed system (with no uncertainty) is linear and independent of disturbances. For a static process model, the transformed input  $v = v_0 = g(u, w, d)$  is given in (13) or (27). For a dynamic process model, the transformed input  $v = v_A = g(u, w, y, d)$  is given in (19) or (30). In most cases, the tuning parameters ( $B_0, A, B$ ) are chosen such that the ideal transformed system is decoupled and gives  $y = v$  at steady state. The inversion  $g^{-1}$  may alternatively be generated by feedback, see Fig. 8(b), but in this case the ideal properties of the transformed system are achieved only after a dynamic transient.

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

$$0 = f(u, y, d) \tag{10}$$

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

$$0 = f_x(u, x, y, d) \quad (11)$$

where  $x$  represents additional internal variables (states). In the more general case in (11), 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 (10).

Since the model equations in (10) 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 input based on static model.** Consider a static nonlinear model in the form

$$y = f_0(u, d) \quad (12)$$

Define from this the ideal static transformed input

$$v_0 = \underbrace{B_0^{-1} f_0(u, d)}_{g(u, d)} \quad (13)$$

where the matrix  $B_0$  is free to choose. Assume that  $v_0$  can be exactly implemented by solving (13) with respect to  $u$  to get the ideal input

$$u = g^{-1}(v_0, d) \quad (14)$$

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

$$y = B_0 v_0 \quad (15)$$

The transformed system in (15) 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 (13) we get  $f_0(u, d) = B_0 v_0$  and substituting this into (12) gives  $y = B_0 v_0$  in (15). The assumptions related to (14) are necessary to be able to generate the corresponding ideal input  $u$ .  $\square$

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

Importantly, it may not be necessary to explicitly derive an expression for  $y = f_0(u, d)$  in (12). 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 (15), it is in some cases simpler to stay with the original model equations in (10) or (11), and rather than solving them with respect to  $y$  (for a given  $u$ ), solve them with respect to  $u$  for a given  $y = B_0 v_0$  to obtain the inverse transformation  $u = g^{-1}(v_0, d)$ . The solution to find the inverse  $g^{-1}$  can be either analytical or numerical. A numerical solution may be necessary for complicated models, like for the heat exchanger example discussed later and in [11].

#### 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

$$B_0 = I \quad (16)$$

such that the ideal transformed input is simply the right-hand side of the model equation,

$$v_0 = f_0(u, d) \quad (17)$$

In this case the transformed system simply becomes  $y = v_0$ , and it may be tempting to think of the transformed input  $v_0$  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 Fig. 1). Thus, it is better to think of  $v_0$  as the transformed process input.

#### 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 (18). The derivation of the resulting ideal transformed input  $v_A$  is closely related to the theory of feedback linearization for the special case of a low-order model (with a relative order of 1).

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

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

For the model in (18), the ideal transformed input is

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

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

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

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

$$\frac{dy}{dt} = Ay + Bv_A \quad (21)$$

The transformed system in (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 (19) into (18) gives (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 notation with a 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)  $n$  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  $\frac{dy}{dt}$  of the controlled output  $y$ , which means that the relative order from  $u$  to  $y$  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 (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 below.)

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 (18) as shown in [8] (p. 293).

$$f(u, y, d) = f_1(y, d) + f_2(y, d) u \quad (22)$$

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 (22). Nevertheless, we do not make this assumption in this paper, so the invertibility may need to be studied separately for each application.



#### 4.2.1. Choice of tuning parameter B

To get dynamic decoupling in (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.

One simple choice is  $B = I$ , which is often used in feedback linearization. Alternatively, 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  $\frac{dy}{dt} = f(u, y, d) = 0$ ).<sup>2</sup> With the choice  $B = -A$ , the transformed input and corresponding transformed system become

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

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

#### 4.2.2. Choice of tuning parameter A

The choice of the parameter A 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

$$A = \tilde{A} \equiv \left( \frac{\partial f}{\partial y} \right)_* \tag{25}$$

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  $y$ .

In the multivariable case, to obtain decoupling, we may choose  $A = \text{diag}(\tilde{A})$ .

**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 (25). 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 (25) 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 [12].

**3. Make the system integrating:**  $A = 0$ . The choice  $A = 0$  is recommended in the standard feedback linearization literature [5]. This results in an integrating transformed system,  $\frac{dy}{dt} = Bv_A$ , where usually one selects  $B = I$ , corresponding to selecting the transformed input as the right-hand side of the differential equation model,  $v_A = f(u, y, d)$ . However, although being very

<sup>2</sup> 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 thus B) is not diagonal. However, to also get dynamic decoupling, we must choose A to be diagonal.

simple, we do not recommend the choice  $A = 0$  for most process control applications. The main reason is that with  $A = 0$ , a system which was originally stable is turned into an integrating transformed system. Thus, 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 undesirable, because usually one wants to be able to operate the transformed system without the outer controller C. Another reason for not selecting  $A = 0$  is that we generally want to use integral action in the outer controller C to correct for uncertainty. With  $A = 0$ , the integrator in the transformed system poses additional performance limitations for disturbances at the plant input (e.g., [12]). 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. Transformed input in terms of measured state variables w

The expressions for the ideal transformed inputs ( $v_0$  and  $v_A$ ) are the same also when we include additional measured dependent variables  $w$  (states) in the model equations (in  $f_0$  and  $f$ ). This allows us to use simpler models, because a measured  $w$  replaces a model equation for  $w$ . For example, a measured flow  $w = F$  may replace a valve Eq. (6).

Following the same derivations as above, we find that the expressions  $v = g(u, w, y, d)$  for the ideal transformed inputs remain the same, except that we add  $w$  in the argument list. Then, assuming that we measure  $d$  and  $w$  perfectly, have a perfect model, are able to generate the exact inverse  $g^{-1}$  and that the resulting transformed system is internally stable, the resulting transformed system becomes as before linear and independent of disturbances.

**Ideal transformed input with w-variables for static case.** Specifically, for the static model

$$y = f_{0,w}(u, w, d) \tag{26}$$

the ideal transformed input becomes

$$v_0 = \underbrace{B_0^{-1}f_{0,w}(u, w, d)}_{g(u,w,d)} \tag{27}$$

resulting in the transformed system

$$y = B_0 v_0 \tag{28}$$

**Ideal transformed input with w-variables for dynamic case.** Similarly, for the dynamic model

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

the ideal transformed input becomes

$$v_A = \underbrace{B^{-1}(f_w(u, w, y, d) - Ay)}_{g(u,w,y,d)} \tag{30}$$

resulting in the transformed system

$$\frac{dy}{dt} = Ay + Bv_A \tag{31}$$

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.

#### 4.3.1. Dynamics of transformed system with measured state ( $w$ ) variables

If the  $w$ -variables depend dynamically on the inputs  $u$ , then the implicit feedback through the measurements  $w$  will affect the dynamics of the transformed system. In some cases, this may result in internal instability as discussed below.

For the static case, the dynamics of the transformed system (from  $v_0$  to  $y$ ) will no longer be the same as that of the original system (from  $u$  to  $y$ ).

The reason is that the feedback from  $w$  through  $v_0$  may either speed up or slow down the response. Note that for the static case, we have no tuning parameter to change the dynamics of the transformed system.

#### 4.3.2. Possible unstable zero dynamics and internal instability

Note that we are essentially treating the variable  $w$  as a measured disturbance when deriving the transformed variables  $v_0$  and  $v_A$  in (27) and (30). Is there any problem in doing this? Yes, there will be a problem with the exact inverse, if the  $w$ -variables result in unstable zero dynamics<sup>3</sup> in the dynamic map from the physical input  $u$  to the transformed input  $v$  (which may be  $v_0$  or  $v_A$  or any other transformed input). The reason is that unstable zero dynamics will result in internal instability for the transformed system if we implement the exact inverse  $u = g^{-1}(v, w, y, d)$ . 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 (Alternative A) in Fig. 8(a) may yield internal instability in some cases. Fortunately, as argued in the discussion section, it is not very likely to happen in practice, because unstable zero dynamics require that the indirect dependency of  $u$  on  $v$  through  $w$  is strong.

The internal instability can in any case be avoided if we use the alternative implementation B in Fig. 8(b) with a slave  $v$ -controller  $C_v$ , but  $C_v$  then needs to be tuned sufficiently slow so that the unfavorable zero dynamics do not cause closed-loop instability. Thus, linearization, decoupling and disturbance rejection will not be perfect dynamically in this case.

Note that these problems with unstable zero dynamics are fundamental and do not arise because we use transformed inputs. In particular, for the ideal transformed inputs ( $v_0$  and  $v_A$ ), the zero dynamics from  $u$  to  $v$  are the same as the zero dynamics from  $u$  to  $y$ . This follows because of the direct relationship between  $v$  and  $y$  in this case, for example,  $y = v_0$  for the static case. Our problem with internal stability with the exact inverse, is thus a special case of the well-known fact that perfect control of the output  $y$  cannot be achieved for a system with unstable zero dynamics (RHP-zeros in the linear case) from  $u$  to  $y$ , no matter how good the model is or what we measure (e.g., [13]).

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

If we apply  $v_0$  in (27) (derived from a static model) as a transformed input to a dynamic system (but with the same static model), then with the choice  $B_0 = I$ , the transformed system becomes  $y = v_0$  and we clearly 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 (29), then we get perfect disturbance rejection also

<sup>3</sup> 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 domain. More generally, for nonlinear systems the unstable zero dynamics from  $u$  to  $v$  correspond to the unstable dynamics of the inverse map from  $v$  to  $u$  [5].

dynamically, if we initially are at steady state. This surprising fact, which is observed in several of the examples and proved in the discussion in Section 6.3, holds because of the particular simple dynamics assumed in (29). The assumption about being initially at steady state is not limiting, because it is desirable that the system stays close its steady state setpoint.

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 very tempting for an engineer to apply  $v_0$  also to dynamic systems.

However, there are some advantages of instead applying the dynamic transformed input  $v_A$  (30) to the dynamic system  $\frac{dy}{dt} = f(u, w, y, d)$ . The main advantage is that 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. Second, for the case when the ideal static transformed input  $v_0$  depends on measured states  $w$ , 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. Ideal transformed inputs for mixing process (Motivating Example 1, continued)

Consider the mixing process in Fig. 2 with the following inputs, outputs and disturbances

$$u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}; \quad y = \begin{bmatrix} F \\ T \end{bmatrix}; \quad d = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \quad (32)$$

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

For simplicity, we assume that the mass  $m$  [kg] of the system is constant, which is a good 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

$$F = F_1 + F_2 \quad (33)$$

Assuming perfect mixing, the dynamic energy balance becomes

$$m \frac{dT}{dt} = F_1 T_1 + F_2 T_2 - FT \quad (34)$$

We have here assumed constant and equal heat capacities so that  $c_p$  drops out of the energy balance. Substituting the mass balance (33) into the energy balance (34) and using the more general notation in (32), then gives the following model equations for the mixing process

$$y_1 = \underbrace{u_1 + u_2}_{f_{0,1}(u,d)} \quad (35a)$$

$$\frac{dy_2}{dt} = \frac{1}{m} \underbrace{(u_1(d_1 - y_2) + u_2(d_2 - y_2))}_{f_2(u,y,d)} \quad (35b)$$

5.1.1. Ideal transformed input  $v_0$  from static model

We first derive the ideal transformed inputs for a purely static model. By setting  $m \frac{dy_2}{dt} = 0$  we derive from (35b) the following static equation for the temperature  $y_2 = T$

$$y_2 = \frac{u_1 d_1 + u_2 d_2}{\underbrace{u_1 + u_2}_{f_{0,2}(u,d)}} \quad (36)$$

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

$$v_{0,1} = \underbrace{u_1 + u_2}_{g_1(u)=f_{0,1}(u,d)} \quad (37a)$$

$$v_{0,2} = \frac{u_1 d_1 + u_2 d_2}{\underbrace{u_1 + u_2}_{g_2(u,d)=f_{0,2}(u,d)}} \quad (37b)$$

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 static 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 (37) for  $v_0 = g(u, d)$  to find the physical inputs (flows)  $u$ . We get  $u = g^{-1}(v_0, d)$  where

$$u_1 = \frac{v_{0,1}(v_{0,2} - d_2)}{d_1 - d_2} \quad (38a)$$

$$u_2 = \frac{v_{0,1}(d_1 - v_{0,2})}{d_1 - d_2} \quad (38b)$$

There is a singularity in the inverse 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.

5.1.2. Comparison with engineering-based variables from Motivating Example 1

Comparing the ideal static transformed inputs in (37) with the engineering-based variables in (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  $d_2 = T_2$ .

5.1.3. 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 (39)? Substituting  $u_1$  and  $u_2$  from (38a) and (38b) into (35b) gives after a little algebra the following transformed dynamic system

$$y_1 = v_{0,1} \quad (39a)$$

$$\frac{dy_2}{dt} = \frac{v_{0,1}}{m}(v_{0,2} - y_2) \quad (39b)$$

We see that both disturbances  $d_1$  and  $d_2$  drop out, so the transformed system in (39) based on  $v_{0,2}$  is independent of disturbances, also dynamically.<sup>4</sup> We note that the transformed system

<sup>4</sup> 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.

in (39) is not truly decoupled, because we see from (39b) 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 (39b) tells that a change in  $v_{0,1}$  will not affect  $y_2$ .

5.1.4. Ideal transformed input  $v_A$  from dynamic model

Actually, since the static transformed input  $v_0$  gives both disturbance rejection and decoupling also for the dynamic process, there is little reason to consider the dynamic transformed input  $v_A$ , but we will do it for completeness and comparison.

The first model equation is static, so the first transformed input is unchanged

$$v_{0,1} = u_1 + u_2 \quad (40)$$

To derive the second ideal transformed input, we use the right-hand side  $f_2(u, y, d)$  of the dynamic energy balance in (35b), and from (24a) we derive

$$v_{A,2} = y_2 - A^{-1}f_2(u, y, d) \quad (41a)$$

$$= y_2 - A^{-1} \underbrace{\frac{1}{m}(u_1(d_1 - y_2) + u_2(d_2 - y_2))}_{g_2(u,y,d)} \quad (41b)$$

Note that we have chosen  $B = -A$  which gives  $y_2 = v_{A,2}$  at steady state. From (40) and (41b) the inverse transformation becomes

$$u_1 = \frac{v_{0,1}(y_2 - d_2) - Am(v_{A,2} - y_2)}{d_1 - d_2} \quad (42a)$$

$$u_2 = \frac{v_{0,1}(d_1 - y_2) + Am(v_{A,2} - y_2)}{d_1 - d_2} \quad (42b)$$

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

$$y_1 = v_{0,1} \quad (43a)$$

$$\frac{dy_2}{dt} = A(y_2 - v_{A,2}) \quad (43b)$$

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 (41b) at the nominal operating point, we choose  $A$  such that we keep the nominal linearized dynamics of the original system, which from (25) gives

$$A = \left( \frac{\partial f_2}{\partial y_2} \right)_* = -\frac{F^*}{m} = -\frac{v_{0,1}^*}{m} \quad (44)$$

where  $F^* = u_1^* + u_2^* = v_{0,1}^*$  is the nominal total flowrate. Note that the expression for the transformed system in (39b) in terms of  $v_{0,2}$  (static model) is very similar to (43b) in terms of  $v_{A,2}$  (dynamic model) if we choose  $A$  as given in (44). The main difference is that the transformed dynamic system (43b) for  $v_{A,2}$  is linear, whereas the transformed dynamic system (39b) for  $v_{0,2}$  is nonlinear because of the multiplication with the term  $v_{0,1}$ .

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 linear dynamically. This benefit is small compared to the added complexity, for example, of having to select the tuning parameter  $A$ . Thus, most likely an engineer will prefer to use the static variables  $v_0$ . If the disturbances  $d_1 = T_1$  and  $d_2 = T_2$  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.

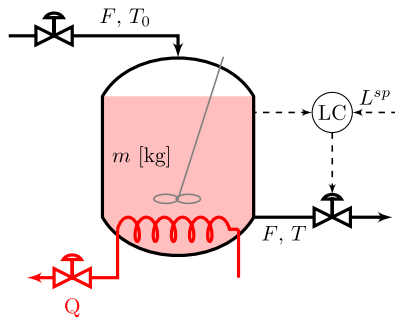


Fig. 10. Process flowsheet of tank heated with an electric coil.

### 5.2. Example 4. Heated tank

Consider the continuous process in Fig. 10 with an electric heater. Assuming perfect mixing, constant heat capacity  $c_p$  [kJ/°C] and constant mass holdup  $m$  [kg], the energy balance gives the following dynamic model

$$\frac{dT}{dt} = f(u, y, d) = \frac{1}{mc_p}(Fc_p(T_0 - T) + Q) \quad (45)$$

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  $\frac{dT}{dt} = 0$ , we derive the corresponding static model for the outlet temperature

$$T = f_0(u, d) = T_0 + \frac{Q}{Fc_p} \quad (46)$$

From (27) and (30), the ideal transformed inputs  $v_0$  and  $v_A$ , based on a static and dynamic model, respectively, become

$$v_0 = f_0(u, d) = T_0 + \frac{Q}{Fc_p} \quad (47a)$$

$$v_A = -A^{-1}f(u, y, d) + y = -A^{-1}\left(\frac{F}{m}(T_0 - T) + \frac{Q}{mc_p}\right) + T \quad (47b)$$

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

If we apply these two transformed variables to the dynamic system in (45) then the transformed dynamic system becomes for the two cases

$$\frac{dT}{dt} = \frac{F}{m}(v_0 - T) \quad (48a)$$

$$\frac{dT}{dt} = -A(v_A - T) \quad (48b)$$

For both cases, we find that the transformed system is independent of disturbances (in  $Q$  and  $T_0$ ). If we choose  $A = -\left(\frac{\partial f}{\partial T}\right)_* = -\frac{F}{m}$  as recommended in (25), 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 ((47a)) in terms of  $v_0$  is nonlinear, whereas the transformed system ((47b)) in terms of  $v_A$  is linear. However, the nonlinearity in (48a) is small unless  $F$  varies a lot, so most likely an engineer will prefer to use the simpler static variable  $v_0$ .

For implementation using the exact inverse in Fig. 8(a), we need to invert the expression for the transformed input  $v$  to find the physical input  $u = F$ . For  $v = v_0$ , based on a static model, (47a) gives the inverse transformation

$$u = F = \frac{Q}{c_p(v_0 - T_0)} \quad (49)$$

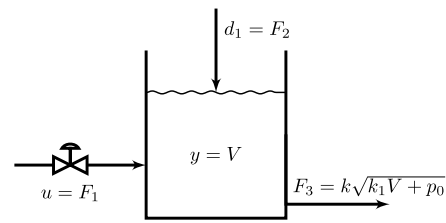


Fig. 11. Flowsheet of the nonlinear level process.

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 (not included), and found that it made the system drift away from the desired steady state, and it did not recover. Fortunately, there are ways of handling the singularity. One is to use logic and set  $u = u_{max}$  when the computed value of  $u$  is negative. Another way is to use the cascade implementation in Fig. 8(b) with a slave  $v$ -controller.

A similar singularity occurs with the ideal transformed input  $v_A$ . Inverting (47b) gives

$$u = F = \frac{Q + Amc_p(v_A - y)}{c_p(y - T_0)} \quad (50)$$

The singularity at  $y = T_0$  may happen in situations with large dynamic variations in  $d = T_0$  or  $y = T$ , but can be handled in a similar way as mentioned for  $v_0$ .

### 5.3. Example 5: Simple nonlinear level process

We consider the control of level (volume) in a tank with two inflows and one outflow, as shown in Fig. 11. We assume that the level measurement is delayed or infrequent, so we would like to make use of feedforward action from measured disturbances.

Assuming constant liquid density, we derive from the mass balance the following dynamic model

$$\frac{dV}{dt} = F_1 + F_2 - F_3 \quad (51)$$

The variables are defined as:

- $y = V =$  volume of fluid (measured)
- $u = F_1 =$  inflow (manipulated)
- $d_1 = F_2 =$  inflow (measured disturbance)
- $F_3 =$  outflow (not measured)
- $d_2 = p_0 =$  pressure difference between before and after tank (measured).

We assume that the inflow  $u = F_1$  can be manipulated directly, for example, because we have a fast slave flow controller. This is not an integrating process, because the outflow  $F_3$  depends on the level ( $V$ ) and provides self-regulation. The outflow is not measured, but from the Bernoulli equation we derive the following model

$$F_3 = C_v\sqrt{\rho gV/A + p_0} = \sqrt{c_1V + c_2p_0} \quad (52)$$

where  $c_1 = C_v^2\rho g/A$  and  $c_2 = C_v^2$  are constants. The dynamic model then becomes

$$\frac{dy}{dt} = f(u, y, d) = u + d_1 - \sqrt{c_1y + c_2d_2} \quad (53)$$

Transformed variable  $v_0$  based on static model. Solving  $0 = f(u, y, d)$  in (53) with respect to  $y$  gives the static model

$$y = f_0(u, d) = \frac{(u + d_1)^2 - c_2 d_2}{c_1} \quad (54)$$

The ideal transformed variable (selecting  $B_0 = 1$ ) is the right-hand side of the static model

$$v_0 = f_0(u, d) = \frac{(u + d_1)^2 - c_2 d_2}{c_1} \quad (55)$$

For a given  $v_0$ , the corresponding input is given by the inverse transformation

$$u = f_0^{-1}(v_0, d) = \sqrt{c_1 v_0 + c_2 d_2} - d_1 \quad (56)$$

At steady state the transformed system is  $y = v_0$ , so it is linear and independent of the disturbances  $d_1$  and  $d_2$ . What happens if we apply the ideal static transformed input  $v_0$  to the dynamic system? Substituting (56) into (53) gives the transformed dynamic system

$$\frac{dy}{dt} = \sqrt{c_1 v_0 + c_2 d_2} - \sqrt{c_1 y + c_2 d_2} \quad (57)$$

Thus, the transformed dynamic system 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 (57) we have  $\frac{dy}{dt} = 0$  for any disturbance  $d_2$ . Thus,  $y$  will remain at  $v_0$  and we have perfect disturbance rejection for  $d_2$  also dynamically.

To see this even more clearly, we linearize the transformed system model (57) around the steady-state (where  $y^* = v_0^*$ ) to get

$$\frac{d\Delta y}{dt} = \frac{c_1}{2\sqrt{c_1 y^* + c_2 d_2^*}} (\Delta v_0 - \Delta y) \quad (58)$$

where the  $\Delta$ -variables represent deviations from the steady state. Note that  $d_2$  drops out in (58). Also note that  $F_3^* = \sqrt{c_1 y^* + c_2 d_2^*}$  so the initial dynamic response will vary as a function of  $F_3^*$  and thus depend on the operating point.

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

$$v_A = y - A^{-1}f(u, y, d) = \underbrace{y - A^{-1}(u + d_1 - \sqrt{c_1 y + c_2 d_2})}_{g(u, y, d)} \quad (59)$$

and the resulting transformed dynamic system becomes as expected

$$\frac{dy}{dt} = A(y - v_A) \quad (60)$$

The exact inverse of (59) gives the corresponding input

$$u = g^{-1}(v_A, y, d) = A(y - v_A) - d_1 + \sqrt{c_1 y + c_2 d_2} \quad (61)$$

To eliminate the feedback from the  $y$  to  $v_A$  at the nominal point, we may from (25) choose

$$A = \left(\frac{\partial f}{\partial y}\right)_* = -\frac{c_1}{2\sqrt{c_1 y^* + c_2 d_2^*}} \quad (62)$$

As expected, the linearized transformed system (58) for  $v_0$  is identical to  $\frac{dy}{dt} = A(y - v_A)$  in (60) if we select  $A$  as in (62). However, there is an important difference. With  $v_A$ , the transformed system is linear dynamically because  $A$  is a constant, whereas with  $v_0$  the dynamic response is nonlinear as the initial

gain is inversely proportional to the value of the outflow  $F_3$ . However, the transformed system with  $v_0$  is linear at steady state, so if the outer loop is slow, the dynamic nonlinearity will not matter. Thus, again an engineer probably will most likely prefer the static transformed input  $v_0$ , which also does not depend on the delayed or infrequent measurement of the volume  $y$ .

Transformed variable  $v_A$  based on dynamic model for the case with measurement of  $w = F_2$ . From the examples so far, it seems that there is not much benefit of using the ideal dynamic transformed input  $v_A$  instead of the simpler static transformed input  $v_0$ . However, we here consider a case where this is not true, simply because the system is integrating and thus has no steady state.

Consider again control of level (volume  $V$ ) in Fig. 11, with the same dynamic model (51), but assume here that the outflow  $F_2$  is measured, that is,

$$w = F_2.$$

If this measurement is fast and accurate, then it is better to use it rather than the model equation  $F_3 = k\sqrt{k_1 V + p_0}$  which is inaccurate and also depends on measuring the disturbance  $d_2 = p_0$ . Introducing  $w = F_2$ , the dynamic model  $\frac{dV}{dt} = F_1 + F_2 - F_3$  becomes

$$\frac{dy}{dt} = f_w(u, w, d_1) = u + d_1 - w \quad (63)$$

Note that the right-hand side ( $f_w$ ) is independent of  $y$  so this model is integrating with no steady-state value for  $y$ . Therefore, we cannot derive a corresponding transformed input  $v_0$ . Instead, we derive the ideal dynamic transformed input  $v_A$ . Selecting  $B = 1$  and  $A = 0$  in (30) gives the transformed input

$$v_A = f_w(u, w, d_1) = u + d_1 - w \quad (64)$$

which is simply the net inflow to the process. The transformed system becomes  $\frac{dy}{dt} = v_A$ , which is linear, integrating and independent of disturbances. Given  $v_A$ , we get from (64) that the corresponding input  $u = F_1$  is

$$u = v_A - d_1 + w \quad (65)$$

Of course, we need an outer controller  $C$  to adjust  $v_A$  and stabilize the integrating transformed system, but otherwise the implementation is very simple.

Let us finally remark that it is possible to select  $v_A = f_w(u, w, d_1) - Ay = u + d_1 - w - Ay$  where  $A < 0$ . In this case, the transformed system  $\frac{dy}{dt} = v_A + Ay$  is stable, although the original system in (63) is integrating. However, selecting  $A \neq 0$  introduces feedback from  $y$  in the input transformation, which will come in addition to the feedback from  $y$  in the outer controller  $C$ , which anyway could be used to stabilize the system. Thus, in this case there is little benefit of selecting a more complicated  $v_A$  with  $A \neq 0$ .

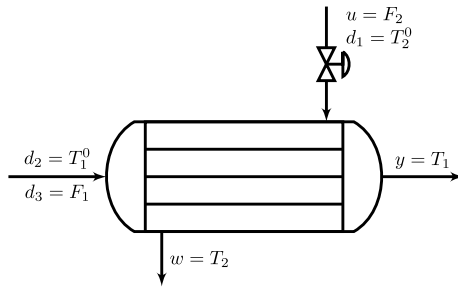
#### 5.4. Example 6: Heat exchanger

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 Fig. 12 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,  $u = F_2$ . In summary, we have for this example

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

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

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



**Fig. 12.** Heat exchanger (Example 6) where the objective is to control the outlet temperature  $T_1$  of stream 1 (process side) by exchanging heat with stream 2 (utility side).

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

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

A possible extra measurement (in addition to  $F_2$ ) which 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 is because we 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 modeled by partial differential equations, but we use a cell model with  $n = 100$  well-mixed cells on each side; see [14] 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  $\frac{dy}{dt} = f(u, y, d)$  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, which we will then apply to the dynamic heat exchanger cell model. Thus, in this example, we analyze by simulation the effect of model uncertainty. We will consider two transformed inputs

$$v_0 = f_0(u, d) \quad (66)$$

$$v_{0,w} = f_{0,w}(u, w, d) \quad (67)$$

The first is the ideal transformed input  $v_0$  that follows from the detailed static model  $y = f_0(u, d)$ . Note that this model does not depend on the measured state variable  $w = T_2$  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, without having to include a model for the heat transfer. 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 have 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 a model equation where  $UA$  is included.

#### 5.4.1. 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:

$$Q = F_1 c_{p1} (T_1^0 - T_1) \quad (68a)$$

$$Q = F_2 c_{p2} (T_2 - T_2^0) \quad (68b)$$

$$Q = UA \frac{(T_1^0 - T_2) - (T_1 - T_2^0)}{\ln \left( \frac{T_1^0 - T_2}{T_1 - T_2^0} \right)} \quad (68c)$$

This gives 3 equations in 3 unknowns ( $Q, T_1, T_2$ ) which can be solved analytically to find the following analytic expression for  $T_1$  as a function of the input and the disturbances (e.g., [15])

$$y = T_1 = \underbrace{T_1^0 + \epsilon(T_2^0 - T_1^0)}_{f_0(u, d)} \quad (69)$$

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 (69) the corresponding ideal static transformed input becomes

$$v_0 = f_0(u, d) = T_1^0 + \epsilon(T_2^0 - T_1^0) \quad (70)$$

where it should be noted that  $\epsilon$  is a nonlinear function of  $F_1, F_2$  and  $UA$ .

#### 5.4.2. Transformed input $v_{0,w}$ based on parts of static model and measured state $w = T_2$

The second transformed variable,  $v_{0,w}$ , follows by using the measured state  $w = T_2$  to replace the heat transfer Eq. (68c) for  $Q$ . We use (68a) to find

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

and then we substitute  $Q$  using (68b) to get

$$y = T_1 = T_1^0 + \underbrace{\frac{F_2 c_{p2}}{F_1 c_{p1}} (T_2^0 - T_2)}_{f_{0,w}(u, w, d)} \quad (71)$$

From (71) the corresponding ideal static transformed input becomes

$$v_{0,w} = f_{0,w}(u, w, d) = T_1^0 + \frac{F_2 c_{p2}}{F_1 c_{p1}} (T_2^0 - T_2) \quad (72)$$

which depends on  $w = T_2$  but not on the  $UA$ -value.

#### 5.4.3. Implementation

For both transformed inputs, we will use the pure feedback-based implementation in Fig. 8(b) with a slave  $v$ -controller ( $C_v$ ). An alternative would be to use an exact model-based inverse (analytical or numerical) to compute  $u = F_2$  plus use a slave flow controller to implement  $F_2$  (Fig. 8(c)). The use of the slave  $v$ -controller avoids the flow controller, and has an additional advantage in both cases. 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  $C_v$ . For  $v_{0,w}$  there is no problem

in generating the exact inverse from (72) to compute  $u = F_2$ , but with a  $v$ -controller we avoid worrying about the potential internal instability or excessive inputs of using the exact inverse (although this is not a problem in this case).

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 ( $y$ ) or state ( $w$ ), 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” (calculation block) from  $u = F_2$  to  $v_0$  in (70) is static, and a pure I-controller is recommended [12]. The tuning of the slave  $v_{0,w}$ -controller is more difficult, because  $w = T_2$  depends dynamically on  $u = F_2$  which results in an overshoot in the dynamic response from  $u = F_2$  to  $v_{0,w}$  (corresponding to a stable (LHP) zero). 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.

#### 5.4.4. Simulations

We consider a cooler with data as given in Table 1, and represented by a dynamic process model with 200 states. The simulations in Fig. 13 compare the two alternative transformed inputs (with the setpoint  $v^s$  for  $v_0$  or  $v_{0,w}$  set at the desired value for  $y = T_1$ ). The results are also compared to the open-loop (OL) response with no input transformation (with the input  $u = F_2$  kept constant). The setpoint of the transformed input  $v^s$  is initially at 297 K and changes to 302 K at time  $t = 167$  min. The simulations show responses to step disturbances in  $F_1$ ,  $T_1^0$  and  $T_2^0$  (all measured) and to a step change in the heat transfer parameter  $UA$  (unmeasured which introduces uncertainty).

From the response for the controlled variable ( $y = T_1$ ) in Fig. 13(a), we see that there is a clear benefit of using transformed inputs. From the theory, both transformed inputs give perfect control ( $y = v$ ) at steady state for measured disturbances and this is confirmed by the simulations. However, disturbance rejection is not perfect dynamically because the process dynamics are quite complex and not described by a first-order model. For the unmeasured disturbance in  $UA$  (towards the end of the simulation in Fig. 13(a)), we see, as expected, that we get a steady-state offset for  $y = T_1$  when we use  $v_0$  (red curve) as the transformed input, but not when we use  $v_{0,w}$  (blue curve) which makes use of the state measurement  $w = T_2$ .

Dynamically, we find for this example that the responses are best (fastest) when we use  $v_0$  as the transformed input (red curves). The dynamics with  $v_0$  are similar to the quite fast dynamics of the uncontrolled heat exchanger (green curves). This is expected because there is no feedback to  $v_0$  which can change the dynamics. 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. The effect of this indirect feedback is case dependent and in this case the feedback from  $w$  slows down the response. It does not help to change the tunings of the slave controller ( $C_v$ ), because 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 controller  $C$  to speed up the response, but one must be careful because this may cause instability if there is a measurement delay for  $y = T_1$ .

In summary, since the implementation of  $v_0$  based on the full static model is complex and gives steady-state offset for disturbances in the  $UA$ -value, it is likely that the simpler implementation using  $v_{0,w}$  with a measured  $w = T_2$  is preferred in practice.

Additional examples, including a steam network, CSTR reactor and a pH control problem may be found in the PhD thesis of Zotică [17] and in the Master theses of Bjorvand [18] and Kingstree [19].

**Table 1**

Nominal operating conditions for the heat exchanger (Example 6) from [16].		
Variable	Value	Unit
$F_1$	3	kg/s
$F_2$	5	kg/s
$T_1^s$	297.2	K
$T_2^0$	293	K
$T_1^0$	343	K
$U$	150	W/m <sup>2</sup> /K
$A$	90	m <sup>2</sup>
$V_1 = V_2$	0.45	m <sup>3</sup>
$c_{p1}$	1200	J/kg K
$c_{p2}$	1500	J/kg K
$\rho_1$	980	kg/m <sup>3</sup>
$\rho_2$	890	kg/m <sup>3</sup>

**Table 2**

Slave  $v$ -controller ( $C_v$ ) tunings for heat exchanger example.

Transformed input	$K_I$	$\tau_c$ [s]
$v_0$	−0.125	10
$v_{0,w}$	−0.01	10

## 6. Discussion

### 6.1. Outer controller $C$ for case with ideal transformed inputs

For the case with ideal transformed inputs (Fig. 9), the transformed system includes in theory perfect disturbance rejection and perfect steady-state setpoint tracking. For this reason all simulations in this paper are without the outer controller  $C$ , and to handle setpoint changes in  $y^s$  we made direct changes in the transformed input  $v$ . This was implemented simply by setting  $v = y^s$  (or  $v^s = y^s$  for the feedback implementations), since we in most of the examples choose the tuning parameters ( $B_0 = I$  or  $B = -A$  for  $v_0$  or  $v_A$ , respectively) such that we have  $y = v$  at steady state.

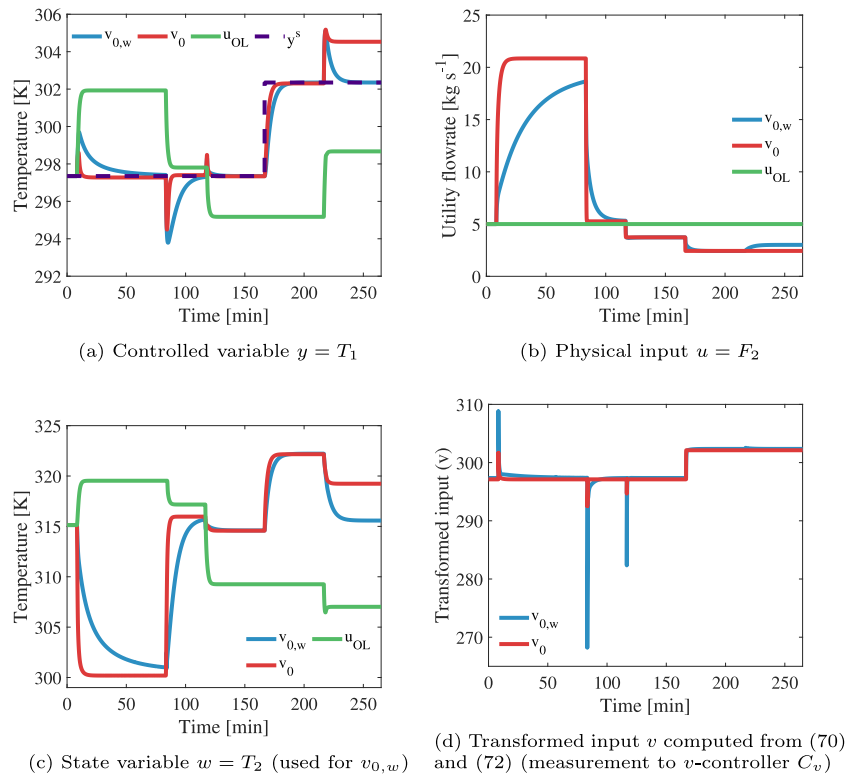
In practice, an outer controller  $C$  that adjusts  $v$  is needed to handle model uncertainty and unmodeled or unmeasured disturbances. In many cases, the outer controller is a rather slow PI-controller and the ability to make fast setpoint changes for  $y$  may be lost. Thus, to retain the direct (and fast) effect from the setpoint  $y^s$  to the transformed input  $v$ , we may add feedforward action from the setpoint  $y^s$ . A simple solution is to use the following two degrees-of-freedom PI-controller  $C$ :

$$v(t) = v(t_0) + K_s y^s(t) - K_c y(t) + K_I \int_{t_0}^t (y^s(t) - y(t)) dt \quad (73)$$

With ideal transformed inputs, we have  $y = v$  at steady state (with no model error). In such cases, we may choose the “feedforward gain” to be  $K_s = I$  to get the same initial setpoint response as for the transformed system without the outer controller  $C$ . On the other hand,  $K_c$  is usually smaller than  $I$ . For example, we may choose  $K_c = 0$  to get pure I-control in  $C$ . The integral gain  $K_I$  in  $C$  will correct for model error and unmeasured disturbances on a slower time scale. The gain matrices  $K_c$  and  $K_I$  should be diagonal because the ideal transformed system from  $v$  to  $y$  is decoupled.

### 6.2. Time scale separation for feedback implementation (alternative B)

The feedback implementation in Fig. 8(b) (alternative B) generates only an approximation of the exact inverse  $g^{-1}$  in Figs. 8(a) and 9, but the error can be neglected if the inner loop (with  $C_v$ ) 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



**Fig. 13.** Dynamic simulation of heat exchanger (Example 6) using the cascade implementation of the transformed input  $v$  with a controller  $C_v$  (Fig. 8(b)). Two choices for the transformed input,  $v_0$  and  $v_{0,w}$  (with setpoint  $v^s$  set equal to the setpoint for  $y$ ), are compared with the open-loop (OL) case with no input transformation (constant  $u$ ). The simulations are with no outer controller ( $C = 0$ ) for the following step disturbances:  $F_1$  from 3 to 4 kg/s at  $t = 8$  min,  $T_2^0$  from 293 to 288 K at  $t = 80$  min,  $T_1^0$  from 343 to 328 K at  $t = 117$  min, setpoint change in  $v^s = y^s$  from 297 to 302 K at  $t = 167$  min, and disturbance in  $U$  from 150 to 100 W/m<sup>2</sup>/K at  $t = 217$  min.

closed-loop time constant of the inner loop involving the slave controller  $C_v$ , and  $\tau_c$  denotes the closed-loop time constant for the outer loop involving the master controller  $C$ . 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  $\tau_c$  is  $(1 - e^{\tau_c/\tau_{c,v}})$ . Thus, the approach to convergence increases from 63% to 95.0% to 99.3% as the time constant ratio  $\tau_c/\tau_{c,v}$  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 [13].

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 [20]. A larger value (larger than 5) allows for robustness to process gain variations which may affect the closed-loop time constants of both the inner and master control loops. Therefore, a time scale separation of 10 or larger is usually recommended in most practical cases. The limiting case of infinite time scale separation corresponds to  $\epsilon = \frac{\tau_{c,v}}{\tau} \rightarrow 0$ , which is the singular perturbation condition in the mathematical literature.

The main fundamental limitation in having a fast response in the inner loop (and thus making  $\tau_{c,v}$  small) is a possible unstable zero (or more generally an effective delay) in the dynamic map  $g_u$  from  $u$  to  $v$  for cases when  $v$  depends on the output (state) variables  $w$  or  $y$ . This is discussed in more detail below.

However, note that unstable zero dynamics from  $u$  to  $v$  are not a particular problem for the feedback implementation in Fig. 8(b), but rather it is a fundamental limitation. To avoid excessive changes (spikes) in the value of  $u$  sent 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 Fig. 8(b)), because we do not want to unnecessary delay this dependency). For example, a first-order filter may be used,  $F = \frac{1}{\tau_F s + 1}$  where  $\tau_F$  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 (18)

In many cases the process is dynamic, but nevertheless we may want to apply the ideal static transformed input  $v_0 = f_0(u, w, d)$  in (13) or (27), where  $f_0$  is the right-hand side of the static process model. Note here that we have here chosen  $B_0 = I$  so that we have  $y = v_0$  at steady state.

When the transformed input  $v_0$  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, assume that we apply the static transformation  $v_0 = f_0(u, w, ..d)$  to the particular dynamic system

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

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

To prove that we retain disturbance rejection for this class of dynamic systems, note that we at steady-state, where  $f(u, w, y, d)$



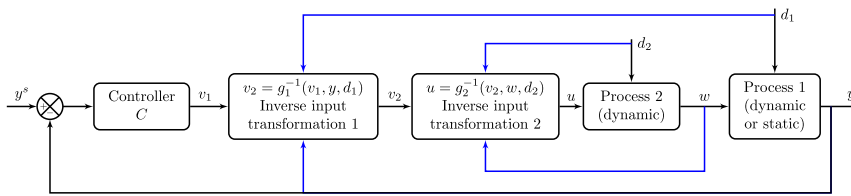


Fig. 14. Chain of input transformations.

= 0, 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 the dynamic system in (18). Furthermore, assume that we generate the exact inverse  $u = f_0^{-1}(v_0, w, d)$  and that the resulting transformed dynamic system becomes

$$\frac{dy}{dt} = f_t(v_0, w, y, d) \tag{74}$$

An example of such a transformed system is given by (57) for the nonlinear level process. Since we have  $y = v_0$  at steady state, and since  $\frac{dy}{dt} = 0$  at steady state, we must have that  $f_t = 0$  when  $y = v_0$ , independent of the value of  $d$  (and  $w$ ).

Assume now that we are initially at steady-state, so that we have  $y = v_0$  and thus  $f_t = 0$ . Next, 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 (75) 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  $\frac{dy}{dt} = f(u, w, y, d)$  in (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 Eqs. (74) is a set of equations of the form

$$\frac{dy_i}{dt} = f_{t,i}(v_0, w, y, d) \tag{75}$$

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 (75) we have that  $\frac{dy_j}{dt} = 0 (j \neq i)$ . 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, also dynamically. Thus, the main advantage of using  $v_A$  (based on the dynamic model (18); similar to feedback linearization) rather than  $v_0$  (based on a static model) when applied to the dynamic system (75) 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. Chain of input transformations

The use of the exact inverse  $u = g^{-1}(v, w, y, d)$  is limited to cases where the relative order from the input  $u$  to the output  $y$  is

0 for the static model case ( $v_0$ ) and 1 for the dynamic model case ( $v_A$ ). By introducing measured state variables  $w$ , we may extend the use of transformed inputs to systems of higher relative order, but this will require the use of feedback (cascade) control for performing the inversion, which implies that the inversion is not perfect as it depends on the tuning of the slave controller ( $C_v$ ).

However, if we have a dynamic model for how  $w$  depends on  $u$ , it is possible in some cases to get exact inversion for systems of higher relative order by using a chain of input transformations, or rather by a chain of inversions on the form (see Fig. 14)

$$v_2 = g_1^{-1}(v_1, y, d_1), \quad u = g_2^{-1}(v_2, w, d_2)$$

Here,  $v_1$  is the transformed input related to the control of  $y$ , whereas  $v_2$  is the transformed input related to the control of  $w$ . Note that  $g_2$  is based on a dynamic model for  $w$ , for example,  $\frac{dw}{dt} = f_2(u, w, d_2)$  (process 2), whereas  $g_1$  may be based on either a dynamic or static model for  $y$ , for example,  $\frac{dy}{dt} = f_1(w, y, d_1)$  or  $y = f_0(w, d_1)$  (process 1). With a chain of transformations we can get perfect disturbance rejection for all disturbances at steady state. We can also get perfect dynamic disturbance rejection for disturbances that enter the model in the same way as the input  $u$ , that is, for disturbances  $d_2$  with the same relative order as the input, but not for disturbances  $d_1$  that have a lower relative order than  $u$ .

For example, assume that we have a continuous process with two mixing tanks in series and we want to control the outlet temperature in the second tank ( $y = T_2$ ) using the heat input to the first tank ( $u = Q_1$ ). Also assume that we have an extra state measurement of the temperature in the first tank ( $w = T_1$ ). In this case the relative order from  $u$  to  $y$  is 2 and perfect disturbance rejection is not possible for disturbances  $d_1$  directly affecting  $y$  (e.g., from another inlet stream to tank 2). However, since the relative order from  $u$  to  $w$  is 1, it is possible, by using a chain of transformations, to get perfect control for disturbances  $d_2$  directly affecting  $w$  (e.g., the inlet feed temperature to tank 1). A more detailed treatment of a similar example is given in the Master Thesis of Kingstree [19].

#### 6.5. Comparison with feedback linearization

The use of ideal transformed inputs  $v_A$  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 of  $v_A$  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  $y$ . Actually, in the feedback linearization literature, it is often assumed to be a chain of linear integrators, but for process control applications it is better to generalize and use a chain of linear first-order systems. However, even with this generalization, there are few, if any, reports of using such chains in practice. There are some reasons for this. First, the feedback linearization theory assumes that all the states  $x$  are measured, which is often not satisfied in process control applications. Second, even if we can measure all the states, the resulting derivation and implementation tend to become complicated, and may not be worth the effort.

An alternative approach, which is simpler but somewhat less general than feedback linearization, is to use a chain of input transformations, as mentioned above, based on using a model for the measured state variables  $w$ . However, also this implementation gets rather complex, and other model-based control methods, like nonlinear model predictive control (MPC), may be better for more complicated dynamic models.

### 6.6. Comparison with nonlinear MPC

MPC provides a generic model-based framework for handling complex control problems that require linearization, decoupling and feedforward control. There are some similarities between nonlinear MPC and the use of ideal transformed inputs. Both methods are based on “open-loop” (feedforward) thinking and feedback from the measurements  $y$  is only introduced as a correction. An advantage with MPC is that it includes weights that makes it possible to deviate from the exact inverse to address more directly the overall control problem to get a trade-off between output performance, input usage and robustness. Thus, with MPC, we are not forced to use exact model inversion, which is the basis of the input transformations and also of feedback linearization. MPC also allows for a much larger class of dynamic models. Constraints on inputs and outputs may also be included more directly with MPC.

On the other hand, the ideal input transformations used in this paper have the advantage that they are more explicit and potentially simpler to understand and implement, especially when based on physical static models. They can easily be simplified to include only decoupling or only parts of the disturbance rejection. They can also easily be combined with conventional feedback controllers  $C$  (e.g., single-loop PID controllers) which may be used to correct for model error and unmeasured disturbances. Unlike MPC, input transformations can be easily implemented within the standard DCS or PLC control platforms used at chemical processing facilities.

A second more fundamental advantage of input transformations is that we may easily use extra measured variables (states)  $w$  to replace model equations. This is generally more difficult with MPC, except for cases where  $w$  is treated as a measured disturbance, which means that we neglect the dependency of  $w$  on  $u$ . With transformed inputs, we have the possibility of generating the input  $u$  by feedback using a cascade implementation (Fig. 8(b)). We then do not need to explicitly know how the measured state  $w$  depends on the input  $u$  as this goes into the response from  $u$  to  $v$ , which we may obtain experimentally and use to tune the slave feedback controller  $C_v$ . Note that we only need a simple dynamic model (e.g., first-order plus delay) to tune  $C_v$  and it does not need to be exact as the inversion (from  $v$  to  $u$ ) is handled by feedback.

For example, this makes it possible with transformed inputs to replace a valve equation by a flow measurement. This is not possible with MPC.

Generally, with MPC, the measurement  $w$  of a state  $x$  can only be exploited if we have a model equation that relates the state  $x$  to the input  $u$ . The measurement  $w$  is then used to correct the prediction by the model, for example, using a Kalman filter. Thus, we cannot easily substitute model equations by measured states  $w$  with MPC.

For the heat exchanger case study, it is not clear how we with MPC can use the measurement  $w = T_2$  to eliminate the heat transfer Eq. (68c) (involving the  $UA$ -value). At least it cannot be done if MPC is based on an experimental model (which is used for most MPC applications) because we cannot here treat  $T_2$  as a measured disturbance. With transformed inputs, we could do this by introducing the transformed input  $v_{0w}$  in (72) which depends on  $w = T_2$ .

### 6.7. Transformed outputs

It is also possible to define transformed outputs

$$z = h(y, w, d) \quad (76)$$

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. When we introduce transformed outputs  $z$ , the transformed input  $v_z$  will be in terms of  $z$  rather than  $y$ , that is,  $v_z = g_z(u, w, z, d)$ . Two alternative implementations are shown in Fig. 15.

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. There is therefore no fundamental advantage of introducing transformed outputs. However, in some cases the use of transformed outputs  $z$  may simplify the implementation of the transformed inputs  $u$  and this may be an important advantage. One example is to introduce the enthalpy  $z = H(T, \dots)$  as a transformed output when the objective is to control temperature  $y = T$  [21]. The reason is that the model may be formulated much simpler in terms of  $z = H$  rather than in terms of  $y = T$ , leading to a much simpler expression for the ideal transformed input  $v_z = g_z(u, w, z, d)$  for  $z$  than for the transformed input  $v = g(u, w, y, d)$  for  $y$ .

### 6.8. Internal instability with exact 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  $u$

$$u = g^{-1}(v, w, y, d) \quad (77)$$

Note that (77) is a purely static expression and therefore by itself does not contain any instabilities. However, when generating the inverse in (77) we are in effect treating  $w$  and  $y$  as measured disturbances, whereas they in reality depend on  $u$ . This dependency may generate internal instability because of the feedback generated when the static inverse  $u = g^{-1}(v, w, y, d)$  in (77) 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.

For the analysis that follows, there is no difference between the variables  $w$  and  $y$ , so to simplify we will for now assume that the transformed input  $v$  is independent of  $y$ .

We have the following general result: Let  $v = g_u(u, d)$  represent the dynamic map from  $u$  to  $v$  when the internal variables  $w$  in the static map  $v = g(u, w, d)$  have been eliminated by using a dynamic model  $w = g^w(u, d)$  for how  $w$  depends on  $u$ . That is,

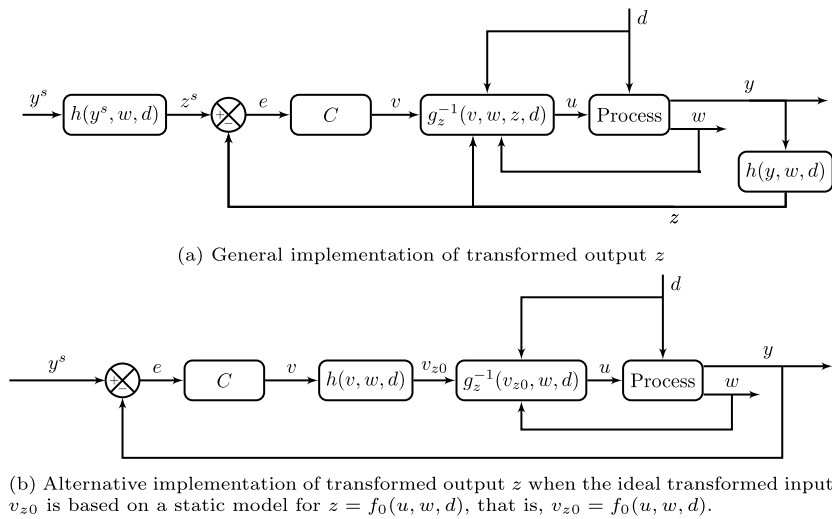


Fig. 15. System with both transformed input  $v$  and transformed output  $z = h(y, w, d)$ .

$g_u(u, d) = g(u, g^w(u, d), d)$ . We then get internal instability when we apply the exact inverse  $u = g^{-1}(v, w, d)$  if the map  $g_u(u, d)$  contains unstable zero dynamics (RHP-zeros in the linear case). This result follows trivially because the inverse will have unstable poles at the unstable zeros of  $g_u$ .

6.8.1. Example: Simple linear system with unstable zeros

As a simple example, consider the system (seemingly static)

$$y = u + w + d \tag{78}$$

for which we from propose from (27) (with  $B_0 = I$ ) to use as the ideal transformed input the right-hand-side of (78)

$$v = g(u, w, d) = u + w + d \tag{79}$$

For implementation, (79) may be solved with respect to  $u$  to get the “inverse input transformation”

$$u = g^{-1}(v, w, d) = v - w - d \tag{80}$$

This may be implemented as in Fig. 8(a) and it gives the (ideal) transformed system

$$y = v \tag{81}$$

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 (80). 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  $-2$

$$w = g^w(s) = \frac{-2u}{4s + 1} \quad \text{or} \quad \frac{dw}{dt} = -0.25(2u + w) \tag{82}$$

Note from (79) that the direct static effect of  $u$  on  $v$  has a gain of 1, whereas from (82) the indirect dynamic effect of  $u$  on  $v$  (through  $w$ ) has a steady-state gain of  $-2$ , that is, it is larger and in the opposite direction. The combined effect causes an unstable (RHP) zero from  $u$  to  $v$ . To see this, eliminate  $w$  from (79) using (82) to get

$$v = g_u(u, d) = \frac{4s - 1}{4s + 1}u + d \tag{83}$$

which has an unstable (RHP) zero at  $z = 1/4$ . This gives internal instability if we use the exact inverse in (79). To see this, solve

(83) with respect to  $u$  to get

$$u = \frac{4s + 1}{4s - 1}(v - d) \tag{84}$$

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

$$w = \frac{-2}{4s - 1}(v - d) \tag{85}$$

The two instabilities in Eqs. (84) and (85) cancel each other when put into (78) 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  $u$ .

6.8.2. Linear stability analysis of transformed system

We now want to discuss more generally how we may get instability when applying the exact inverse transformation (Fig. 8(a)). We allow for any transformed input  $v = g(u, w, y, d)$ , not only the ones derived systematically from a model as shown in Section 4. For simplicity we consider linear systems. Nonlinear systems may have additional stability problems, but the local stability can be assessed from a linear analysis.

When is the transformed system stable? We know from the examples above that it is not sufficient to study the stability from  $v$  to  $y$  because of the possibility for internal instability when the transformed input depends on the dependent (state) variables  $w$ . Therefore, to eliminate the issue of internal stability, we will consider the stability of the “closed-loop” map from  $v$  to  $u$ , which will be unstable if we have internal instability, e.g., see (84). By “closed-loop” we mean that we have included the effect of feedback through the dependent variables  $w$  and  $y$ .

For the analysis that follows, there is no difference between the variables  $w$  and  $y$ , so to simplify we will for now assume that the transformed input is independent of  $y$

$$v = g(u, w, d) \tag{86}$$

Here all the variables may be vectors. The transformation in (86) is static so it can be linearized to give

$$dv = K_u du + K_w dw + K_d dd \tag{87}$$

The linearized process dynamics from  $u$  to  $w$  can be written

$$dw = G^w(s)du \tag{88}$$

We here set  $dd = 0$  since  $d$  does not matter for local (linear) stability considerations. Substituting (88) into (87) gives the response from  $u$  to  $v$

$$dv = (K_u + K_w G^w(s))du = (I + L_w(s))K_u du \quad (89)$$

where

$$L_w = K_w G^w(s)K_u^{-1} \quad (90)$$

Physically,  $L_w(s)$  represents the ratio of the indirect effect  $K_w G^w$  (through  $w$ ) and the direct effect  $K_u$  of  $u$  on the transformed input  $v$ . Solving (89) respect to the input  $u$  gives the “closed-loop” input that achieves the desired perfect control of  $v$

$$du = K_u^{-1}(I + L_w(s))^{-1}dv \quad (91)$$

Mathematically, we note that the linearized transformed system in (91) corresponds to the closed-loop response from  $dv$  to  $du$  of a system with negative feedback and a loop transfer function  $L_w(s)$ . This is very useful because of the existence of stability conditions for such systems (e.g. [13]). These stability conditions include the small gain theorem as well as the more general Bode stability condition.

The small gain theorem says that we have stability if the norm of  $L_w$  is less than 1 (e.g.,  $|L_w(j\omega)|$  is less than 1 at all frequencies if we use the H-infinity norm). For a scalar system, this says that if the indirect effect  $K_w G^w$  of  $u$  on  $v$  (via  $w$ ) is less in magnitude than the direct effect  $K_u$ , then the transformed system is internally stable, that is, there are no unstable zeros in the map from  $u$  to  $v$ . This is a useful condition which yields significant insight. However, the small gain theorem gives only a sufficient condition for stability, but not necessary. The Bode stability condition is tighter because it provides a necessary and sufficient conditions for stability.

Consider again the example in (79) and (82). From (79) we have  $K_u = 1$ ,  $K_w = 1$ , and thus  $L_w(s) = G^w(s) = -\frac{2}{4s+1}$ . The steady-state gain  $|L_w(0)| = 2$  is larger than 1, but from the small gain theorem we cannot conclude that the system is unstable. However, the Bode condition says that we need to look at the gain at the frequency where the phase shift is  $-180^\circ$ . Since the steady state gain of  $L_w(s)$  is negative we have that phase shift at steady state is  $-180^\circ$ . Thus  $|L_w(jw_{180})| = |L_w(0)| = 2$  and we conclude from the Bode stability condition that the transformed system is (internally) unstable, as expected.

### 6.8.3. Relationship between internal stability and stable zero dynamics for output $y$ (for systematic transformed inputs $v_0$ and $v_A$ )

We consider here the special case when the ideal transformed input has been derived from a model using (27) or (30). We can then instead look at the zero dynamics for the output  $y$  (rather than for  $v$ ). This follows because  $y$  and  $v$  are directly related in this case. In the static case, we have from (15) that the transformed system is  $y = B_0 v_0$  (even dynamically for the case when  $w$  depends dynamically on  $u$ ). Similarly, for the dynamic case, we see from (21) that  $v_A$  and  $y$  are directly related. In particular, the zero dynamics will be the same. It then follows, that to have stability of the transformed system when  $v$  is derived using Eqs. (13) or (19) we must require that the map from the physical input  $u$  to the output  $y$  has stable zero dynamics, that is, the process has stable zero dynamics [5].

## 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 analyze, implement and derive model-based nonlinear calculation blocks and cascade control schemes which are frequently used in practice for industrial processes.

The starting point is often a nonlinear static model,  $y = f_0(u, d)$ . From (13) with  $B_0 = I$ , the ideal static transformed input is 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 Fig. 8(a), but if the equations are complex, we may use feedback control as a “trick” to solve the equations by using the cascade implementation in Fig. 8(b) with a slave  $v$ -controller.

The model may often be written in a simpler form,  $y = f_{0w}(u, w, d)$ , by replacing parts of the model by a measured dependent variable (state)  $w$ . The most common is to use a measured flow to replace a valve-like equation, like (6) or (52), or to use a measured temperature to replace the energy balance or parts of it. In such cases, the cascade implementation in Fig. 8(b) is often preferred to invert the transformation  $v_0 = f_{0w}(u, w, d)$ . First, it may happen that the function  $f_{0w}$  does not depend explicitly on the input  $u$  and then it is not possible to use a model-based inverse. Second, there is a potential problem with internal instability if we use the model-based inverse, but instability will not occur with the cascade implementation because the tuning of the inner controller  $C_v$  is based on the experimental response from  $u$  to  $v_0$ . Internal stability with the exact inverse occurs if the indirect (dynamic) effect of  $u$  on  $v_0$  through  $w$  is large compared to the direct (static) effect of  $u$  on  $v_0$ , such that the overall response from  $u$  to  $v_0$  has unstable zero dynamics (inverse response in the linear scalar case). This is not very common, and with an ideal transformed input  $v_0$  it only happens when the original system response from the input  $u$  to the output  $y$  has unstable zero dynamics (inverse response). In such cases, control of  $y$  is fundamentally difficult with any control approach.

It is also possible to derive ideal transformed inputs,  $v_A$ , based on a dynamic model,  $\frac{dy}{dt} = f(u, y, d)$ ; see (30). This approach is closely related to the theory of feedback linearization. At first sight, this seems to be a much more powerful approach than with 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 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 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 will be drifting for unknown disturbances. The main advantage with  $v_A$  compared to  $v_0$  is that it linearizes the system dynamically, which may simplify the design of the outer controller  $C$  in some cases. In

summary, we recommend that the engineer starts with static models when deriving transformed inputs.

The use of ideal transformed inputs  $v$  may in theory produce no offset at steady state ( $y = v$ ), but this is based on feedforward control and assumes an exact model and perfect measurements of the disturbances. This is not realistic, so 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 at steady state. The objective of the outer controller is to correct for errors in the model and measurements and to reject unmeasured or unmodeled disturbances. The outer controller should include integral action to get offset-free control at steady state.

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 be designed and tuned independently. Second, it is easier to understand and modify by engineers and operators, and it reduces errors in the implementation.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

No data was used for the research described in the article.

### Acknowledgments

The authors gratefully acknowledge fruitful discussions with Krister Forsman and Christian Holden. This work was 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).

### Appendix. Tuning parameter $A$ for ideal dynamic transformed inputs $v_A$

Consider the nonlinear system model  $\frac{dy}{dt} = f(u, y, d)$ , which can be linearized to get

$$\frac{dy}{dt} = df = \tilde{A}y + \tilde{B}du + \tilde{B}_d dd \quad (\text{A.1})$$

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  $\frac{dy}{dt} = Ay + Bv_A$ . 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 (A.1).

Furthermore, from (19) the linearized transformed input becomes

$$dv_A = B^{-1}(df - Ady) = B^{-1}(\tilde{B}du + \tilde{B}_d dd) \quad (\text{A.2})$$

and we find that  $dv_A$  is independent of  $dy$ .

Thus, with the choice for  $A$  in (25), there is no feedback from  $y$  on the transformed input  $v_A$  at the nominal 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

$$A = \text{diag}(\tilde{A}) = \text{diag}\left(\frac{\partial f}{\partial y}\right)_* \quad (\text{A.3})$$

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 compromise between 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  $C$ .

### References

- [1] F.G. Shinskey, *Controlling Multivariable Processes*, Instrument Society of America, 1981.
- [2] E. Welfonder, Dynamic interactions between power plants and power systems, *Control Eng. Pract.* 7 (1999) 27–40, [http://dx.doi.org/10.1016/S0967-0661\(98\)00146-4](http://dx.doi.org/10.1016/S0967-0661(98)00146-4).
- [3] F.G. Shinskey, *Process Control Systems*, McGraw-Hill, 1967.
- [4] J. Lee, R. Mukherjee, H.K. Khalil, Output feedback performance recovery in the presence of uncertainties, *Systems Control Lett.* 90 (2016) 31–37.
- [5] A. Isidori, *Nonlinear Control Systems*, third ed., in: *Communications and Control Engineering*, Springer London, London, 1995.
- [6] C. Kravaris, C.-b. Chung, Nonlinear state feedback synthesis by global input / output linearization, *AIChE J.* 33 (1987).
- [7] G. Bastin, D. Dochain, *On-Line Estimation and Adaptive Control of Bioreactors*, Elsevier Science Publishers B.V., 1990.
- [8] H.K. Khalil, *Nonlinear Control*, second ed., Pearson, Boston, 2015.
- [9] M.A. Henson, D.E. Seborg (Eds.), *Nonlinear Process Control*, Prentice-Hall, 1997.
- [10] A. Isidori, personal communication, 2020.
- [11] C. Zotică, N. Alsop, S. Skogestad, Transformed manipulated variables for linearization, decoupling and perfect disturbance rejection, *IFAC-PapersOnLine* 53 (2) (2020) 4052–4057, <http://dx.doi.org/10.1016/j.ifacol.2020.12.2331>, 21st IFAC World Congress.
- [12] S. Skogestad, Simple analytic rules for model reduction and PID controller tuning, *J. Process Control* 13 (4) (2003) 291–309, [http://dx.doi.org/10.1016/S0959-1524\(02\)00062-8](http://dx.doi.org/10.1016/S0959-1524(02)00062-8).
- [13] S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, second ed., Wiley, 2005.
- [14] A. Reyes-Lúa, C. Zotică, T. Das, D. Krishnamoorthy, S. Skogestad, Changing between active constraint regions for optimal operation: classical advanced control versus Model Predictive Control, *Comput.-Aided Chem. Eng.* 43 (2018) 1015–1020.
- [15] N. Soave, M. Barolo, On the effectiveness of heat-exchanger bypass control, 2021, <http://dx.doi.org/10.3390/pr9020244>, *Processes*, Vol. 9 URL: <https://www.mdpi.com/2227-9717/9/2/244>.
- [16] S. Skogestad, *Chemical and Energy Process Engineering*, CRC Press, 2008.
- [17] C. Zotică, *Optimal Operation and Control of Thermal Energy Systems* (Ph.D. thesis), Norwegian University of Science and Technology, 2023, URL: [https://folk.ntnu.no/skoge/publications/thesis/2023\\_zotica/](https://folk.ntnu.no/skoge/publications/thesis/2023_zotica/).
- [18] S. Bjorvand, *Control Structures with Embedded Process Knowledge* (Master thesis), Norwegian University of Science and Technology, 2020, URL: <https://folk.ntnu.no/skoge/diplom/diplom20/simen-bjorvand/>.
- [19] C.R. Kingtree, *Application of Transformed Manipulated Variables to Cascade Control Systems for Linearization and Disturbance Rejection* (Master thesis), University of Edingborough and Norwegian University of Science and Technology, 2021, URL: <https://folk.ntnu.no/skoge/diplom/diplom21/kingstree/>.
- [20] M. Baldea, P. Daoutidis, Control of integrated process networks—A multi-time scale perspective, *Comput. Chem. Eng.* 31 (5) (2007) 426–444, <http://dx.doi.org/10.1016/j.compchemeng.2006.05.017>, URL: <https://www.sciencedirect.com/science/article/pii/S009813540600130X>, ESCAPE-15.
- [21] C. Zotică, R. Mocholí Montañés, A. Reyes Lua, S. Skogestad, Control of steam bottoming cycles using nonlinear input and output transformations for feedforward disturbance rejection, in: *The 13th IFAC Symposium on Dynamics and Control of Process Systems, Including Biosystems (DYCOPS)*, Busan, Republic of Korea, 2022.