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 now 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 the decoupling, and feedforward action from the disturbance $d$. This new procedure can be applied both for static and dynamic process, which is typical in process control.

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

1. INTRODUCTION

Control of nonlinear processes can be classified into three main categories:

(1) Linear controllers designed using a linearized model of the nonlinear system (e.g. PID controllers, linear model predictive control etc.);
(2) Adaptive control (Åström and Wittenmark, 2008)
(3) Nonlinear model predictive control (NMPC) (Rawlings et al., 2017)

The objective of this work falls under category 1), and it 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 list a few of them here: (1) feedback linearization (Isidori, 1989; Khalil, 2015; Nijmeijer and van der Schaft, 1990a) (2) input-output linearization (Henson and Seborg, 1997) (3) disturbance decoupling (Huijberts et al., 1991) (4) input decoupling (Balchen et al., 1988) (Isidori et al., 1981), (Nijmeijer and van der Schaft, 1990b) (5) elementary nonlinear decoupling (Balchen, 1998)

Feedback and input-output 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), Khalil (2015), Nijmeijer and van der Schaft (1990a)).

A comprehensive overview and analysis of feedback linearization for process control is presented in the work of (Henson and Seborg, 1991). A typical application presented in the literature for process control is pH control (where the nonlinearity is introduced by the measurement) (Henson and Seborg, 1997). However, to the best of the authors knowledge this method has not been practically implemented for chemical processes. We will explain briefly why this is not used for chemical processes.

Feedback linearization works by transforming the nonlinear system given by Eq.1 into a new system described by a chain of integrators, therefore linear and controllable.

$$\dot{x} = f(x) + g(x)u \quad (1a)$$
$$y = h(x) \quad (1b)$$

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 the states to the output ($y$).
The method consists of two steps (Isidori, 1989):

**Step 1:** a change of coordinates using directional Lie derivatives, defined locally around some point \( x^0 \). This introduces positive feedback which destabilizes the system (i.e. transforms it into an integrating system).

**Step 2:** a state feedback, defined locally around the same point \( x^0 \).

Note that the order of the two steps does not matter, meaning we can first apply a state feedback law, and second change the coordinates to obtain the same linear system. However, an integrating process is difficult to control, and a chain of integrators is almost impossible without measuring all the states. This means that feedback linearization introduces additional and unnecessary control limitations to the system.

In most cases, this method linearizes the input-states behaviour. The full input-output behaviour is linearizable only if the states \( x \) are equal to the outputs \( y \). On the other hand, input-output linearization, transforms only a part of the system into a linear one, while the rest remains nonlinear (Henson and Seborg, 1997). The main limitations of these linearization methods are:

- require a detailed model to be inverted, which may not be robust
- give a chain of integrators which is difficult to control
- require full-state feedback
- only for minimum phase systems
- do not apply for static systems

These limitations may not be cumbersome 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 \( x \) (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, \( y^* \) is the setpoint vector. 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 present in the next sections the structure of each block in Fig. 1.

### 2.1 Assumptions

- as many outputs as inputs \( n_y = n_u \)
- all disturbances can be measured

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.

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 positive 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 these issues, we rewrite the model Eq. 2 slightly and introduce the new tuning parameter $\tau_0$. 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 $d$):

$$\tau_0 \frac{dy_1}{dt} + y_1 = f_1(u_1, u_2, d, y_1, y_2)$$  \hfill (5a)

$$\tau_2 \frac{dy_2}{dt} + y_2 = f_2(u_1, u_2, d, y_1, y_2)$$  \hfill (5b)

Comparing Eq. 5 with Eq. 2, we see that $f_1 = f_1' \ast \tau_0 + y_1$. Now we introduce two new transformed input variables (input functions)

$$v_1 = f_1(u_1, u_2, d, y_1, y_2)$$  \hfill (6a)

$$v_2 = f_2(u_1, u_2, d, y_1, y_2)$$  \hfill (6b)

where we assume that $d$ is measured. We then have two decoupled linear systems, both first-order and independent of disturbances.

$$\tau_0 \frac{dv_1}{dt} + v_1 = f_1'(v_1)$$  \hfill (7a)

$$\tau_0 \frac{dv_2}{dt} + v_2 = f_2'(v_2)$$  \hfill (7b)

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.

$$u(t) = K_c \left( c(t) + \frac{1}{\tau_I} \int_0^t c(t) dt \right)$$  \hfill (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 may use the tuning rules given by Eq. 9 to tune a PI-controller.

$$K_c = \frac{1}{k \tau_C + \theta}$$  \hfill (9a)

$$\tau_I = \min(\tau, 4(\tau_C + \theta))$$  \hfill (9b)

where, $k$ is the process gain $k$, $\tau$ is 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 I-controller given by Eq. 10.

$$K_c = 0$$  \hfill (10a)

$$\tau_I = \tau = 0$$  \hfill (10b)

$$K_I = \lim_{\tau \to 0} \frac{K_C}{\tau_I} = \frac{1}{k \tau_C + \theta}$$  \hfill (10c)

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. 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 $u$.

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 $y_1 = 1$. 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, $dy_1/dt = v_1$ 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 $dy_1/dt = v$. 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. 11. 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).

3. SIMULATION CASE STUDIES

We apply the proposed method from Section 2 to two simulation examples. We illustrate the effect of unmeasured delay of the disturbance signal (case 1) and plant-model mismatch (case 2). The two cases are:

1. Control of flow and temperature in a mixing process with both slow temperature dynamics and fast mass dynamics including measurement delays.
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. 2 shows the mixing process with two inflows and one outflow that we are analyzing.

Fig. 3 shows the block diagram of the proposed inputs transformation for the mixing process. The original inputs of the process are the two inlet flows:

$u_1 = F_1 \ [kg/s]$  
$u_2 = F_2 \ [kg/s]$  

The process outputs are the outlet flow $F$ and temperature $T$.
The main disturbances that we consider are the temperature of the two inlet flows:

\[ \begin{align*}
  y_1 &= F \quad [\text{kg/s}] \\
  y_2 &= T \quad [\text{°C}]
\end{align*} \]

The main disturbances that we consider are the temperature of the two inlet flows:

\[ \begin{align*}
  d_1 &= T_1 \quad [\text{°C}] \\
  d_2 &= T_2 \quad [\text{°C}]
\end{align*} \]

Following the procedure from Section 2, we find the transform inputs \( v_1 \) and \( v_2 \) for the mixing process with the objective of decoupling and perfect disturbance rejection.

Assuming constant \( m \) holdup, and fast mixing, the mass balance (static) is given by Eq. 12.

\[ \begin{align*}
  F &= F_1 + F_2
\end{align*} \]  

Assuming constant and equal heat capacity \( c_p \), and after substituting the mass balance (Eq. 12), we can rearrange the dynamic energy balance as given by Eq. 13 (similar with Eq. 5).

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

where \( \tau_r = \frac{m}{F_1 + F_2} \) is the residence time [s], and \( \tau_0 \) is the new tuning parameter. The question now is how should we choose the tuning parameter \( \tau_0 \)? One way of selecting it is to minimize the implicit feedback from the output \( y \) to the new input \( v \). Note that if \( \tau_0 = \tau_r \), then \( v_2 \) is independent of \( y_2 \).

We define the transformed inputs \( v_1 \) and \( v_2 \) as the right hand side of Eq. 12 and Eq. 13 as shown in Eq. 14.

\[ \begin{align*}
  v_1 &= F_1 + F_2 \quad (14a) \\
  v_2 &= \tau_0 m (F_1 T_1 + F_2 T_2) + \left(1 - \frac{\tau_0}{\tau_r}\right) T \quad (14b)
\end{align*} \]

With the new transformed input, \( v_1 \) and \( v_2 \), the new system is given by Eq. 15, which represents two decoupled processes with no effect from disturbances.

\[ \begin{align*}
  v_1 &= \frac{dy_1}{dt} \\
  v_2 &= \frac{dy_2}{dt}
\end{align*} \]  

In transfer function matrix form, from input \( v = [v_1 \quad v_2]^T \) and disturbances \( d = [d_1 \quad d_2]^T \) to outputs \( y = [F \quad T]^T \), the system can be rewritten as given in Eq. 16.

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

with

\[ G(s) = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{\tau_0 s + 1} \end{bmatrix}, \quad G_d(s) = 0 \]

**Calculation block.** Algebraic solver The calculation block solves Eq. 14 for \( v_1 \) and \( v_2 \) given inputs \( v_1 \) and \( v_2 \), outputs \( y_1 = F \) and \( y_2 = T \) and disturbances \( d \). In this case, this is a linear system, Eq. 17.

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

**Simulation results** Table 1 shows the nominal operating conditions (marked with *+) for the mixing process. Note that at nominally 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>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_1^* )</td>
<td>5</td>
<td>kg/s</td>
</tr>
<tr>
<td>( F_2^* )</td>
<td>5</td>
<td>kg/s</td>
</tr>
<tr>
<td>( F^* )</td>
<td>10</td>
<td>kg/s</td>
</tr>
<tr>
<td>( T_1^* )</td>
<td>20</td>
<td>°C</td>
</tr>
<tr>
<td>( T_2^* )</td>
<td>50</td>
<td>°C</td>
</tr>
<tr>
<td>( T^* )</td>
<td>35</td>
<td>°C</td>
</tr>
<tr>
<td>( m )</td>
<td>100</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>
<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 \leftrightarrow y_1 )</td>
<td>1</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>( v_2 \leftrightarrow y_2 )</td>
<td>100</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

The simulation results 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 that we want to test the robustness of our method by using a static model (\( \epsilon \)-NTU method) to derive the new input transformation, and then apply it to a dynamic cells heat exchanger model. Fig. 8 shows the countercurrent heat exchanger that we are analyzing, with one original input (the cold side flow:
The original input of the process is the cold side flow: $u = F_c$ [kg/s] and one output (hot side outlet temperature ($y = T_h$)).

The main disturbances that we consider are the hot side flow and the inlet temperatures of the two flows:

- $d_1 = T_{h, in}^{in}$ [°C]
- $d_2 = T_{c, in}^{in}$ [°C]
- $d_3 = F_h$ [kg/s]

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

$$T_h = (1 - \epsilon_h)T_{h, in}^{in} + \epsilon_h T_{c, in}^{in}$$

$$T_c = \epsilon_c T_{c, in}^{in} + (1 - \epsilon_c) F_c$$

$$\epsilon_c = \frac{1 - \exp(-NTU(C - 1))}{C - \exp(-NTU(C - 1))}$$

$$\epsilon_h = \frac{F_h C_p_h}{\epsilon_c}$$

$$NTU = \frac{U A}{F_c C_p_c}$$

We define the transformed inputs $v$ as the right hand side of Eq. 18a resulting in Eq. 19.

$$v = T_h$$

With the new transformed input, $v$, the new system is given by Eq. 20, with no effect from disturbances.

$$y = v$$

In transfer function matrix form, from input $v$ and disturbances $d$ to output $y$, the system can be rewritten as given in Eq. 21.

$$y(s) = G(s)v(s) + G_d d(s)$$

with

$$G(s) = 1, \quad G_d(s) = 0$$
Calculation block. Algebraic solver The calculation block numerically solves Eq. 18 for \( u \) given inputs \( v \) and disturbances \( d \). Note that we are using \( v = y = T_h \) in the calculation block.

Dynamic cells heat exchanger model The process is represented by the dynamic lumped model given in Eq. (22), 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 \( T_{h}^{N+1} = T_{h}^{in} \). Note that with infinite cells, the dynamic model has the same steady-state values as the static model. Wall capacities are neglected in the model.

\[
\frac{dT_c^i}{dt} = \frac{F_c}{\rho c}\left(T_{c}^{i+1} - T_{c}^{i}\right) + \frac{UA}{\rho c V_h^i} (T_h^i - T_c^i) \tag{22a}
\]
\[
\frac{dT_h^i}{dt} = \frac{F_h}{\rho V_h^i} (T_h^{i-1} - T_h^i) + \frac{UA}{\rho V_h^i c_p} (T_h^i - T_c^i) \tag{22b}
\]

\( \forall i \in 1 \ldots N \)

where \( c \) is the cold side, \( h \) is the hot side, \( V \) is 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 \( e-NTU \) (Eq. 18) and cells models (Eq. 22) 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. Nominal operating conditions for the heat exchanger

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_c^* )</td>
<td>5</td>
<td>kg/s</td>
</tr>
<tr>
<td>( F_h^* )</td>
<td>3</td>
<td>kg/s</td>
</tr>
<tr>
<td>( T_h^{stat, in}^* )</td>
<td>24.2</td>
<td>°C</td>
</tr>
<tr>
<td>( T_h^{dyn, in}^* )</td>
<td>24.2</td>
<td>°C</td>
</tr>
<tr>
<td>( T_h^{in}^* )</td>
<td>20</td>
<td>°C</td>
</tr>
<tr>
<td>( T_c^{in}^* )</td>
<td>70</td>
<td>°C</td>
</tr>
</tbody>
</table>

The controller tuning is given in Table 4. Note that we use an I-controller for controlling \( y = T_h \) using \( v \). We cannot have fast control for robusteness 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.

Table 4. Controller tunings for the heat exchanger

<table>
<thead>
<tr>
<th>MV=CV</th>
<th>( \tau_C ) (s)</th>
<th>( K_I )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v \leftrightarrow y )</td>
<td>100</td>
<td>0.01</td>
</tr>
</tbody>
</table>

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

4. DISCUSSION

For the MIMO mixing example (Section 3.1), the simulation results in Fig. 7, Fig. 5 and Fig. 6 show a decoupled process for which that the controller and the calculation block are able to compensate for the measurement delay.

For the heat exchanger example (Section 3.2), the simulation does not start at the nominal values in Fig. 11, and Fig. 10 because the steady-state outputs values for the static and dynamic model are different.

4.1 Output transformation

For some chemical processes, the nonlinearities arise from state-measurement relationship, e.g. pH, or density measurement. A second reason for introduction of an output transformation is that the structure (i.e. algebraic equations) of the input calculation block is fixed from the start and this may not be robust enough.

For such systems, we may want to introduce new outputs transformation that linearizes the input (\( v \))- internal state (\( x \)) behaviour. The effect is that we will control a subset of the internal state \( x \), and not the outputs \( y \). The principle of both input and output transformation is illustrated in Fig. 12.
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). The proposed method 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).

REFERENCES


