**Abstract:** The objective of this work is to find new transformed manipulated variables (MVs) for nonlinear systems which linearizes and decouples the system, and gives perfect feedforward control for disturbances (at least at steady-state). The proposed new input transformation is more general than feedback linearization in that it also allows for multiple-inputs multiple-outputs (MIMO) systems, disturbances, a more general class of models and introduces a tuning parameter $\tau_0$. The key idea is to use decentralized SISO controllers for the output $y$ using the new transformed inputs $v$ as MVs. The SISO controllers give $v$, and a nonlinear calculation block solves algebraic equations which explicitly gives the original input $u$ as a function of the controller output $v$, output $y$ and disturbances $d$. The calculation block also handles decoupling, and feedforward action from the disturbance $d$. This new procedure can be applied both for static and dynamic processes, which is typical in process control.

**Keywords:** linearization, decentralized control, decoupling problems, process control, disturbance rejection

1. **INTRODUCTION**

Different techniques for controlling nonlinear systems have been presented in the literature, including: linear controllers designed using a linearized model around an operating point (e.g. PID controllers, linear model predictive control etc.); adaptive control (Åström and Wittenmark, 2008); nonlinear model predictive control (Rawlings et al., 2017) or nonlinear control (e.g. feedback linearization (Isidori, 1989; Khalil, 2015; Nijmeijer and van der Schaft, 1990a), input-output linearization (Henson and Seborg, 1997), disturbance decoupling (Huijberts et al., 1991), input decoupling (Isidori et al., 1981; Balchen et al., 1988; Nijmeijer and van der Schaft, 1990b) elementary nonlinear decoupling (Balchen, 1998) etc.).

The objective of this work is to find new manipulated variables that transform a nonlinear process into a linear one (preferably first order), give decoupling and perfect disturbance rejection. The literature presents a few approaches with similar objectives (often with different names) though with different methodologies, and we discuss a few of them. Feedback linearization received a large interest in the control literature starting with the differential geometry approach introduced by the work of Isidori et al. (1981) and peaking in the work of Isidori (1989), Nijmeijer and van der Schaft (1990a) and Khalil (2015). A comprehensive overview and analysis for nonlinear process control is presented in the work of Henson and Seborg (1991). However, to the best of the authors knowledge this methods have not been practically implemented for chemical processes. We will explain briefly why this is not used for chemical processes.

Consider the nonlinear process given by Eq.1

$$\dot{x} = f(x) + g(x)u, \quad y = h(x)$$

where, $x$ are the states, $u$ is the input, $y$ is the output, and $f$ and $g$ are nonlinear functions relating the states $(x)$ and input $u$ to the differential of the state $(\dot{x})$ respectively, and $h$ is a nonlinear function relating $x$ to $y$.

Feedback linearization works by transforming a nonlinear $n$th order system given by Eq.1 into a new system described by a chain of $r$ (i.e. relative degree with $r \leq n$) integrators, therefore linear and controllable. The method consists of two steps (Isidori, 1989):

Step 1: a change of coordinates using directional Lie derivatives, defined locally around some point $x^0$.

Step 2: a state feedback, defined locally around the same point $x^0$.

Input-output linearization is another nonlinear technique similar to feedback linearization, but applied to systems for which the state-inputs equations cannot be linearized (usually with $r < n$ and RHP-zero). It partly linearizes the system, that is, it linearizes the output-input behaviour, while keeping some nonlinear state-input equa-
tions (Isidori, 1989; Henson and Seborg, 1997). Feedback linearization is more suitable for stabilization purposes, where as input-output linearization can be applied for systems for which the output is specified a priori which makes it more appropriate for process control applications (Henson and Seborg, 1997).

The main limitations of these linearization methods are:

- lack of robustness to model uncertainty in as it requires an accurate process inverse;
- difficult to extend to multivariate systems as it needs a type of non-robust decoupling control;
- cannot explicitly handle process constraints;
- all the states must be available for measurement, or can be estimated;
- inability to deal with uncertainty in RHP-zeros and time delays.

These limitations may be acceptable for mechanical systems which inherently have few states that can be easily measured or estimated. Moreover, mechanical system are often integrating processes, and thus transforming them into a chain of integrators does not necessarily bring additional control limitations. However, this is rarely the case for most process control applications, and this is arguably the reason feedback linearization is yet to be implemented in chemical processes.

Elementary nonlinear decoupling on the other hand, generates a directly invertible system based on designing of a property transformation of the state $x$ and generating an input $u$ such that the property transformation has the desired rate of change. For systems with relative degree one, it turns a $n$th order system into one linear integrator (Balchen, 1998).

2. METHODOLOGY

The principle of our proposed method is shown in the block diagram in Fig. 1, where, $y$ is the outputs vector, $u$ is the original inputs vector (manipulated variables MVs), $v$ is the new inputs (transformed MVs) vector, $d$ is the disturbance vector, $e$ is error vector, $y^s$ is the setpoint vector.

![Fig. 1. Proposed method for linearization, decoupling and perfect disturbance rejection.](image)

In Fig. 1, a decentralized PI-controller computes the transformed input $v$, and the original input $u$ is back-calculated by numerically solving a set of algebraic equations with given outputs $y$ and disturbances $d$. We discuss the structure of each block in Fig.1 in the following.

2.1 Assumptions

- as many outputs (i.e. differential equations) as inputs (i.e. $n_y = n_u$);
- all disturbances can be measured.

Thus, we can handle low-order systems, but this is often the case in process control applications. We present two cases:

1. Simple input transformation (that gives an integrating process, similar to feedback linearization)
2. Refined input transformation (that introduces a new tuning parameter $\tau_0$ to give a first order process).

2.2 Simple input transformation

Assume that we can write the nonlinear dynamic model as shown in Eq. 2, which is more general than Eq. 1 (for simplicity; we consider two controlled variables (CVs), $y_1$ and $y_2$, two MVs, $u_1$ and $u_2$, and a disturbance vector $d$ without compromising the generality of the method).

$$
\frac{dy_1}{dt} = f_1(u_1, u_2, d, y_1, y_2) \quad (2a)
$$

$$
\frac{dy_2}{dt} = f_2(u_1, u_2, d, y_1, y_2) \quad (2b)
$$

We follow the idea of the classical nonlinear control method of feedback linearization, and introduce two new transformed input variables in Eq. 3 (input functions) which simply are the right hand side of the differential Eq. 2.

$$
v'_1 = f'_1(u_1, u_2, d, y_1, y_2) \quad (3a)
$$

$$
v'_2 = f'_2(u_1, u_2, d, y_1, y_2) \quad (3b)
$$

We then get have two decoupled linear integrating systems, Eq. 4, which also are independent of disturbances.

$$
\frac{dy_1}{dt} = v'_1 \quad (4a)
$$

$$
\frac{dy_2}{dt} = v'_2 \quad (4b)
$$

With $v'$ as the controller outputs (or transformed inputs to the process), this is a linear decoupled system for which controller design in principle is straightforward. We assume that $d$ is measured, so that the physical input $u$ can be back-calculated from $v'$ using a calculation block. However, Eq. 4 is a set of integrating systems, and integrating systems are not easy to control.

Limitations. The above approach cannot handle static systems. More generally, it will not work well for cases where the original dynamics are very fast, because we are replacing any dynamics by an integrating system by introducing an implicit feedback through the variable transformation in Eq. 3. In general, integrating systems are difficult to control, so the transformation used in feedback linearization may introduce unnecessary limitations. As mentioned below, we will use it for integrating processes only.

2.3 Refined input transformation

Because of the mentioned limitations of the simple input transformation, we rewrite the model Eq. 2 slightly and introduce the new tuning parameter $\tau_0$. The reason is to transform the process into a first-order system instead of an integrating one. To do this, we assume that we can write the nonlinear model with the outputs (CVs) separated from the other variables as follows (for simplicity we consider two CVs, $y_1$ and $y_2$, two MVs, $u_1$ and $u_2$, and
a disturbance vector \( d \), without reducing the generality of the method):

\[
\begin{align*}
\tau_1 \frac{dy_1}{dt} + y_1 &= f_1(u_1, u_2, d, y_1, y_2) \quad (5a) \\
\tau_2 \frac{dy_2}{dt} + y_2 &= f_2(u_1, u_2, d, y_1, y_2) \quad (5b)
\end{align*}
\]

Comparing Eq. 5 with Eq. 2, we see that \( f_1 = \tau_1 f'_1 + y_1 \). Now we introduce two new transformed input variables (input functions) as the right hand side of Eq. 5, resulting in Eq. 6.

\[
\begin{align*}
v_1 &= f_1(u_1, u_2, d, y_1, y_2) \quad (6a) \\
v_2 &= f_2(u_1, u_2, d, y_1, y_2) \quad (6b)
\end{align*}
\]

where we assume that \( d \) is measured.

We then have two decoupled linear systems, both first-order and independent of disturbances, as shown in Eq. 7.

\[
\begin{align*}
\tau_1 \frac{dy_1}{dt} + y_1 &= v_1 \quad (7a) \\
\tau_2 \frac{dy_2}{dt} + y_2 &= v_2 \quad (7b)
\end{align*}
\]

### 2.4 Controller design

The key idea is now to use decentralized SISO controllers (Eq. 8) for controlling \( y = [y_1 \ y_2]^T \) using \( v = [v_1 \ v_2]^T \) as MVs.

\[
u(t) = K_C \left( c(t) + \frac{1}{\tau_I} \int_0^t c(t) dt \right) \quad (8)
\]

where \( K_C \) is the proportional gain and \( \tau_I \) is the integral time. To tune the PI-controller we may use a systematic tuning method such as the SIMC tuning rules (Skogestad, 2003). For a transformed system in the form of Eq. 7, we use a PI-controller given by Eq. 9.

\[
\begin{align*}
K_C &= \frac{1}{k} \frac{\tau}{\tau_C + \theta} \quad (9a) \\
\tau_I &= \min(\tau, 4(\tau_C + \theta)) \quad (9b)
\end{align*}
\]

where, \( k \) is the process gain, \( \tau \) is the open loop time constant and \( \theta \) is the time delay.

For static system (i.e. \( \tau_0 = 0 \) in Eq. 7) we use a pure PI-controller given by Eq. 10.

\[
K_I = \lim_{\tau \rightarrow \infty} \frac{K_C}{\tau_I} = \frac{1}{k} \frac{1}{\tau_C + \theta} \quad (10)
\]

where \( K_I \) is the integral gain.

### 2.5 Calculation block

For finding the actual \( u \) from \( v \), we use a static calculation block that inverts Eq. 6, resulting in Eq. 11, where we assume \( f^{-1} \) exists. When we write the equations in the form Eq. 5 rather than Eq. 2, we will quite often find that \( v \) in Eq. 6 is independent of the outputs \( y \), at least nominally. Thus, solving Eq. 6 with respect to \( u \), frequently avoids the implicit nonlinear static feedback resulting from solving Eq. 3 with respect to \( u \).

\[
u = f^{-1}(v, y, d) \quad (11)
\]

Eq. 11 can be solved algebraically (i.e. explicitly) or numerically (i.e. by using a numeric solver or by using a fast inner loop PI-controller).

The time constants \( \tau_0 \) are tuning parameters, but it seems reasonable to keep them close to the original systems dynamics. For a static model equation we have that the time constant is zero, e.g. \( \tau_0 = 0 \), and \( y_1 = v_1 \). Note that this means that the output from the controller \( (v_1) \) is equal to the CV \( (y_1) \). On the other hand, a pure integrating system, such as a liquid level, would correspond to an infinite time constant. For these cases we write the model on the original form in Eq. 4, that is, \( \frac{dy_1}{dt} = v_1 \) with \( v_1 = f'_1 \).

As indicated, the approach in Eq. 2 and Eq. 3, where we introduce new inputs \( v' \) and end up with an integrating system Eq. 4, is closely related to the classical nonlinear control method of feedback linearization which considers SISO systems of the form (here written for the case with one differential equation, that is, \( n = 1 \)):

\[
\frac{dy}{dt} = f(y) + g(y)u \quad (12)
\]

With \( n = 1 \) the new input is \( v = f(y) + g(y)u \) and we end up with an integrating system \( \frac{dy}{dt} = v \). This is identical to Eq. 2.3 and 4, except that we in Eq. 2 allow for a more general right hand side than in Eq. 12. The proposed new input transformation is more general than feedback linearization in that it also allows for MIMO systems, disturbances, a more general class of models and introduces a tuning parameter \( \tau_0 \). The limitation with the new approach compared to feedback linearization is that we must have a low-order system (with as many number of inputs \( u \) as number outputs \( y \)).

### 2.6 Output transformation

The principle of both input and output transformation is illustrated in Fig. 2. For some chemical processes, the nonlinearities arise from state-measurement relationship, e.g. pH, or density measurement. Therefore, we may want to introduce an output transformation, which is also a static calculation block. In addition, the structure (i.e. algebraic equations) of the input calculation block is fixed from the start and an output transformation block may make the system more robust. For such systems, we may want to introduce new outputs transformation that linearizes the input \( (v) \)-state \( (x) \) behaviour, where the choice of \( x \) is a degree of freedom.

![Fig. 2. Input and output transformation for linearization, decoupling and perfect disturbance rejection.](image-url)
(2) Control of the hot stream temperature of a heat exchanger using a static model to derive the input transformation and construct the calculation block, and a different dynamic lumped model for the real process (plant model mismatch).

3.1 Case 1. Control of flow and temperature in a mixing process

Fig. 3 shows the mixing process with two inflows and one outflow that we are analyzing.

Fig. 4 shows the block diagram of the proposed inputs transformation for the mixing process.

\[
\begin{align*}
v_1 & = F_1 + F_2 \\
v_2 & = \frac{\tau_0}{m} (F_1 T_1 + F_2 T_2) + \left( 1 - \frac{\tau_0}{\tau_r} \right) T
\end{align*}
\]

With the new transformed input, \( v_1 \) and \( v_2 \), the new system is given by Eq. 16, which represents two decoupled processes with no effect from disturbances.

\[
y_1 = v_1 \\
\tau_0 \frac{d y_2}{dt} + y_2 = v_2
\]

In transfer function matrix form, from input \( v = [v_1 \ v_2]^T \) and disturbances \( d = [d_1 \ d_2]^T \) to outputs \( y = [F \ T]^T \), the system can be rewritten as given in Eq. 17.

\[
y(s) = G(s)v(s) + G_d(s)d(s)
\]

Calculation block. Algebraic solver The calculation block solves Eq. 15 for \( v_1 \) and \( v_2 \) gives inputs \( v_1 \) and \( v_2 \), outputs \( y_1 = F \) and \( y_2 = T \) and disturbances \( d \). In this case, this a linear system, Eq. 18. We select the new tuning parameter equal to the nominal residence time, i.e. \( \tau_0 = 10s \).

\[
\begin{bmatrix}
 v_1 \\
v_2
\end{bmatrix} =
\begin{bmatrix}
 1 & 1 & \tau_0 (T_1 - T) \\
 1 & \tau_0 (T_2 - T)
\end{bmatrix}^{-1}
\begin{bmatrix}
 v_1 \\
v_2 - T
\end{bmatrix}
\]

Note that the matrix inverted in Eq. 18 loses rank when \( T_1 \neq T_2 \). However, this is physically less likely to happen.

Simulation results Table 1 shows the nominal operating conditions (marked with *) for the mixing process. Note that at nominal the two inputs are equal \( (F_1^* = F_2^*) \), which makes the process highly coupled and difficult to control using conventional PID-controllers.

<table>
<thead>
<tr>
<th>Variable</th>
<th>( F_1^* )</th>
<th>( F_2^* )</th>
<th>( F^* )</th>
<th>( T_1^* )</th>
<th>( T_2^* )</th>
<th>( T^* )</th>
<th>( m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>50</td>
<td>35</td>
<td>100</td>
</tr>
<tr>
<td>Unit</td>
<td>kg/s</td>
<td>kg/s</td>
<td>kg/s</td>
<td>°C</td>
<td>°C</td>
<td>°C</td>
<td>kg</td>
</tr>
</tbody>
</table>

In addition, we include a constant delay measurement of 5 s.

Table 2 shows the controller tunings. Note that we use an I-controller for controlling \( y_1 = F \) using \( v_1 \), and an PI-controller for controlling \( y_2 = T \) using \( v_2 \).

Table 2. Case 1 controller tunings

<table>
<thead>
<tr>
<th>MV+CV</th>
<th>( \tau_C ) (s)</th>
<th>( K_C )</th>
<th>( K_I )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_1 \rightarrow y_1 )</td>
<td>1</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>( v_2 \rightarrow y_2 )</td>
<td>1</td>
<td>100</td>
<td>25</td>
</tr>
</tbody>
</table>

Fig. 5 shows the response for \( y_1 = F \), Fig. 6 shows the response for \( y_2 = T \), and Fig. 7 shows the response for \( u_1 = F_1 \) and \( u_2 = F_2 \) for a step increase in disturbance \( d_1 = T_1 \) of 3 °C at time 50 s, and another step increase in disturbance \( d_2 = T_2 \) of 5 °C at time 200 s .

3.2 Case 2. Control of hot outlet temperature of a heat exchanger

The main idea for this example is to test the robustness of our method by using a static model (ε-NTU method)
Controller
Calculation block
(Static ϵ - NTU)
Heat exchanger
(dynamic cells model)

Fig. 8. Heat exchanger with one original MV (u = Fc) and one CV (y = Th).

Fig. 9 shows the block diagram of the proposed method applied to the heat exchanger.

Fig. 9. Block diagram of the proposed method for the heat exchanger using a static model for the calculation block, and a dynamic model for the process.

The original input of the process is the cold side flow: 
\[ u = F_c \ \text{[kg/s]} \]. The process output is the hot side outlet temperature \( T_h \); \[ y = T_h \ \text{[°C]} \]. The main disturbances that we consider are the hot side flow and the inlet temperatures of the two flows: 
\[ d_1 = T_c^{in} \ \text{[°C]} \]; 
\[ d_2 = T_h^{in} \ \text{[°C]} \]; 
\[ d_3 = F_h \ \text{[kg/s]} \].

Following the procedure from Section 2, we find the transform inputs \( v \) with the objective of decoupling and perfect disturbance rejection. We use a static model \( ϵ \)-NTU (number of transfer units) \( \) (Welty, 2008).

\[
T_h = (1 - ϵ_h)T_h^{in} + ϵ_hT_c^{in} \quad \text{(19a)}
\]
\[
T_c = ϵ_cT_c^{in} + (1 - ϵ_c)T_h^{in} \quad \text{(19b)}
\]
\[
ϵ_c = \frac{1}{C - \exp(-NTU(C - 1))} \quad \text{(19c)}
\]
\[
ϵ_h = ϵ_cC \quad \text{(19d)}
\]
\[
C = \frac{F_cC_p}{F_hC_p} \quad \text{(19e)}
\]
\[
NTU = \frac{UA}{F_cC_p} \quad \text{(19f)}
\]

We define the transformed inputs \( v \) as the right hand side of Eq. 19a resulting in Eq. 20.

\[
v = T_h \quad \text{(20)}
\]

With the new transformed input, \( v \), the new system is given by Eq. 21, with no effect from disturbances.

\[
y = v \quad \text{(21)}
\]

In transfer function matrix form, from input \( v \) and disturbances \( d \) to output \( y \), the system can be rewritten as given in Eq. 22.

\[
y(s) = G(s)v(s) + G_d(s) \quad \text{(22)}
\]

with

\[
G(s) = 1, \quad G_d(s) = 0
\]

Calculation block. Algebraic solver The calculation block numerically solves Eq. 19 for \( u \) given inputs \( v \) and disturbances \( d \). Note that we are using \( v = y = T_h \) in the calculation block, and select the new tuning parameter to \( \tau_0 = 0 \) because we use a static heat exchanger map to derive the transformed input.

Dynamic cells heat exchanger model The process is represented by the dynamic lumped model given in Eq. (23), where the heat exchanger is discretized in space in \( N \) cells. The boundary conditions for cell \( i \) is \( T_h^0 = T_h^{in} \), and for cell \( i = N \) is \( T_c^{N+1} = T_c^{in} \). Note that with infinite cells, the dynamic model has the same steady-state values as the static model. Wall capacities are neglected.

\[
\frac{dT_h^i}{dt} = \frac{F_c}{\rho_hV_h^c}(T_h^{i+1} - T_h^i) + \frac{UA(T_h^0 - T_h^i)}{N\rho_hV_h^cC_p} \quad \text{(23a)}
\]
\[
\frac{dT_c^i}{dt} = \frac{F_h}{\rho_cV_h^c}(T_c^{i+1} - T_c^i) + \frac{UA(T_h^i - T_c^i)}{N\rho_hV_h^cC_p} \quad \text{(23b)}
\]

\[ \forall i \in 1 \ldots N \]

where \( c \) is the cold side, \( h \) is the hot side, \( V \) is the cell volume, \( U \) is the heat transfer coefficient, \( A \) is the heat transfer area, \( \rho \) is density and \( C_p \) is specific heat.

Simulation results The nominal operating conditions (marked with *) for the heat exchanger example are shown
in Table 3. Note that the nominal (steady-state) values for the static \( c \)-NTU (Eq. 19) and cells models (Eq. 23) are slightly different because we cannot use an infinite number of cells (\( N = 200 \) cells in simulations). This means that we need to use slower control, i.e. large \( \tau_C \), to be robust and account for the steady-state gain uncertainty.

Table 3. Case 2 nominal operating conditions

<table>
<thead>
<tr>
<th>Variable</th>
<th>( F^* )</th>
<th>( F^*_h )</th>
<th>( T^*_h )</th>
<th>( T^{**} )</th>
<th>( T^{**}_h )</th>
<th>( U )</th>
<th>( A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>( 5 )</td>
<td>( 5 )</td>
<td>( 24.2 )</td>
<td>( 20 )</td>
<td>( 70 )</td>
<td>( 150 )</td>
<td>( 90 )</td>
</tr>
<tr>
<td>Unit</td>
<td>kg/s</td>
<td>kg/s</td>
<td>°C</td>
<td>°C</td>
<td>°C</td>
<td>W/m²°C</td>
<td>m²</td>
</tr>
</tbody>
</table>

We use an I-controller for controlling \( y = T_h \) using \( v \). We cannot have fast control for robustness reasons because we are designing a controller based on a static model and applying it to a dynamic model. Therefore, the closed loop time constant \( \tau_C \) needs to be large. We find \( \tau_C = 100 \) (s) and \( K_I = 0.01 \).

The simulation results for a step increase in disturbance \( d_1 = T^{**}_c \) of 1 °C at time 300 s are shown in Fig. 10 and Fig. 11.

![Fig. 10. Response in \( y = T_h \) for a step increase in disturbance \( d_1 = T^{**}_c \) of 1 °C at time 300 s.](image1)

![Fig. 11. Response in \( u = F_c \) for a step increase in disturbance \( d_1 = T^{**}_c \) of 1 °C at time 300 s.](image2)

5. CONCLUSION

The main contribution is the introduction of the new tuning parameter \( \tau_0 \) that transforms a general nonlinear process into a first order system (Eq. 5) instead of an integrating system as in the classical feedback linearization (Eq. 2). In addition, the proposed method decouples the nonlinear system and gives perfect disturbance rejection. Moreover, it shows good results for two simulations examples, 1) a mixing process with both slow and fast dynamics (Section 3.1), and 2) a heat exchanger with plant-model mismatch (Section 3.2), which makes it appealing to process control applications.

REFERENCES


