A normal form for two-input forward-flat nonlinear discrete-time systems

Johannes Diwold, Bernd Kolar and Markus Schöberl

Institute of Automatic Control and Control Systems Technology, Johannes Kepler University, Linz, Austria

ABSTRACT

We show that every forward-flat nonlinear discrete-time system with two inputs can be transformed into a structurally flat normal form by state- and input transformations. This normal form has a triangular structure and allows to read off the flat output, as well as a systematic construction of the parameterisation of all system variables by the flat output and its forward-shifts. For flat continuous-time systems, no comparable normal form exists.

1. Introduction

In the 1990s, the concept of flatness has been introduced by Fliess, Lévine, Martin and Rouchon for nonlinear continuous-time systems (see e.g. Fliess et al., 1992, 1995, 1999). Flat continuous-time systems have the characteristic feature that all system variables can be parametrised by a flat output and its time derivatives. They form an extension of the class of static feedback linearisable systems and can be linearised by endogenous dynamic feedback. Their popularity stems from the fact that a lot of physical systems possess the property of flatness and that the knowledge of a flat output allows an elegant solution to motion planning problems and a systematic design of tracking controllers.

For nonlinear discrete-time systems, flatness can be defined analogously to the continuous-time case. The main difference is that time derivatives have to be replaced by forward-shifts. Like in the continuous-time case, flat systems form an extension of static feedback linearisable systems. The problem of static feedback linearisation for discrete-time systems is already solved, see Grizzle (1986), Jakubczyk (1987) and Aranda-Bricaire et al. (1996). An important difference to the continuous-time case is the existence of discrete-time systems that can be linearised by dynamic feedbacks which are not contained within the class of endogenous dynamic feedbacks as proposed in Aranda-Bricaire and Moog (2008). The corresponding linearising output then depends not only on forward-shifts but also on backward-shifts of system variables. A recent definition of flatness proposed in Guillot and Millérioux (2020) includes also backward-shifts of the input variables in the flat output. However, within this contribution, similar to Kaldmäe and Kotta (2013), Kolar, Kaldmäe, et al. (2016) we restrict ourselves to forward-shifts in the flat output and therefore use the term forward-flatness. The property of forward-flatness is equivalent to linearisability by an endogenous dynamic feedback as proposed in Aranda-Bricaire and Moog (2008).

In general, the analysis of flat systems can be divided into two separate tasks. First, we are interested in checking whether a system is flat or not in order to clarify if flatness based control strategies can be applied in principle. In Grizzle (1986) and Nijmeijer and van der Schaft (1990), an efficient test for static feedback linearisable systems, which is based on the computation of certain distributions, can be found. As we have shown in Kolar, Diwold, et al. (2019), this test can be generalised to the class of forward-flat systems. The test is based on the results of Kolar, Schöberl, et al. (2019). It should be noted that for continuous-time flat systems no comparable test is
available so far. Second, in order to use the flatness property for control strategies, the knowledge of a flat output as well as the corresponding parameterisation of all system variables is necessary. For this purpose, the use of structurally flat normal forms (see e.g. Kolar, Schöberl, et al., 2016; Schöberl, 2014; Schöberl & Schlacher, 2014) has turned out to be helpful. Structurally flat normal forms allow to read off the flat output, as well as a systematic construction of the parameterisation of all system variables. The most famous example for such a normal form is the Brunovsky normal form. However, a transformation to Brunovsky normal form is possible if and only if the system is static feedback linearisable. In Kolar, Diwold, et al. (2019), we have shown that every forward-flat system can be transformed into a structurally flat implicit normal form. The main feature of this normal form is that the equations depend on the system variables in a triangular manner. The reason for the implicit character of this normal form is that the required coordinate transformations are possibly more general than the usual state- and input transformations. In the present contribution, in order to preserve an explicit state representation we restrict ourselves to state- and input transformations. We introduce a structurally flat explicit system representation, and show that for every forward-flat system with two inputs the existence of a transformation into such a representation is guaranteed. Hence, for two-input forward-flat systems we also use the term explicit triangular normal form. We want to emphasise that for flat continuous-time systems no comparable normal form exists.

The paper is organised as follows: In Section 2 we recapitulate the concept of flatness in terms of forward-shifts and the corresponding test according to Kolar, Diwold, et al. (2019). In Section 3 we discuss certain coordinate transformations, which will be useful later on. In Section 4 we introduce a structurally flat explicit triangular form. Then, we prove that every two-input forward-flat discrete-time system can be transformed into such a representation by successive state- and input transformations. Finally, in Section 5, we illustrate our results by an example.

2. Forward-flatness of discrete-time systems

Throughout this contribution we consider discrete-time nonlinear systems in explicit state representation of the form
\[ x^{i+} = f^i(x, u), \quad i = 1, \ldots, n \]
with \( \text{dim}(x) = n, \text{dim}(u) = m \) and smooth functions \( f^i(x, u) \). Geometrically, the system (1) can be interpreted as a map \( f \) from a manifold \( \mathcal{X} \times \mathcal{U} \) with coordinates \( (x, u) \) to a manifold \( \mathcal{X}^+ \) with coordinates \( x^+ \). Furthermore, we assume that the system meets \( \text{rank}(\partial_{(x,u)}f) = n \), which is a necessary condition for accessibility and consequently also for flatness. Apart from this, we assume that the system possesses no redundant inputs, i.e. \( \text{rank}(\partial_uf) = m \), and define forward-flatness according to Kolar, Diwold, et al. (2019).

**Definition 2.1:** A system (1) is said to be forward-flat around an equilibrium \((x_0, u_0)\), if there exists an \(m\)-tuple of functions
\[ y^j = \varphi^j(x, u, u_{[1]}, u_{[2]}, \ldots, u_{[q]}), \quad j = 1, \ldots, m, \]
where \( u_{[\alpha]} \) denotes the \( \alpha \)-th forward-shift of \( u \), such that the \( n + m \) coordinate functions \( x \) and \( u \) can be expressed locally by \( y \) and forward-shifts of \( y \) up to some finite order, i.e.
\[ x^i = F^i_x(y, y_{[1]}, y_{[2]}, \ldots, y_{[R-1]}), \quad i = 1, \ldots, n \\
 u^j = F^j_u(y, y_{[1]}, y_{[2]}, \ldots, y_{[R]}), \quad j = 1, \ldots, m. \]
The \(m\)-tuple (2) is called a forward-flat output.

The test for forward-flatness, as stated in Kolar, Diwold, et al. (2019), is based on the construction of sequences of nested distributions on \( \mathcal{X} \times \mathcal{U} \) and \( \mathcal{X}^+ \). The construction makes use of the system equations (1) and the map
\[ \pi : \mathcal{X} \times \mathcal{U} \to \mathcal{X}^+ \]
defined by
\[ x^{i+} = x^i, \quad i = 1, \ldots, n, \]
as well as their tangent maps \( f_x \) and \( \pi_x \). In the following, we call a distribution \( D \) on \( \mathcal{X} \times \mathcal{U} \) projectable (with respect to the map \( f \)), if its pushforward \( f_x(D) \) induces a well-defined distribution on \( \mathcal{X}^+ \). Whether a distribution is projectable or not can be checked by the use of adapted coordinates on \( \mathcal{X} \times \mathcal{U} \). Further details can be found in Kolar, Diwold, et al. (2019).
Algorithm 2.1: Step k = 0: Define the distribution
\[ \Delta_0 = 0 \]
on \mathcal{X}^+ and
\[ E_0 = \pi^{-1}_u(\Delta_0) = \text{span}\{\partial_u\} \]
on \mathcal{X} \times \mathcal{U}. Then compute the largest subdistribution
\[ D_0 \subset E_0 \]
which is projectable with respect to the map f of (1). The distribution \( D_0 \) is involutive\(^3\) and its pushforward
\[ \Delta_1 = f_*(D_0) \]
is a well-defined involutive distribution on \( \mathcal{X}^+ \).

Step k ≥ 1: Compute
\[ E_k = \pi^{-1}_u(\Delta_k) \] and the largest subdistribution
\[ D_k \subset E_k \]
which is projectable with respect to the map f of (1). The distribution \( D_k \) is involutive and its pushforward
\[ \Delta_{k+1} = f_*(D_k) \]
is a well-defined involutive distribution on \( \mathcal{X}^+ \).

Stop if for some \( k = \bar{k} \),
\[ \dim(\Delta_{\bar{k}+1}) = \dim(\Delta_{\bar{k}}). \]

The procedure according to Algorithm 2.1 yields a unique nested sequence of projectable and involutive distributions
\[ D_0 \subset D_1 \subset \cdots \subset D_{\bar{k}-1} \]
on \mathcal{X} \times \mathcal{U} and a unique nested sequence of involutive distributions
\[ \Delta_1 \subset \Delta_2 \subset \cdots \subset \Delta_{\bar{k}} \]
on \mathcal{X}^+ so that
\[ f_*(D_k) = \Delta_{k+1}, \quad k = 0, \ldots, \bar{k} - 1. \]

Whether a system is forward-flat or not can now be checked by use of the following theorem.

Theorem 2.1: A system (1) with \( \text{rank}(\partial_u f) = m \) is forward-flat if and only if \( \dim(\Delta_{\bar{k}}) = n \).

For the proof we refer to Kolar, Diwold, et al. (2019). The test for forward-flatness contains the test for static feedback linearisability (see Nijmeijer & van der Schaft, 1990) as a special case. The only difference is that the distributions (5), according to Algorithm 2.1, are defined as the largest projectable subdistributions \( D_k \subset E_k \), while in the static feedback linearisable case these distributions coincide, i.e. \( D_k = E_k \).

Theorem 2.2: A system (1) with \( \text{rank}(\partial_u f) = m \) is static feedback linearisable if and only if \( D_k = E_k, k \geq 0 \) and \( \dim(\Delta_k) = n \).

Since all distributions \( E_k \) must be completely projectable, the test for static feedback linearisability is more restrictive. For a system that meets Theorem 2.1, a single step where \( D_k \neq E_k \) can be interpreted as a defect in the test of static feedback linearisability, as we will demonstrate by the following example.

Example 2.1: For the system
\[ x^{5,+} = x^4 + x^1 + x^5 \]
\[ x^{4,+} = x^1(x^4 + 1) + x^3 \]
\[ x^{3,+} = x^1 + x^2 \]
\[ x^{2,+} = u^1 \]
\[ x^{1,+} = u^2 \]
we obtain the sequence of distributions
\[ D_0 = \text{span}\{\partial_{x^1}, \partial_{x^2}\} = E_0 \]
\[ D_1 = \text{span}\{\partial_{x^1}, \partial_{x^2}, \partial_{x^3}\} \subset E_1 \]
\[ D_2 = \text{span}\{\partial_{x^1}, \partial_{x^2}, \partial_{x^3}, \partial_{x^4}, \partial_{x^5}\} = E_2 \]
on \mathcal{X} \times \mathcal{U} and
\[ \Delta_1 = \text{span}\{\partial_{x^{1,+}}, \partial_{x^{2,+}}\} \]
\[ \Delta_2 = \text{span}\{\partial_{x^{1,+}}, \partial_{x^{2,+}}, \partial_{x^{3,+}}\} \]
\[ \Delta_3 = \text{span}\{\partial_{x^{1,+}}, \partial_{x^{2,+}}, \partial_{x^{3,+}}, \partial_{x^{4,+}}, \partial_{x^{5,+}}\} \]
on \mathcal{X}^+. The computations can be found in some detail in the appendix. Despite the fact that \( E_1 \) is not completely projectable, i.e. \( D_1 \neq E_1 \), the distribution \( \Delta_3 \) meets \( \dim(\Delta_3) = n \) and the system possesses the weaker property of forward-flatness instead of static feedback linearisability. A forward-flat output is given by \( y = (x^4, x^5) \).
Before we state our main results in Section 4, we discuss certain state- and input transformations which will be useful for straightening out the sequences of distributions. In order to preserve an explicit system representation like (1), within this contribution we restrict ourselves to state- and input transformations

\begin{equation}
\dot{x}^i = \Phi^i(x), \quad i = 1, \ldots, n
\end{equation}

\begin{equation}
\dot{u}^j = \Phi^j_u(x, u), \quad j = 1, \ldots, m,
\end{equation}

where both \( x \) and \( x^+ \) are transformed equally. The transformed system is given by

\begin{equation}
\dot{x}^{i+} = \Phi^i_u(x^+), \quad i = 1, \ldots, n
\end{equation}

\begin{equation}
\dot{u}(\tilde{x}, \tilde{u}) = f(\tilde{x}, \tilde{u})
\end{equation}

where \( \Phi^{-1}(\tilde{x}, \tilde{u}) \) denotes the inverse of (9). Like in the static feedback linearisable case, the first step in achieving a triangular representation is to straighten out the sequence (6). Since (6) is a nested sequence of involutive distributions on \( x^+ \), by an extension of the Frobenius theorem there exists a state transformation

\begin{equation}
(\tilde{x}_1, \ldots, \tilde{x}_k) = \Phi(x),
\end{equation}

with \( \dim(\tilde{x}_k) = \dim(\Delta_k) - \dim(\Delta_{k-1}) \), which straightens out the distributions

\begin{equation}
\Delta_1 = \text{span}\{\partial_{x^+}^1\}
\end{equation}

\begin{equation}
\Delta_2 = \text{span}\{\partial_{x^+}^1, \partial_{x^+}^2\}
\end{equation}

\vdots

\begin{equation}
\Delta_k = \text{span}\{\partial_{x^+}^1, \partial_{x^+}^2, \ldots, \partial_{x^+}^k\}
\end{equation}

simultaneously. The system in new coordinates reads as

\begin{equation}
\tilde{x}_k^+ = f_k(\tilde{x}, u), \quad k = 1, \ldots, \bar{k}
\end{equation}

and meets

\begin{equation}
f_u(D_{k-1}) = \text{span}\{\partial_{x^+}^1, \partial_{x^+}^2, \ldots, \partial_{x^+}^k\}
\end{equation}

for \( k = 1, \ldots, \bar{k} \). Additionally, from the definition of \( E_k \) according to (4) and (12), it follows automatically that \( E_k \) is also straightened out and reads as

\begin{equation}
E_k = \pi^{-1}_a(\Delta_k) = \text{span}\{\partial_{x^+}^1, \ldots, \partial_{x^+}^k, \partial_{u}\}
\end{equation}

for \( k = 0, \ldots, \bar{k} - 1 \). The \( D \)-sequence, in contrast, is only straightened out automatically if the system possesses the stronger property of static feedback linearisability, since then \( D_{\bar{k}} = E_{\bar{k}} \). Thus, for finding coordinates that straighten out both sequences of distributions (5) and (6), the transformation (11) alone is not
sufficient. Therefore, we need an additional transformation that straightens out the \( D \)-sequence while the \( \Delta \)-sequence remains straightened out. In the following, we introduce transformations that meet the latter condition.

**Lemma 3.1:** State- and input transformations of the form

\[
\hat{x}_k = \Phi_{\hat{x},k}(\bar{x}_k, \ldots, \bar{x}_k), \quad k = 1, \ldots, \bar{k}
\]

\[
\hat{u} = \Phi_u(\bar{x}, u)
\]

preserve the structure (12) of the sequence of distributions (6), i.e.

\[
\Delta_1 = \text{span}\{\partial_{\xi^1}\}
\]

\[
\Delta_2 = \text{span}\{\partial_{\xi^1}, \partial_{\xi^2}\}
\]

\[
\vdots
\]

\[
\Delta_k = \text{span}\{\partial_{\xi^1}, \partial_{\xi^2}, \ldots, \partial_{\xi^k}\}.
\]

The proof follows from the triangular structure of the state transformation of (16). The input transformation does not affect the \( \Delta \)-sequence.

For systems with two inputs the distributions of the \( D \)-sequence have a very special structure. Since we will deal with two-input systems in Section 4.2, we state the following important lemma.

**Lemma 3.2:** Consider an \( n \)-dimensional manifold \( Z \) with coordinates \( \xi = (\xi^1, \ldots, \xi^n) \) and an involutive distribution

\[
D = \text{span}\{\partial_{\xi^1}, \ldots, \partial_{\xi^{k-1}}, \partial_{\xi^j} + \alpha(\xi)\partial_{\xi^i}\}
\]

for some \( i, j \geq k \). There exists a transformation

\[
\hat{\xi}^j = \Phi^j(\xi^k, \ldots, \xi^n)
\]

of the coordinate \( \xi^j \) such that

\[
D = \text{span}\{\partial_{\xi^1}, \ldots, \partial_{\xi^{k-1}}, \partial_{\xi^j}\}.
\]

The proof can be found in the appendix. For two-input systems we will encounter distributions \( D_k \) of the form (17) on the manifold \( \mathcal{X} \times \mathcal{U} \), and straighten them out by state- or input transformations of the type (18). These transformations will also exhibit the structure-preserving form (16) with respect to the \( \Delta \)-sequence.

**4. Explicit triangular form**

In Nijmeijer and van der Schaft (1990) it is shown how a static feedback linearisable system can be transformed into Brunovsky normal form. In the first step, a state transformation is performed that straightens out the sequences of distributions simultaneously. This yields an explicit triangular system representation which can be interpreted as a composition of smaller subsystems. With respect to the inputs of these subsystems, there may occur redundancies. Following Nijmeijer and van der Schaft (1990), further state- and input transformations are successively performed in order to obtain the Brunovsky normal form. The above mentioned redundancies appear if the chains of the Brunovsky normal form have different lengths.

For forward-flat systems that are not static feedback linearisable, a transformation to Brunovsky normal form is not possible. Thus, we introduce a more general structurally flat explicit triangular form that can be obtained by straightening out the \( D \)- and \( \Delta \)-sequences by suitable state- and input transformations. For two-input systems, we prove that such a transformation which straightens out both sequences of distributions simultaneously always exists. Subsequently, similar to the static feedback linearisable case, redundant inputs of the subsystems can be eliminated by further structure-preserving state- and input transformations. For the resulting system representation we use the term explicit triangular normal form. It allows to read off a forward-flat output and the corresponding parameterising map, according to Definition 2.1, in a systematic way.

**4.1. Explicit triangular form for multi-input systems**

In the following we present an explicit triangular representation for forward-flat systems. Note, we do not refer to it as a normal form, since for forward-flat systems with an arbitrary number of inputs the existence of such coordinates is not guaranteed in general.

**Theorem 4.1:** Assume there exists a state- and input transformation

\[
\hat{x} = \Phi_x(x)
\]

\[
\hat{u} = \Phi_u(x, u)
\]
that straightens out the sequences (5) and (6) simultaneously, i.e.

$$\Delta_j = \text{span}\{\partial_{x_1^j}, \ldots, \partial_{x_k^j}\}, \quad j = 1, \ldots, \tilde{k}$$  \hspace{1cm} (21)

with $$\dim(\Delta_j) = \dim(\Delta_j) - \dim(\Delta_{j-1})$$ and

$$D_k = \text{span}\{\partial_{x_0}, \ldots, \partial_{x_k}\}, \quad k = 0, \ldots, \tilde{k} - 1.$$  \hspace{1cm} (22)

Here $$z_k$$ denotes a selection of components of $$(\hat{u}, \hat{x}_1, \ldots, \hat{x}_k)$$ with $$\dim(z_0) = \dim(D_0)$$ and $$\dim(z_k) = \dim(D_k) - \dim(D_{k-1})$$ for $$k = 1, \ldots, \tilde{k} - 1$$. In such coordinates, the system (1) has the triangular form

$$\begin{align*}
\hat{x}_k^+ &= f_k(\hat{x}_k, z_{k-1}) \\
\hat{x}_{k-1}^+ &= f_{k-1}(\hat{x}_k, z_{k-1}, z_{k-2}) \\
&\vdots \\
\hat{x}_1^+ &= f_1(\hat{x}_k, z_{k-1}, \ldots, z_0)
\end{align*}$$  \hspace{1cm} (23)

with

$$\hat{x}_k \subset (z_k, \ldots, z_{k-1}), \quad k = 1, \ldots, \tilde{k} - 1$$  \hspace{1cm} (24)

and meets

$$\text{rank}(\partial_{z_{j-1} \cdots}) = \dim(\hat{x}_j), \quad j = 1, \ldots, \tilde{k}.$$  \hspace{1cm} (25)

The proof can be found in the appendix. Hereinafter, we state the intention of defining $$z_k$$. Since the $$\Delta$$-sequence is straightened out, likewise is $$E_k = \text{span}\{\partial_{x_1^k}, \ldots, \partial_{x_k^k}\}$$. Furthermore, by assumption the $$D$$-sequence is also straightened out, and due to $$D_k \subset E_k$$ it follows that $$D_k$$ might not contain all components of $$\partial_{x_1^k}, \ldots, \partial_{x_k^k}$$. Therefore, we introduce the variable $$z_k$$, which acts as placeholder and describes states and/or inputs of $$(\hat{u}, \hat{x}_1, \ldots, \hat{x}_k)$$ so that $$D_k$$ reads as (22). It is important to mention that the variables $$z_0, \ldots, z_{k-1}$$ contain all inputs and states except $$\hat{x}_k$$ (see the proof in the appendix). Note, the system (23) is still in an explicit state representation (1), as we can replace $$z_k$$ by the corresponding states and inputs. We clarify the definition of $$z_k$$ using the system of Example 2.1.

**Example 4.1**: Consider the system (8) of Example 2.1. In this example, both the $$D$$- and the $$\Delta$$-sequence are already straightened out. Thus, the coordinate transformation of Theorem 4.1 is just a renaming

$$\begin{align*} 
\hat{x}_1^1 &= x^5 \\
\hat{x}_1^2 &= x^3 \\
\hat{x}_1^1 &= x^2 \\
\hat{u}_1 &= u^1 \\
\hat{x}_1^2 &= x^4 \\
\hat{x}_1^2 &= x^1 \\
\hat{u}_2 &= u^2. 
\end{align*}$$

According to Theorem 4.1 we define

$$\begin{align*}
z_0 &= (\hat{u}_1^1, \hat{u}_1^2), \\
z_1 &= (\hat{x}_1^1), \\
z_2 &= (\hat{x}_1^2, \hat{x}_1^2)
\end{align*}$$  \hspace{1cm} (26)

such that the $$D$$-sequence of distributions reads as

$$\begin{align*}
D_0 &= \text{span}\{\partial_{z_0}\} \\
D_1 &= \text{span}\{\partial_{z_0}, \partial_{z_1}\} \\
D_2 &= \text{span}\{\partial_{z_0}, \partial_{z_1}, \partial_{z_2}\}
\end{align*}$$

and the system follows as

$$\begin{align*}
x_3^{1+} &= x_3^2 + x_2^1 + x_1^1 \\
x_3^{2+} &= x_3^2(x_3^2 + 1) + x_1^1 \\
x_2^{1+} &= x_2^1 + x_1^1 \\
x_1^{1+} &= x_1^1 \\
x_1^{2+} &= 0.
\end{align*}$$

The system has the structure of (23), and a forward-flat output is given by $$y = (x_1^1, x_2^2)$$.

It is important to emphasise that for systems with $$m > 2$$ inputs the existence of a state- and input transformation (20) that straightens out both sequences (5) and (6) simultaneously is not guaranteed. Thus, an explicit triangular form (23) does not necessarily exist. However, at least a transformation into an implicit triangular form as discussed in Kolar, Diwold, et al. (2019) is always possible.

### 4.2. Explicit triangular form for two-input systems

In the following we restrict ourselves to forward-flat systems with two inputs and state our main result.

**Theorem 4.2**: A two-input forward-flat system (1) is locally transformable into an explicit triangular representation (23).

According to Theorem 4.1, we must show that there exists a state- and input transformation (20) which straightens out the sequences of distributions (5) and (6) simultaneously. We start with the system
representation (13), where the $\Delta$-sequence has already been straightened out by a suitable state transformation. Next, we want to straighten out the $D$-sequence step by step, starting with $D_0$. For this purpose, we can exploit the fact that for systems with two inputs the dimension of these distributions grows in every step by either one or two. In the first case, the distribution $D_k$ is of the form (17) and can be straightened out by a transformation (18), whereas in the latter case the distribution $D_k$ is already straightened out and no transformation is required.

The following algorithm straightens out the $D$-sequence step by step with state- and input transformations that preserve the structure of the $\Delta$-sequence according to Lemma 3.1. In every step $k$, after performing the transformation the corresponding states or inputs are renamed by $z_k$ and $z_{k,c}$. As mentioned before, $z_k$ is just a selection of states and inputs so that

$$D_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_k}\},$$

whereas $z_{k,c}$ denotes the complementary state or input so that

$$E_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_k}, \partial_{z_{k,c}}\}.$$  

To keep the successive transformations readable, after each step $k$ we return from the hat notation for the transformed variables again to the bar notation. However, for the final system representation after the last step we use the hat notation.

**Algorithm 4.1**: Step $k = 0$, we distinguish between the two cases:

(a) If the entire input distribution is projectable, i.e. $D_0 = E_0$, then there is no need for an input transformation because $E_0$ is already straightened out. We define $z_0 = (u^1, u^2)$ and $z_{0,c}$ is empty.

(b) If $D_0 \neq E_0$, then

$$D_0 = \text{span}\{\alpha(\bar{x}, u)\partial_{u^1} + \partial_{u^2}\},$$

up to a renumbering of the components of $u$. According to Lemma 3.2, there exists an input transformation $\bar{u}^1 = \Phi_{u^1}(\bar{x}, u)$ such that $D_0 = \text{span}\{\partial_{\bar{u}^2}\}$. We define $z_0 = u^2$ and $z_{0,c} = \bar{u}^1$.

Finally, the distributions are given by

$$D_0 = \text{span}\{\partial_{z_0}\},$$

$$E_0 = \text{span}\{\partial_{z_0}, \partial_{z_{0,c}}\}.$$  

Step $k = 1, \ldots, \tilde{k} - 1$, we repeat the procedure with the distribution

$$D_k \subset E_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_{k-1}}, \partial_{z_{k-1,c}}, \partial\tilde{x}_k\}.$$  

It can be shown that the dimensions of $z_{k-1,c}$ and $\tilde{x}_k$ meet

$$\dim(z_{k-1,c}) \leq 1,$$

$$\dim(\tilde{x}_k) \geq 1,$$

$$\dim(z_{k-1,c}) + \dim(\tilde{x}_k) \leq 2.$$  

Thus, in every step we must distinguish three cases:

(a) If the entire distribution $E_k$ is projectable, i.e. $D_k = E_k$, then there is no need for a transformation because $E_k$ is already straightened out. We define $z_k = (\bar{x}_k, z_{k-1,c})$ and $z_{k,c}$ is empty.

(b) If $D_k \neq E_k$ and $\dim(\tilde{x}_k) = 2$, then $z_{k-1,c}$ is empty and

$$D_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_{k-1}}, \alpha(z, \bar{x}_k)\partial_{\bar{x}_1} + \partial_{\bar{x}_2}\},$$

up to a renumbering of the components of $\bar{x}_k$.

According to Lemma 3.2, there exists a state transformation

$$\hat{\lambda}_k^1 = \Phi_{\hat{x}_1}(\tilde{x}_k, \ldots, \tilde{x}_k)$$

such that $D_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_{k-1}}, \partial_{\bar{x}_1}\}$. We define $z_k = \hat{x}_1^2$ and $z_{k,c} = \hat{x}_k^1$.

(c) If $D_k \neq E_k$ and $\dim(\tilde{x}_k) = 1$, then necessarily also $\dim(z_{k-1,c}) = 1$. Otherwise, we would have $D_k = E_k$ and case (a) would apply. Thus,

$$D_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_{k-1}}, \alpha(z, z_{k-1,c}, \bar{x})\partial_{z_{k-1,c}} + \partial_{\tilde{x}_k}\}.$$  

According to Lemma 3.2, there exists a transformation

$$\hat{z}_{k-1,c} = \Phi_{z_{k-1,c}}(\tilde{x}_k, \ldots, \tilde{x}_k, z_{k-1,c})$$

such that $D_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_{k-1}}, \partial_{\tilde{x}_k}\}$. Since $z_{k-1,c}$ could represent both an input or state variable of the system, the transformation is either an input- or a state transformation. We define $z_k = \bar{x}_k$ and $z_{k,c} = \hat{z}_{k-1,c}$.
Finally, the distributions are given by
\[ D_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_k}\} \]
\[ E_k = \text{span}\{\partial_{z_0}, \ldots, \partial_{z_k}, \partial_{z_{k'}}\} \]

Within the algorithm, only state transformations (27) and input- or state transformations (28) are performed. Since both are of the structure-preserving form (16), the resulting entire transformation law is also of the form (16) and the \( \Delta \)-sequence remains straightened out. Thus, after the last step of the algorithm, both sequences of distributions are straightened out according to (21) and (22) in Theorem 4.1.

Remark 4.1: In the last step \( k = \tilde{k} - 1 \), according to Lemma 2.1 the distribution \( E_{\tilde{k} - 1} \) is completely projectable and case (a) applies. Consequently, the last distribution meets \( D_{\tilde{k} - 1} = E_{\tilde{k} - 1} \) and \( z_{\tilde{k} - 1} \) is empty. Thus, it is ensured that the variables \( z_0, \ldots, z_{\tilde{k} - 1} \) indeed contain all inputs and states except \( \hat{x}_{\tilde{k}} \).

4.3. Explicit triangular normal form for two-input systems

The explicit triangular form (23) consists of the \( \tilde{k} \) subsystems
\[ \hat{x}_{\tilde{k}}^+ = f_k(\hat{x}_{\tilde{k}}, z_{\tilde{k} - 1}) \]
\[ \hat{x}_{\tilde{k} - 1}^+ = f_{\tilde{k} - 1}(\hat{x}_{\tilde{k}}, z_{\tilde{k} - 1}, z_{\tilde{k} - 2}) \]
\[ \vdots \]
\[ \hat{x}_1^+ = f_1(\hat{x}_1, z_{1}) \]
with \( k = 1, \ldots, \tilde{k} \). The parameterisation of the system variables of the system (23) by the forward-flat output can be obtained by determining step by step the parameterisation of the system variables of the subsystems (29), starting with the topmost subsystem
\[ \hat{x}_1^+ = f_1(\hat{x}_1, z_1). \]
(30)

If \( \text{dim}(\hat{x}_k) = \text{dim}(z_{k - 1}) \) for all \( k = 1, \ldots, \tilde{k} \), then due to the rank conditions (25) this is particularly simple, and \( y = \hat{x}_\tilde{k} \) with \( \text{dim}(\hat{x}_\tilde{k}) = m \) is a forward-flat output (see Example 4.1). By applying the implicit function theorem to the topmost subsystem (30) we immediately get the parameterisation of the variables \( z_{\tilde{k} - 1} \). Next, since \( z_{\tilde{k} - 1} \) contains the state variables \( \hat{x}_{\tilde{k} - 1} \) (see (24)), by applying the implicit function theorem to the equations
\[ \hat{x}_{k - 1}^+ = f_{k - 1}(\hat{x}_{k}, z_{k - 1}, z_{k - 2}) \]
we get the parameterisation of the variables \( z_{k - 2} \). Continuing this procedure finally yields the parameterisation of all system variables by the forward-flat output \( y = \hat{x}_k \). However, if \( \text{dim}(\hat{x}_k) < m \), then for at least one \( k \in \{2, \ldots, \tilde{k}\} \) we have \( \text{dim}(\hat{x}_k) < \text{dim}(z_{k - 1}) \), which means that the equations
\[ \hat{x}_k^+ = f_k(\hat{x}_k, z_{k - 1}, \ldots, z_{k - 1}) \]
cannot be solved for all components of \( z_{k - 1} \). In this case, the subsystem (29) has redundant inputs, and in addition to \( \hat{x}_k \) the forward-flat output has further components. The redundant inputs can be eliminated from the subsystem by suitable coordinate transformations of the structure-preserving form (16). The forward-flat output of the complete system (23) consists of \( \hat{x}_k \) and the eliminated redundant inputs of all subsystems.

For systems with two inputs there can occur exactly two cases. If \( \text{dim}(\hat{x}_k) = 2 \), then \( y = \hat{x}_k \) is a forward-flat output and none of the subsystems has redundant inputs. Otherwise, if \( \text{dim}(\hat{x}_k) = 1 \), then there is exactly one \( k \in \{2, \ldots, \tilde{k}\} \) with \( \text{dim}(\hat{x}_k) = 1 < \text{dim}(z_{k - 1}) = 2 \). Thus, the corresponding subsystem (29) has one redundant input. By the use of the regular transformation
\[ z_{k - 1}^2 = f_k(\hat{x}_k, z_{k - 1}, \ldots, z_{k - 1}), \]
(31)
which is either an input- or state transformation but still of type (16), the subsystem reads as
\[ \hat{x}_k^+ = f_k(\hat{x}_k, z_{k - 1}) \]
\[ \hat{x}_{k - 1}^+ = f_{k - 1}(\hat{x}_k, z_{k - 1}, z_{k - 2}) \]
\[ \vdots \]
\[ \hat{x}_1^+ = z_{k - 1} \]
and is independent of \( z_{k - 1}^1 \). The forward-flat output of the complete system is given by \( y = (\hat{x}_k, z_{k - 1}^1) \). After the elimination of the possibly occurring redundant input of the subsystem (29) by the coordinate transformation (31), we refer to the resulting system representation as explicit triangular normal form for two-input systems.
5. Example

Practical examples that are contained within the class of two-input forward-flat discrete-time systems, and thus possess an explicit triangular representation, are for instance the Euler-discretised model of an induction motor (e.g. in Chiasson, 1998; Schöberl, 2014) and the Euler-discretised simplified model of a car (e.g. in Kolar, 2017). However, for illustrational purposes we demonstrate our results with an academic example already discussed in Kolar, Schöberl, et al. (2019). The system reads as

\[
x^{1,+} = \frac{x^2 + x^3 + 3x^4}{u^1 + 2u^2 + 1} \\
x^{2,+} = x^1(x^3 + 1)((u^1 + 2u^2 - 3) + x^4 - 3u^2 \\
x^{3,+} = u^1 + 2u^2 \\
x^{4,+} = x^1(x^3 + 1) + u^2,
\]

and the sequences of distributions (5) and (6) are given by

\[
D_0 = \text{span}\{-2\partial_{u^1} + \partial_{u^2}\} \subseteq E_0 = \text{span}\{\partial_{u^1}, \partial_{u^2}\} \\
D_1 = \text{span}\{\partial_{u^1}, \partial_{u^2}, -3\partial_{x^2} + \partial_{x^4}\} = E_1 \\
D_2 = \text{span}\left\{\partial_{u^1}, \partial_{u^2}, -3\partial_{x^2} + \partial_{x^4}, \frac{x^1}{x^3 + 1}\partial_{x^1} - \partial_{x^3} - \partial_{x^4}, \frac{2x^1}{x^3 + 1}\partial_{x^1} - 2\partial_{x^3} - \partial_{x^4}\right\} = E_2
\]
on \mathcal{X} \times \mathcal{U} and

\[
\Delta_1 = \text{span}\{-3\partial_{x^2} + \partial_{x^4}\} \\
\Delta_2 = \text{span}\left\{-3\partial_{x^2} + \partial_{x^4}, \frac{x^1}{x^3 + 1}\partial_{x^1} - \partial_{x^3} - \partial_{x^4}, \frac{2x^1}{x^3 + 1}\partial_{x^1} - 2\partial_{x^3} - \partial_{x^4}\right\} \\
\Delta_3 = \text{span}\{\partial_{x^1}, \partial_{x^2}, \partial_{x^3}, \partial_{x^4}\}
\]
on \mathcal{X}^+. Following the procedure of Section 4, first we straighten out the \(\Delta\)-sequence by a state transformation of the form (11) with \(\tilde{x}_1 = x_1^1, \tilde{x}_2 = (x_2^1, x_2^2)\) and \(\tilde{x}_3 = x_3^1\). With the transformation

\[
\tilde{x}_1^1 = x^1(x^3 + 1) \quad \tilde{x}_2^1 = x^2 + 3x^4 \quad \tilde{x}_1^1 = x^4, \\
\tilde{x}_2^1 = x^3 \quad \tilde{x}_2 = x^3
\]

the \(\Delta\)-sequence reads as

\[
\Delta_1 = \text{span}\{\partial_{\tilde{x}_1^1}\} \\
\Delta_2 = \text{span}\{\partial_{\tilde{x}_1^1}, \partial_{\tilde{x}_2^1}, \partial_{\tilde{x}_3^1}\} \\
\Delta_3 = \text{span}\{\partial_{\tilde{x}_1^1}, \partial_{\tilde{x}_2^1}, \partial_{\tilde{x}_2^1}, \partial_{\tilde{x}_3^1}\},
\]

and the system in new coordinates is given by

\[
\begin{align*}
\tilde{x}_3^1 &= \tilde{x}_1^1 + \tilde{x}_2^1 \\
\tilde{x}_2^1 &= \tilde{x}_1^1 + \tilde{x}_2^1(u^1 + 2u^2) \\
\tilde{x}_1^1 &= \tilde{x}_3^1 + u^2.
\end{align*}
\]

The \(D\)-sequence in new coordinates reads as

\[
\begin{align*}
D_0 &= \text{span}\{-2\partial_{\tilde{u}^1} + \partial_{\tilde{u}^2}\} \subseteq E_0 \\
D_1 &= \text{span}\{\partial_{\tilde{u}^1}, \partial_{\tilde{u}^2}, \partial_{\tilde{x}_1^1}\} = E_1 \\
D_2 &= \text{span}\{\partial_{\tilde{u}^1}, \partial_{\tilde{u}^2}, \partial_{\tilde{x}_1^1}, \partial_{\tilde{x}_2^1}, \partial_{\tilde{x}_3^1}\} = E_2.
\end{align*}
\]

Next, we use Algorithm 4.1 in order to straighten out the \(D\)-sequence and transform the system into the explicit triangular representation (23). Due to the fact that \(E_0\) is not completely projectable, the case (b) applies and we need to perform an input transformation

\[
\tilde{u}^1 = u^1 + 2u^2
\]

which yields \(D_0 = \text{span}\{\partial_{\tilde{u}^1}\}\). We define \(z_0 = u^2, \quad z_{0,c} = \tilde{u}^1\) and the distribution reads as

\[
D_0 = \text{span}\{\partial_{z_0}\}.
\]

In the second step, due to the fact that \(E_1\) is completely projectable, case (a) applies. We just define \(z_1 = (\tilde{x}_1^1, z_{0,c}) = (\tilde{x}_1^1, \tilde{u}^1)\), and the distribution \(D_1\) reads as

\[
D_1 = \text{span}\{\partial_{z_0}, \partial_{z_1}\}.
\]

Similarly, the last distribution \(E_2\) is also completely projectable (cf. Lemma 2.1) and thus we have again case (a). We just define \(z_2 = (\tilde{x}_2^1, \tilde{x}_2^2)\), and the distribution \(D_2\) reads as

\[
D_2 = \text{span}\{\partial_{z_0}, \partial_{z_1}, \partial_{z_2}\}.
\]

Consequently, with

\[
\begin{align*}
z_0 &= (u^2), \quad z_1 = (\tilde{x}_1^1, \tilde{u}^1), \quad z_2 = (\tilde{x}_2^1, \tilde{x}_2^2)
\end{align*}
\]
the system has the structure of (23) and reads as

\[
\begin{align*}
\dot{x}^1_3 &= z^1_2 + z^2_2 \\
\dot{x}^1_2 &= \dot{x}^1_3 z^2_1 + z^1_1 \\
\dot{x}^1_1 &= z^1_2 \\
\dot{x}^1_0 &= \dot{x}^1_1 + z^1_2.
\end{align*}
\]  

(33)

As mentioned before, the subsystems of (33) may still have redundant inputs. Indeed, because of \( \dim(x^1_3) < \dim(z^2_2) \), the inputs \( z^1_2 \) and \( z^2_2 \) of the topmost subsystem are redundant. This redundancy can be eliminated by the final transformation

\[
\tilde{z}^2_2 = z^1_2 + z^2_2.
\]

(34)

Since \( z^2_2 \) represents a state variable, the Equation (34) defines a state transformation and can be rewritten as

\[
\tilde{x}^2_2 = \dot{x}^1_2 + \dot{x}^2_2.
\]

Collecting all transformations we performed so far, we obtain the complete transformation

\[
\begin{align*}
\dot{x}^1_3 &= x^1(x^1 + 1) \\
\dot{x}^1_2 &= x^2 + 3x^4 \\
\dot{x}^1_1 &= u^1 + 2u^2 \\
\dot{x}^1_0 &= u^2 = u^2,
\end{align*}
\]

(35)

which transforms the system (32) into the explicit triangular normal form

\[
\begin{align*}
\dot{x}^1_3 &= \tilde{x}^2_2 \\
\dot{x}^1_2 &= \dot{u}^1 \dot{x}^1_1 + \dot{x}^1_1 \\
\dot{x}^1_1 &= \dot{u}^1 + \dot{x}^1_1 + \dot{x}^1_1 \\
\dot{x}^1_0 &= \dot{x}^1_1 + u^2.
\end{align*}
\]

with the forward-flat output \( y = (\dot{x}^1_3, \dot{x}^1_1) \). The parameterising map can now be constructed in a systematic way. From the first equation we immediately get the parameterisation of \( \dot{x}^1_2 \). Inserting this parameterisation into the second and the third equation yields the parameterisation of \( \dot{x}^1_1 \) and \( \dot{u}^1 \). Finally, inserting the parameterisation of \( \dot{x}^1_1 \) into the last equation yields the parameterisation of \( u^2 \). With the inverse of (35), the parameterisation of the original system variables \( x \) and \( u \) follows.

6. Conclusion

We have shown that every forward-flat nonlinear discrete-time system with two inputs can be transformed into a structurally flat explicit triangular normal form. In contrast to the implicit triangular form discussed in Kolar, Diwold, et al. (2019), this normal form is a state representation. The transformation is based on the sequences of distributions (5) and (6), that arise in the test for forward-flatness introduced in Kolar, Diwold, et al. (2019). If it is possible to straighten out both sequences of distributions simultaneously by state- and input transformations, then the transformed system has the triangular structure (23). For static feedback linearisable systems, even in the multi-input case with \( m > 2 \), this can always be achieved by a state transformation. For forward-flat systems that are not static feedback linearisable, in contrast, there is no guarantee that both sequences can be straightened out simultaneously, even if additionally input transformations are permitted. However, for forward-flat systems with two inputs, straightening out (5) and (6) by state- and input transformations is always possible. Thus, every forward-flat discrete-time system with two inputs can be transformed into an explicit triangular form.

It is important to emphasise that for flat continuous-time systems no comparable result exists. An obvious reason is that the explicit triangular form allows to read off a forward-flat output which depends only on the state variables. In contrast to continuous-time systems, it is shown in Kolar, Schöberl, et al. (2019) that for forward-flat discrete-time systems such a flat output always exists.

Notes

1. We use the same notation as in Kolar, Schöberl, et al. (2019) and Kolar, Diwold, et al. (2019) with a multi-index \( R = (r_1, \ldots, r_m) \).
2. The largest projectable subdistribution \( D \) (with resp. to \( f \)) of a distribution \( E \) is uniquely determined. Furthermore, if \( E \) is involutive, then \( D \) is also involutive (see Kolar, Diwold, et al., 2019).
3. By the term ‘explicit’ we refer to a state representation (1), in order to distinguish it from the implicit triangular representation discussed in Kolar, Diwold, et al. (2019).
4. Note that subsequently we will use the bar notation for system representations where the \( \Delta \)-sequence is already straightened out.
Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The first author and the second author have been supported by the Austrian Science Fund (FWF) under grant number P 32151 and P 29964.

Notes on contributors

**Johannes Diwold** received the M.Sc. degree in mechatronics from the Johannes Kepler University (JKU), Linz, Austria, in 2019. Since 2019 he is working towards the Ph.D. degree as a Research Assistant at JKU. His research interests include the analysis and control of nonlinear flat discrete-time systems.

**Bernd Kolar** received the M.Sc. degree in mechatronics in 2013 and his Ph.D. degree (Dr.techn.) in control theory in 2017, both from the Johannes Kepler University (JKU), Linz, Austria. From 2017 to 2020 he was a Post-doctoral Researcher at JKU. His research interests include the analysis and control of continuous-time and discrete-time nonlinear flat systems, and the system and control-theoretic analysis of infinite-dimensional systems based on differential-geometric and functional analytic methods.

**Markus Schöberl** received the Dipl.-Ing. degree in mechatronics in 2004 and his Ph.D. degree (Dr.techn.) in control theory in 2007, both from the Johannes Kepler University (JKU), Linz, Austria. From 2007 to 2011 he was a Post-doctoral Researcher at the JKU, and from 2011 to 2014 he was an APART fellowship holder of the Austrian Academy of Sciences. Since 2014 he is an associate professor at JKU. His research interests include nonlinear control and analysis of physical systems based on geometric and algebraic methods, covariant system theory, and Lagrangian and Hamiltonian field theory.

**ORCID**

Markus Schöberl http://orcid.org/0000-0001-5539-7015

References

Aranda-Bricaire, E., Kotta, Ü., & Moog, C. (1996). Linearization of discrete-Time systems. *SIAM Journal on Control and Optimization*, 34(6), 1999–2023. https://doi.org/10.1137/S0363012994267315

Aranda-Bricaire, E., & Moog, C. (2008). Linearization of discrete-time systems by exogenous dynamic feedback. *Automatica*, 44(7), 1707–1717. https://doi.org/10.1016/j.automatica.2007.10.030

Chiasson, J. (1998). A new approach to dynamic feedback linearization control of an induction motor. *IEEE Transactions on Automatic Control*, 43 (3), 391–397. https://doi.org/10.1109/9.661597

Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1992). Sur les systèmes non linéaires différentiellement plats. *Comptes Rendus de L’Académie des Sciences. Série I, Mathématique*, 315, 619–624.

Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6), 1327–1361. https://doi.org/10.1080/00207179508921959

Fliess, M., Lévine, J., Martin, P., & Rouchon, P. (1999). A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Transactions on Automatic Control*, 44(5), 922–937. https://doi.org/10.1109/9.763209

Grizzle, J. (1986). Feedback linearization of discrete-time systems. In A. Bensoussan & J. Lions (Eds.), *Analysis and optimization of systems* (Vol. 83, pp. 273–281). Springer.

Guillot, P., & Millérioux, G. (2020). Flatness and submersivity of discrete-time dynamical systems. *IEEE Control Systems Letters*, 4(2), 337–342. https://doi.org/10.1109/LCSYS.7782633

Jakubczyk, B. (1987). Feedback linearization of discrete-time systems. *Systems & Control Letters*, 9(5), 411–416. https://doi.org/10.1016/0167-6911(87)90070-3

Kaldmäe, A., & Kotta, Ü. (2013). *On flatness of discrete-time nonlinear systems*. Proceedings 9th IFAC Symposium on Nonlinear Control Systems (pp. 588–593). https://doi.org/10.3182/20130904-3-FR-2041.00017

Kolar, B. (2017). *Contributions to the differential geometric analysis and control of flat systems*. Shaker Verlag.

Kolar, B., Diwold, J., & Schöberl, M. (2019). *Necessary and sufficient conditions for difference flatness*. arXiv e-prints. (arXiv:1909.02868v2 [math.OC]).

Kolar, B., Kaldmäe, A., Schöberl, M., Kotta, Ü., & Schlacher, K. (2016). *Construction of flat outputs of nonlinear discrete-time systems in a geometric and an algebraic framework*. Proceedings 10th IFAC Symposium on Nonlinear Control Systems (pp. 808–813). https://doi.org/10.1016/j.ifacol.2016.10.263

Kolar, B., Schöberl, M., & Diwold, J. (2019). Differential-geometric decomposition of flat nonlinear discrete-time systems. arXiv e-prints. (arXiv:1907.00596 [math.OC]).

Kolar, B., Schöberl, M., & Schlacher, K. (2016). A decomposition procedure for the construction of flat outputs of discrete-time nonlinear control systems. Proceedings 22nd International Symposium on Mathematical Theory of Networks and Systems (MTNS) (pp. 775–782). http://hdl.handle.net/11299/181518

Nijmeijer, H., & van der Schaft, A. (1990). *Nonlinear dynamical control systems*. Springer.

Schöberl, M. (2014). *Contributions to the analysis of structural properties of dynamical systems in control and systems theory – A geometric approach*. Shaker Verlag.

Schöberl, M., & Schlacher, K. (2014). On an implicit triangular decomposition of nonlinear control systems that are 1-flat – A constructive approach. *Automatica*, 50 (6), 1649–1655. https://doi.org/10.1016/j.automatica.2014.04.007
Appendix

A.1. Calculation of the distributions of Example 2.1

In this section, we show how the distributions of Example 2.1 can be calculated and briefly recall the notion of projectability discussed in Kolar, Diwold, et al. (2019). A vector field on $\mathcal{X} \times \mathcal{U}$ is said to be projectable if its pointwise pushforward yields a well-defined vector field on $\mathcal{X}^+$. To check whether a vector field is projectable or not we introduce adapted coordinates

$$\theta^i = f^i(x, u), \quad i = 1, \ldots, n$$
$$\xi^j = h^j(x, u), \quad j = 1, \ldots, m \tag{A1}$$
on $\mathcal{X} \times \mathcal{U}$, where the $m$ functions $h^j(x, u)$ must be chosen such that the map (A1) forms locally a diffeomorphism. In such coordinates, a vector field is projectable if and only if it has the form

$$a'(\theta)\partial_{\theta^i} + b'(\theta, \xi)\partial_{\xi^j}, \tag{A2}$$

with the functions $a'(\theta)$ independent of $\xi$. Its pushforward is the well-defined vector field

$$a'(x^+)\partial_{\hat{x}^i} \tag{A3}$$
on $\mathcal{X}^+$. Similarly, a distribution $D$ on $\mathcal{X} \times \mathcal{U}$ is said to be projectable if and only if it admits a basis of projectable vector fields.

For the system (8) of Example 2.1, a possible choice of adapted coordinates $(\theta, \xi)$ on $\mathcal{X} \times \mathcal{U}$ is given by the transformation

$$\theta^1 = x^4 + x^3 + x^5 \quad \theta^2 = u^1 \quad \xi^1 = x^4$$
$$\theta^3 = x^4(x^4 + 1) + x^3 \quad \theta^1 = u^2 \quad \xi^2 = x^3.$$In such coordinates, it can be immediately seen that the distribution

$$E_0 = \text{span}[\partial_{\theta^1}, \partial_{\theta^2}]$$
is completely projectable, since $\partial_{\theta^1}$ and $\partial_{\theta^2}$ are obviously of the form (A2). Thus, $D_0 = E_0$ and the pushforward $f_*(D_0)$ yields the well-defined distribution $\Delta_1 = \text{span}[\partial_{\hat{x}^1}, \partial_{\hat{x}^2}]$ on $\mathcal{X}^+$. The distribution

$$E_1 = \text{span}[\partial_{\theta^1}, \partial_{\theta^2}, (\xi^1 + 1)\partial_{\theta^1} + \partial_{\theta^3}]$$
in contrast, possesses no basis of projectable vector fields (A2), and is thus not completely projectable. The largest projectable subdistribution $D_1 \subset E_1$ is given by $D_1 = \text{span}[\partial_{\theta^1}, \partial_{\theta^2}, \partial_{\theta^3}]$. The pushforward $f_*(D_1)$ yields the well-defined distribution $\Delta_2 = \text{span}[\partial_{\hat{x}^1}, \partial_{\hat{x}^2}, \partial_{\hat{x}^3}]$ on $\mathcal{X}^+$. Finally, the distribution $E_2$ admits a basis of the form

$$E_2 = \text{span}[\partial_{\theta^1}, \partial_{\theta^2}, \partial_{\theta^3} + \partial_{\hat{x}^2}, \partial_{\theta^3} = (\xi^1 + 1)\partial_{\hat{x}^2}]$$

and is thus again completely projectable. Hence, $D_2 = E_2$ and the pushforward $f_*(D_2)$ yields the well-defined distribution $\Delta_3 = \text{span}[\partial_{\hat{x}^1}, \partial_{\hat{x}^2}, \partial_{\hat{x}^3}, \partial_{\hat{x}^4}, \partial_{\hat{x}^5}]$ on $\mathcal{X}^+$.

A.2. Proof of Lemma 3.2

Due to the involutivity, all pairwise Lie Brackets must be contained in $D$, i.e.

$$[\partial_{\xi^i}, \partial_{\xi^l} + \alpha(\xi)\partial_{\xi^l}] \subset D, \quad l = 1, \ldots, k - 1.$$Because of the special structure of the basis of $D$, this implies that all pairwise Lie brackets vanish identically. Consequently, the coefficient $\alpha$ meets $\partial_{\xi^i}\alpha = 0$ for $l = 1, \ldots, k - 1$, i.e. $\alpha$ is independent of $\xi^1, \ldots, \xi^{k-1}$. Next, the flow $\phi_{\xi}(\zeta_0)$ of the vector field $\partial_{\xi^l} + \alpha(\xi^k, \ldots, \xi^n)\partial_{\xi_k}$ is of the form

$$\xi^l(t, \zeta_0) = t + \zeta^l_0$$
$$\xi^l(t, \zeta_0) = \phi_{\xi}(\zeta^k_0, \ldots, \zeta^n_0),$$i.e. it only affects the coordinates $\xi^i$ and $\xi^l$. According to the flow-box theorem, by setting $t = \xi^l$, $\zeta^l_0 = 0$, $\zeta^n_0 = \xi^l$ and $\zeta_0 = \zeta^l$ for $l = k, \ldots, n$ with $l \neq i, j$ on the right hand side, we obtain a coordinate transformation which transforms the above vector field into the form $\partial_{\xi^j}$. In fact, only $\xi^j$ is replaced by the transformed coordinate $\tilde{\xi}^j$, and all other coordinates remain unchanged. In these coordinates, the distribution $D$ reads as (19). The inverse coordinate transformation is of the form (18).

A.3. Proof of Theorem 4.1

First, we show that the variables $z_0, \ldots, z_{k-1}$ contain all inputs and states except $\hat{x}_k$. Since the distributions (12) are straightened out, according to (15) the distribution $E_{k-1}$ reads as

$$E_{k-1} = \text{span}[\partial_{x^i}, \partial_1, \ldots, \partial_{x_{k-1}}].$$Lemma 2.1 guarantees that $E_{k-1}$ is completely projectable, and thus it coincides with the distribution

$$D_{k-1} = \text{span}[\partial_{x_0}, \ldots, \partial_{x_{k-1}}].$$Thus, the variables $z_0, \ldots, z_{k-1}$ contain all inputs and states except $\hat{x}_k$. The property (24) is a consequence of $D_{k-1} \subset E_{k-1}$, i.e.

$$\text{span}[\partial_{x_0}, \ldots, \partial_{x_{k-1}}] \subset \text{span}[\partial_{x^i}, \partial_1, \ldots, \partial_{x_{k-1}}] \tag{A4}$$

and

$$D_{k-1} = E_{k-1}, \text{i.e.}$$

$$\text{span}[\partial_{x_0}, \ldots, \partial_{x_{k-1}}] = \text{span}[\partial_{x^i}, \partial_1, \ldots, \partial_{x_{k-1}}]. \tag{A5}$$Because of (A4), the variables $\hat{x}_k$ cannot be contained in $(z_0, \ldots, z_{k-1})$. However, according to (A5), they must be contained in $(z_0, \ldots, z_{k-1})$.

The triangular structure of (23) is a consequence of

$$f_*(D_{k-1}) = \Delta_k = \text{span}[\partial_{x_1}, \ldots, \partial_{x_k}], \quad k = 1, \ldots, \tilde{k}. \tag{A6}$$
For $k = 0$, from (A6) and $D_0 = \text{span}\{\partial_{z_0}\}$ we get $\partial_{z_0}f_i = 0$ for $i = 2, \ldots, \tilde{k}$, i.e.
\[
\begin{align*}
\hat{x}_k^+ &= f_k(\hat{x}_k, z_{k-1}, \ldots, z_1) \\
\vdots \\
\hat{x}_2^+ &= f_2(\hat{x}_k, z_{k-1}, \ldots, z_1) \\
\hat{x}_1^+ &= f_1(\hat{x}_k, z_{k-1}, \ldots, z_0).
\end{align*}
\]
Furthermore, because of $\dim(\Delta_1) = \dim(\hat{x}_1)$, the rank condition $\text{rank}(\partial_{z_0}f_i) = \dim(\hat{x}_1)$ follows. Next, for $k = 1$, from (A6) and $D_1 = \text{span}\{\partial_{z_0}, \partial_{z_1}\}$, we get $\partial_{z_1}f_i = 0$ for $i = 3, \ldots, \tilde{k}$, i.e.
\[
\begin{align*}
\hat{x}_k^+ &= f_k(\hat{x}_k, z_{k-1}, \ldots, z_2) \\
\vdots \\
\hat{x}_3^+ &= f_3(\hat{x}_k, z_{k-1}, \ldots, z_2) \\
\hat{x}_2^+ &= f_2(\hat{x}_k, z_{k-1}, \ldots, z_1) \\
\hat{x}_1^+ &= f_1(\hat{x}_k, z_{k-1}, \ldots, z_0).
\end{align*}
\]
Again, because of $\dim(\Delta_2) = \dim(\hat{x}_1) + \dim(\hat{x}_2)$, the rank condition $\text{rank}(\partial_{z_1}f_2) = \dim(\hat{x}_2)$ follows. Repeating this argumentation shows that the system has the triangular structure (23) and meets the rank conditions (25).