A quotient method for designing nonlinear controllers

S. S. Willson, Philippe Müllhaupt and Dominique Bonvin

***Abstract*— An algorithmic method is proposed to design stabilizing control laws for a class of nonlinear systems that comprises single-input feedback-linearizable systems and a particular set of single-input non feedback-linearizable systems. The method proceeds iteratively and consists of two stages; it converts the system into cascade form and reduces the dimension at every step by creating quotient manifold in the forward stage, while it constructs the feedback law iteratively in the backward stage. The paper shows that the construction of these quotient manifolds is well defined for feedback-linearizable system and, furthermore, it can also be applied to a class of non feedback-linearizable systems.**

I. Introduction

Several elegant and powerful methods are available to handle feedback-linearizable (FBL) systems and certain classes of single-input nonlinear systems ([1], [2], [3] and [4]). These methods proceed by either solving partial differential equations or finding an appropriate function or surface that guides the control design. For example, the passivity-based approach to handle Lagrangian or Hamiltonian structures [5] and the feedback linearization approach [2], [3] both require solving partial differential equations. On the other hand, in Lyapunov's direct method [1], an appropriate control Lyapunov function is essential to the success of the method, while in the sliding-mode method [6], the selection of an appropriate sliding surface is key.

In contrast, for linear systems, there are many algorithmic methods. Linear-quadratic regulation (LQR) is one of the most popular methods for controlling linear systems, not only because of the intrinsic robustness of the resulting controllers, but because of its algorithmic nature. Indeed, it rests on solving a Riccati equation, which can be done via successive transformations that bring the original equations into a simpler form, from which the control gains can be easily inferred. Other methods such as pole placement also resort to successive transformations that bring the original system into a more manageable form. For example, a single-input linear system can be brought to upper Hessenberg form [7] using the Miminis-Paige algorithm [8] or the staircase algorithm [9], [10] before proceeding to stabilization.

The aim of this paper is to present an algorithmic method for designing an asymptotically stabilizing controller for a class of nonlinear system that comprises, but is not limited to, single-input FBL systems. The present approach builds on an algorithm designed to compute the output for static feedback linearization [11]. However, that algorithm shares the limitations of feedback linearization since the control law uses feedback linearization and linear control techniques. To overcome these limitations, this paper proposes to embed control design in the algorithm, which helps go around the singularities when they would arise in feedback linearization. Furthermore, with certain assumptions, the proposed approach can approximate non-FBL systems as FBL systems. For example, it has been possible to achieve both swing-up and stabilization of the acrobot [12] and the inverted pendulum [13].

The approach has two main algorithmic parts, namely (i) a forward stage that transforms the system into a cascade form at each step, and (ii) a backward stage that builds successive control laws, starting from the smallest cascade system obtained at the end of the forward stage to the complete system. The paper will present the following new results:

- A control design that can exploit the degrees of freedom available in the forward stage to avoid the singularities that appear in feedback linearization.
- The conditions for the algorithm to be able to incorporate control design.
- Under certain conditions, backstepping can be used with this algorithm, thus resulting in a globally asymptotically stabilizing control law.
- The present results are perfectly in line with the existing results for feedback linearization and state linearization.

The paper is organized as follows. Section 2 presents preliminary material, in particular the basic lemma and a corollary that supports the method developed in this work. Section 3 provides the algorithms for both the forward and backward stages. The approach is illustrated via the example of a FBL system in Section 4. Section 5 discusses the conditions under which backstepping can be used with this algorithm, while Section 6 presents concluding remarks.

II. Preliminaries

The preliminary material required for the forward stage is presented in [11]. In this section, results required to prove the stability of the control law designed through the backward stage are presented.

Lemma 1: Consider the system

$$\begin{aligned} \dot{\mathbf{z}} &= f(\mathbf{z}, y), \\ \dot{y} &= p(y), \end{aligned} \tag{1}$$

and let $\dot{y} = p(y)$ have an asymptotically stable equilibrium at $y = 0$. If $\dot{\mathbf{z}} = f(\mathbf{z}, 0)$ has an asymptotically stable equi-

Laboratoire d'Automatique, STI-DGM, Station 9, Ecole Polytechnique Federale de lausanne, CH-1015 lausanne, Switzerland. willson.shibani@epfl.ch philippe.muellhaupt@epfl.ch dominique.bonvin@epfl.ch

librium at $\mathbf{z} = 0$, then the system (1) has an asymptotically stable equilibrium at $(\mathbf{z}, y) = (0, 0)$.
This result is provided in Appendix B.2 of [2]. A corollary of this lemma is given next.

Corollary 1: Consider the system

$$\dot{\mathbf{x}} = f_x(\mathbf{x}, \xi), \tag{2}$$
$$\dot{\xi} = f_\xi(\mathbf{x}, \xi) + g(\mathbf{x}, \xi)u, \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^{n-1}$, $\xi \in \mathbb{R}$, $u \in \mathbb{R}$, $f_x : \mathbb{R}^n \to \mathbb{R}^{n-1}$, $f_\xi : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ with the equilibrium point $(\mathbf{x}, \xi) = (0, 0)$ and $g(\mathbf{x}, \xi) \neq 0$ in the domain of operation. If there exists a function $\xi_d(\mathbf{x})$ with $\xi_d(0) = 0$, which, when substituted for ξ in (2), asymptotically stabilizes the system (2), then

$$u = \frac{\frac{\partial \xi_d}{\partial x} f_x(\mathbf{x}, \xi) - f_\xi(\mathbf{x}, \xi) + k(\xi_d(\mathbf{x}) - \xi)}{g(\mathbf{x}, \xi)}, \tag{4}$$

where k is any arbitrary positive constant, locally asymptotically stabilizes both (2) and (3).

Proof: Let the error variable be

$$e = \xi_d(\mathbf{x}) - \xi. \tag{5}$$

Substituting (4) and (5) into (2) and (3) gives the new set of equations

$$\dot{\mathbf{x}} = f_x(\mathbf{x}, \xi_d(\mathbf{x}) - e), \tag{6}$$
$$\dot{e} = -ke. \tag{7}$$

It can be shown using Lemma 1 that the above system is locally asymptotically stable since

- $e = 0$ is an asymptotically stable equilibrium of (7),
- for $e = 0$, by assumption, $\dot{\mathbf{x}} = f_x(\mathbf{x}, \xi_d(\mathbf{x}))$ is also asymptotically stable at $\mathbf{x} = 0$.

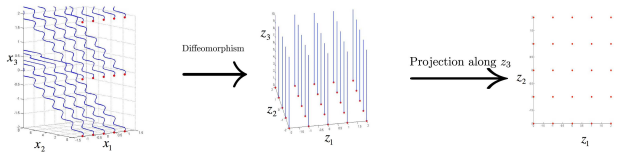
Hence, the system (6)-(7) is locally asymptotically stable at $(\mathbf{x}, e) = (0, 0)$, which implies that the system is locally asymptotically stable at $(\mathbf{x}, \xi) = (0, 0)$. ∎

III. The Algorithm

The algorithm consists of two stages. In the forward stage, the system is systematically reduced using the algorithm described in [11]. This paper presents additional results required to adapt the algorithm to the present application. The effect of the input is isolated on a single variable, and this variable is then used to control the rest of the system. This step is repeated as many times as possible to obtain a smaller-dimensional cascade system. The necessary condition for building equivalence classes at each iteration is also provided. In the stabilizing stage, a locally asymptotically stabilizing controller is constructed iteratively by proceeding backwards.

A. Forward stage for achieving cascade form

The effect of the input is isolated on a single state, which is then used to control the remaining states. This is achieved by generating a quotient manifold using the integral manifold of the input vector field as the equivalent class. To achieve this, the integral manifold of the input vector field is straightened and aligned along the last coordinate axis by suitable diffeomorphism. Then, projection is taken along the last coordinate to generate the quotient manifold. This is visualized for a three-dimensional system in Figure 1.



(a) Integral manifold of $g(\mathbf{x})$ (b) The diffeomorphism $\mathbf{z} = \Phi(\mathbf{x})$ straightens the integral manifold (c) Projection gives the quotient manifold

Fig. 1. Method for generating quotient manifold for a 3-dimensional system.

Let us consider a p-dimensional system of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})u, \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^p$, $f : \mathbb{R}^p \to \mathbb{R}^p$ and $g : \mathbb{R}^p \to \mathbb{R}^p$. The generic dimension p is chosen here since the system dimension reduces with every iteration.
The effect of u is isolated on the last component of $\mathbf{x}$ through suitable diffeomorphism. This is possible since the distribution $\Delta = \text{span}\{g(\mathbf{x})\}$ is nonsingular and trivially involutive (it has dimension 1). By Frobenius theorem [2], Δ is completely integrable. Hence, there exist $p - 1$ smooth functions $\phi_1(\mathbf{x}), ..., \phi_{p-1}(\mathbf{x})$ with linearly independent differentials such that

$$L_g \phi_i(\mathbf{x}) = 0, \qquad 1 \leq i \leq p - 1.$$

Constructing $\phi_1(\mathbf{x}), ..., \phi_{p-1}(\mathbf{x})$ can be achieved using the Lagrange subsidiary system [11] or by using flowbox theorem [14]. These ϕ_i's are collected to define the diffeomorphism $\mathbf{\Phi}_p(\mathbf{x})$,

$$\mathbf{\Phi_p}(\mathbf{x}) = \begin{pmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_{p-1}(\mathbf{x}) \\ \gamma(\mathbf{x}) \end{pmatrix}, \tag{9}$$

with the p^{th} component being any function $\gamma(\mathbf{x})$ chosen such that

$$\text{rank}\left(\frac{\partial \mathbf{\Phi}_p(\mathbf{x})}{\partial \mathbf{x}}\right) = p. \tag{10}$$

Define $\mathbf{z} \triangleq \mathbf{\Phi}_p(\mathbf{x})$ with $\mathbf{z} \in \mathbb{R}^p$ so that (9) is a diffeomorphism with the inverse $\mathbf{x} = \mathbf{\Phi}_p^{-1}(\mathbf{z})$:

$$\begin{aligned}
\mathbf{z} &= \mathbf{\Phi}_p(\mathbf{x}), \\
\dot{\mathbf{z}} &= \frac{\partial \mathbf{\Phi}_p(\mathbf{x})}{\partial \mathbf{x}} \dot{\mathbf{x}} \\
&= \frac{\partial \mathbf{\Phi}_p(\mathbf{x})}{\partial \mathbf{x}} (f(\mathbf{x}) + g(\mathbf{x})u) \\
&= \frac{\partial \mathbf{\Phi}_p(\mathbf{x})}{\partial \mathbf{x}} f(\mathbf{x})|_{\mathbf{x}=\mathbf{\Phi}_p^{-1}(\mathbf{z})} + \frac{\partial \mathbf{\Phi}_p(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}) u|_{x=\mathbf{\Phi}_p^{-1}(\mathbf{z})} \\
\dot{\mathbf{z}} &= f_z(\mathbf{z}) + g_z(\mathbf{z})u. \qquad (11)
\end{aligned}$$

This **z**-system is in cascade form. The construction of $\mathbf{\Phi}_p(\mathbf{x})$ implies that the first $p-1$ components of $g_z(\mathbf{z})$ are zero, i.e. $g_z(\mathbf{z}) = (0, 0, ..., 0, \alpha(\mathbf{z}))^T$. Hence, in the new coordinate system, the input affects only the last coordinate. Next, in order to reduce the dimension of the system, the following projection map is defined.

Definition 1: The projection map, $\Pr : \mathbb{R}^k \to \mathbb{R}^{k-1}$, is defined as $Pr((a_1, a_2, ..., a_{k-1}, a_k)^T) = (a_1, a_2, ..., a_{k-1})^T$.

This projection map is applied to system (11),

$$\Pr(\dot{\mathbf{z}}) = \Pr(f_z(\mathbf{z}) + g_z(\mathbf{z})u), \qquad (12)$$

with the effect of removing the p^{th} component of the state. Let $\hat{\mathbf{x}} \triangleq \Pr(\mathbf{z})$ and $\hat{f}_z(\hat{\mathbf{x}}, z_p) \triangleq \Pr(f_z(\mathbf{z}))$. Since $\Pr(g_z(\mathbf{z})) = 0$, (12) reduces to

$$\dot{\hat{\mathbf{x}}} = \hat{f}_z(\hat{\mathbf{x}}, z_p), \qquad (13)$$

where z_p is the p^{th} component of **z**. The complete step can be summarized through the commutative diagram given in Figure 2.

Fig. 2. Commutative diagram for a step in the forward stage.

If $\hat{f}_z(\hat{\mathbf{x}}, z_p)$ depends linearly on z_p, which is the case when the distribution $\Delta = \text{span}\{g, [f, g]\}$ is involutive and a certain condition is satisfied (see Lemma 3 below), then (13) can be separated as follows:

$$\dot{\hat{\mathbf{x}}} = f_{\hat{x}}(\hat{\mathbf{x}}) + g_{\hat{x}}(\hat{\mathbf{x}})\hat{u}, \qquad (14)$$

where $\hat{u} = z_p$ and with the vector fields $f_{\hat{x}}(\hat{\mathbf{x}}) : \mathbb{R}^{p-1} \to \mathbb{R}^{p-1}$ and $g_{\hat{x}}(\hat{\mathbf{x}}) : \mathbb{R}^{p-1} \to \mathbb{R}^{p-1}$. The necessary condition for the separation given by (14) is presented through the following lemma. For notational purpose, the operator $\mathbf{\Phi}_*$ is defined as $g_z(\mathbf{z}) = \mathbf{\Phi}_* g(\mathbf{x}) \triangleq \frac{\partial \mathbf{\Phi}}{\partial \mathbf{x}} g(\mathbf{x})|_{\mathbf{x}=\mathbf{\Phi}^{-1}(\mathbf{z})}$.

Lemma 2: If the separation given by (14) is possible for system (8), then the distribution $\Delta = \text{span}\{g, [f, g]\}$ is involutive.

Proof: When the separation given by (14) is possible, we can write

$$\Pr\left(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}\right) = 0, \qquad (15)$$

since $\mathbf{\Phi}_* f$ is only linearly dependent on z_p. Also, $\mathbf{\Phi}_* g = (0, \cdots, 0, \alpha(\mathbf{z}))^T$. Hence $\Pr(\mathbf{\Phi}_* g) = 0$. Next, consider

$$\begin{aligned}
[\mathbf{\Phi}_* f, \mathbf{\Phi}_* g] &= \frac{\partial \mathbf{\Phi}_* g}{\partial \mathbf{z}} \mathbf{\Phi}_* f - \frac{\partial \mathbf{\Phi}_* f}{\partial \mathbf{z}} \mathbf{\Phi}_* g \\
&= \begin{pmatrix} 0_{p-1,p} \\ \frac{\partial \alpha(\mathbf{z})}{\partial \mathbf{z}} \end{pmatrix} \mathbf{\Phi}_* f - \frac{\partial \mathbf{\Phi}_* f}{\partial \mathbf{z}} \begin{pmatrix} 0_{p-1,1} \\ \alpha(\mathbf{z}) \end{pmatrix} \\
&= \begin{pmatrix} 0_{p-1,1} \\ \frac{\partial \alpha(\mathbf{z})}{\partial \mathbf{z}} \mathbf{\Phi}_* f \end{pmatrix} - \alpha(\mathbf{z}) \frac{\partial \mathbf{\Phi}_* f}{\partial z_p} \\
&= \begin{pmatrix} 0_{p-1,1} \\ \beta(\mathbf{z}) \end{pmatrix} - \alpha(\mathbf{z}) \frac{\partial \mathbf{\Phi}_* f}{\partial z_p},
\end{aligned}$$

where

$$\beta(\mathbf{z}) \triangleq \frac{\partial \alpha(\mathbf{z})}{\partial \mathbf{z}} \mathbf{\Phi}_* f. \qquad (16)$$

Projection gives:

$$\begin{aligned}
\Pr([\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]) &= \Pr(\begin{pmatrix} 0_{p-1,1} \\ \beta(\mathbf{z}) \end{pmatrix}) - \Pr(\alpha(\mathbf{z}) \frac{\partial \mathbf{\Phi}_* f}{\partial z_p}) \\
&= -\alpha(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}). \qquad (17)
\end{aligned}$$

Consider next

$$\begin{aligned}
[\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] &= [\mathbf{\Phi}_* g, \begin{pmatrix} 0_{p-1,1} \\ \beta(\mathbf{z}) \end{pmatrix} - \alpha(\mathbf{z}) \tfrac{\partial \mathbf{\Phi}_* f}{\partial z_p}] \\
&= \frac{\partial \begin{pmatrix} 0_{p-1,1} \\ \beta(\mathbf{z}) \end{pmatrix}}{\partial \mathbf{z}} \mathbf{\Phi}_* g - \frac{\partial (\alpha(\mathbf{z}) \frac{\partial \mathbf{\Phi}_* f}{\partial z_p})}{\partial \mathbf{z}} \mathbf{\Phi}_* g \\
&\quad - \tfrac{\partial \mathbf{\Phi}_* g}{\partial \mathbf{z}} [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g] \\
&= \begin{pmatrix} 0_{p-1,1} \\ \alpha(\mathbf{z}) \frac{\partial \beta(\mathbf{z})}{\partial z_p} \end{pmatrix} - \alpha(\mathbf{z}) \tfrac{\partial \alpha(\mathbf{z})}{\partial z_p} \tfrac{\partial \mathbf{\Phi}_* f}{\partial z_p} \\
&\quad - \alpha(\mathbf{z})^2 \tfrac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2} - \begin{pmatrix} 0_{p-1,1} \\ \frac{\partial \alpha(\mathbf{z})}{\partial \mathbf{z}} [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g] \end{pmatrix}. \qquad (18)
\end{aligned}$$

Applying the projection map and using (15) gives:

$$\begin{aligned}
\Pr([\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]]) &= 0 - \Pr(\alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} \frac{\partial \mathbf{\Phi}_* f}{\partial z_p}) \\
&\quad - \Pr(\alpha(\mathbf{z})^2 \frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}) - 0 \qquad (19) \\
&= -\alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}). \qquad (20)
\end{aligned}$$

Using (17) and (20), we can write:

$$\Pr[\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] = \kappa_1(\mathbf{z}) \Pr[\mathbf{\Phi}_* f, \mathbf{\Phi}_* g], \qquad (21)$$

$$\kappa_1(\mathbf{z}) = \frac{\partial \alpha(\mathbf{z})}{\partial z_p}. \qquad (22)$$

By defining

$$\kappa_2(\mathbf{z}) := \frac{e_p^T [\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] - \kappa_1(\mathbf{z}) e_p^T [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]}{\alpha(\mathbf{z})},$$

it is possible to write

$$[\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] = \kappa_1(\mathbf{z}) [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g] + \kappa_2(\mathbf{z}) \mathbf{\Phi}_* g. \qquad (23)$$

Hence the distribution $\Delta = \text{span}\{\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]\}$ is involutive. Because a diffeomorphism does not affect the involutivity properties, $\text{span}\{g, [f, g]\}$ is also involutive. Hence, if the separation given by (14) is possible, then the distribution $\Delta = \text{span}\{g, [f, g]\}$ has to be involutive. ∎

The sufficient condition for the separation given by (14) is given below.

Lemma 3: If the distribution $\Delta = \text{span}\{g, [f, g]\}$ is involutive and if $\gamma(x)$ is chosen such that $\alpha(\mathbf{z})$ satisfies (22), then the separation given by (14) is possible.

Proof: If $\Delta = \text{span}\{g, [f, g]\}$ is involutive, then (23) and (21) are satisfied. Substituting (22) and (17) into (21) gives:

$$\Pr[\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] = -\alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}). \quad (24)$$

Next, equating the RHS of (24) with RHS of (19) gives:

$$\begin{aligned} \alpha(\mathbf{z})^2 \Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}) &- \alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}) \\ &= -\alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}), \\ \Rightarrow \alpha(\mathbf{z})^2 \Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}) &= 0, \\ \Rightarrow \Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}) &= 0. \end{aligned}$$

Note that $\alpha(\mathbf{z}) \neq 0$ is a requirement for choosing $\gamma(\mathbf{x})$. Since all the higher order derivatives of $\Pr(\mathbf{\Phi}_* f)$ with respect to z_p are zero, it can be concluded that $\Pr(\mathbf{\Phi}_* f)$ is only linearly dependent on z_p. Hence, the linear separation given by (14) is possible. ∎

It has also been shown in Corollary 1 in [11] that, if the distribution $\Delta = \text{span}\{g, [f, g]\}$ is involutive, then it is always possible to construct $(p-2)$ 1-forms, independent of z_p, that span the null space of $\Pr(\mathbf{\Phi}_* [f, g])$. The following lemma show the way to construct such 1-forms even if the separation given by (14) is not possible.

Lemma 4: If $\Delta = \text{span}\{g, [f, g]\}$ is involutive, then it is always possible to construct $(p-2)$ 1-forms, independent of z_p, that span the kernel of $\Pr(\mathbf{\Phi}_* [f, g])$. The same 1-forms also span the kernel of

$$g_{\hat{x}}(\hat{x}, z_p) := \Pr(\mathbf{\Phi}_* f) - \Pr(\mathbf{\Phi}_* f)|_{z_p = 0}, \quad (25)$$

where $\hat{x} := \Pr(\mathbf{z}) = (z_1, \ldots, z_{p-1})$.

Proof: If $\Delta = \text{span}\{g, [f, g]\}$ is involutive, then (23) and (21) are satisfied. Substituting (17) into (21) and comparing with (19) gives:

$$\begin{aligned} \alpha(\mathbf{z})^2 \Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}) &- \alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}) \\ &= -\kappa_1(\mathbf{z}) \alpha(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}). \end{aligned}$$

Re-arranging the above equation gives:

$$\Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}) = \lambda_2(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}), \quad (26)$$

where $\lambda_2(\mathbf{z})$ is a scalar function and is defined as

$$\lambda_2(\mathbf{z}) := \frac{\alpha(\mathbf{z}) \frac{\partial \alpha(\mathbf{z})}{\partial z_p} - \kappa_1(\mathbf{z}) \alpha(\mathbf{z})}{\alpha(\mathbf{z})^2}.$$

It is easy to see, from the relation (26), that all the higher derivatives will also be proportional to the first-order derivative, that is,

$$\begin{aligned} \Pr(\frac{\partial^3 \mathbf{\Phi}_* f}{\partial z_p^3}) &= \lambda_3(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}), \\ \Pr(\frac{\partial^4 \mathbf{\Phi}_* f}{\partial z_p^4}) &= \lambda_4(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}), \\ &\vdots \end{aligned}$$

and so on. Since all these equations are true for all $\mathbf{z}$, they are also true for $z_p = 0$, and thus:

$$\begin{aligned} \Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2})|_{z_p=0} &= \lambda_2(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p})|_{z_p=0}, \\ \Pr(\frac{\partial^3 \mathbf{\Phi}_* f}{\partial z_p^3})|_{z_p=0} &= \lambda_3(\mathbf{z}) \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p})|_{z_p=0}, \quad (27) \\ &\vdots \end{aligned}$$

Let us define the $(p-1)$-dimensional vector field $\tilde{g}_{\hat{x}}(\hat{x})$ as

$$\tilde{g}_{\hat{x}}(\hat{x}) := \left. \frac{\partial \Pr(\mathbf{\Phi}_* f)}{\partial z_p} \right|_{z_p = 0}. \quad (28)$$

It is always possible (guaranteed by Frobenious theorem) to construct $(p-2)$ linearly independent 1-forms, $\omega_1, \ldots, \omega_{p-2}$, such that they span the kernel of $\tilde{g}_{\hat{x}}(\hat{x})$, i.e. $\omega_i . \tilde{g}_{\hat{x}}(\hat{x}) = 0, \forall i = 1, \ldots, p-2$. Since $\tilde{g}_{\hat{x}}(\hat{x})$ does not contain z_p, $\omega_1, \ldots, \omega_{p-2}$ are independent of z_p. Using the definition (28) along with the relation (27) gives:

$$\begin{aligned} \omega_i . \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p})|_{z_p=0} &= 0, \\ \omega_i . \Pr(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2})|_{z_p=0} &= \lambda_2(\mathbf{z}) \omega_i . \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p})|_{z_p=0} = 0, \\ \omega_i . \Pr(\frac{\partial^3 \mathbf{\Phi}_* f}{\partial z_p^3})|_{z_p=0} &= \lambda_3(\mathbf{z}) \omega_i . \Pr(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p})|_{z_p=0} = 0, \\ &\vdots \quad (29) \end{aligned}$$

$\forall i = 1, \ldots, p-2$. From (17),

$$\begin{aligned} \Pr(\mathbf{\Phi}_* [f, g]) &= \Pr([\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]) \\ &= -\alpha(\mathbf{z}) \frac{\partial \Pr(\mathbf{\Phi}_* f)}{\partial z_p}. \quad (30) \end{aligned}$$

Next, consider the Taylor expansion of $\frac{\partial \Pr(\mathbf{\Phi}_* f)}{\partial z_p}$ around $z_p = 0$:

$$\begin{aligned} \frac{\partial \Pr(\mathbf{\Phi}_* f)}{\partial z_p} &= \Pr \left(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p} \right) \Big|_{z_p=0} \\ &+ \Pr \left(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2} \right) \Big|_{z_p=0} z_p \\ &+ \frac{1}{2!} . \Pr \left(\frac{\partial^3 \mathbf{\Phi}_* f}{\partial z_p^3} \right) \Big|_{z_p=0} z_p^2 + \ldots \quad (31) \end{aligned}$$

Using the relations (30) and (31) and results from (29) gives $\omega_i.\text{Pr}(\mathbf{\Phi}_*[f,g]) = 0, \forall i = 1,\ldots,p-2$. Hence, $\omega_1,\ldots,\omega_{p-2}$, which are independent of z_p, span the kernel of $\text{Pr}(\mathbf{\Phi}_*[f,g])$.

Next, considering the Taylor expansion of $\text{Pr}(\mathbf{\Phi}_* f)$ and using the definition (25) yields:

$$\begin{aligned} g_{\hat{x}}(\hat{x}, z_p) &= \text{Pr}\left(\frac{\partial \mathbf{\Phi}_* f}{\partial z_p}\right)\Big|_{z_p=0}.z_p \\ &+ \frac{1}{2!}.\text{Pr}\left(\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2}\right)\Big|_{z_p=0} z_p^2 \\ &+ \frac{1}{3!}.\text{Pr}\left(\frac{\partial^3 \mathbf{\Phi}_* f}{\partial z_p^3}\right)\Big|_{z_p=0} z_p^3 + \ldots \quad (32) \end{aligned}$$

Considering $\omega_i.g_{\hat{x}}(\hat{x}, z_p)$, when substituted with (32) and using the relations (29), gives $\omega_i.g_{\hat{x}}(\hat{x}, z_p) = 0, \forall i = 1,\ldots,p-2$. Since $\omega_1,\ldots,\omega_{p-2}$ are linearly independent 1-forms, they span the kernel of $g_{\hat{x}}(\hat{x}, z_p)$. ∎

These lemmas provide an alternative explanation for the termination condition of the state and feedback linearizing algorithms provided in [15]. For the feedback linearizing algorithm, the termination condition is the same as (26). Hence, at any stage, if the distribution $\Delta = \text{span}\{g, [f,g]\}$ is not involutive, then the system is not feedback linearizable. A stronger condition is required for state linearization since, at every step, $\alpha(\mathbf{z})$ has to be constant (in case of [15], $\alpha(\mathbf{z}) = 1$). This results in the termination condition (15), which is the same as $\text{Pr}(\mathbf{\Phi}_*[g,[f,g]]) = 0$. The following corollary deals with the termination condition of the state linearizing algorithm in [15] and presents an alternative explanation for it.

Corollary 2: If $\Delta = \text{span}\{g, [f,g]\}$ is involutive, $\gamma(\mathbf{x})$ can be chosen such that $\alpha(\mathbf{z})$ is constant if and only if $\text{Pr}(\mathbf{\Phi}_*[g,[f,g]]) = 0$.

Proof: If $\Delta = \text{span}\{g, [f,g]\}$ is involutive, then (23) and (21) are satisfied.

Necessary condition: If $\text{Pr}(\mathbf{\Phi}_*[g,[f,g]]) = 0$, then from (21), $\kappa_1(\mathbf{z}) = 0$ since the controllability condition stipulates $\text{Pr}(\mathbf{\Phi}_*[f,g]) \neq 0$. This in turn, using (22), implies $\frac{\partial \alpha(\mathbf{z})}{\partial z_p} = 0$, which is true if $\alpha(\mathbf{z})$ is constant. Hence $\gamma(\mathbf{x})$ can be chosen such that $\alpha(\mathbf{z})$ is constant.

Sufficiency condition: If $\gamma(\mathbf{x})$ is chosen such that $\alpha(\mathbf{z})$ is constant. From (22), $\kappa_1 = \frac{\partial \alpha(\mathbf{z})}{\partial z_p} = 0$. This implies, using (21), $\text{Pr}(\mathbf{\Phi}_*[g,[f,g]]) = 0$. ∎

Additionally, it is essential for state linearization that the last function of $(\mathbf{\Phi}_* f)$ be a linear function of z_p. This results from the fact that the last line of $\mathbf{\Phi}_* f$ is not transformed since the Pr function removes it. Hence, if a linear system is sought, $e_p^T(\mathbf{\Phi}_* f)$ should be a linear function of z_p, where $e_p = (0,\ldots,0,1)$ is the standard unit vector. If $\alpha(\mathbf{z})$ is constant, using (16) implies $\beta(\mathbf{z}) = 0$, which using (18) and the fact that $\frac{\partial \alpha(\mathbf{z})}{\partial z_p} = 0$, further implies:

$$e_p^T[\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] = \alpha(\mathbf{z}) \frac{\partial^2 e_p^T \mathbf{\Phi}_* f}{\partial z_p^2} = 0. \quad (33)$$

Hence, using the result of Corollary 2 and (33), state linearization requires $[\mathbf{\Phi}_* g, [\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]] = 0$, which implies $[g,[f,g]] = 0$. In this case, the condition (15) becomes

$$\frac{\partial^2 \mathbf{\Phi}_* f}{\partial z_p^2} = 0,$$

as stated in [15]. Note that this analysis only applies to one step in the algorithm and thus fails to show the sufficiency condition for state linearization.

Next, if the distribution $\Delta = \text{span}\{g, [f,g]\}$ is not involutive, then some approximation has to be introduced to make it linearly dependent and thus obtain the separation given by (14). This is required for the decomposition process to proceed according to the following algorithm.

Algorithm to achieve a cascade form

- **Initialization:** The system is initially in the form (8) with $p = n$.
- **Induction:** At the k^{th} iteration, a $p = (n+1-k)$-dimensional system of the form (8) is available.
  - Construct the diffeomorphism $\mathbf{\Phi}_p(\mathbf{x})$ given in (9) and choose $\gamma(\mathbf{x})$. $\mathbf{\Phi}_p(\mathbf{x})$ must satisfy the condition (10).
  - Define $\mathbf{z} \triangleq \mathbf{\Phi}_p(\mathbf{x})$ and obtain the system dynamics in the $\mathbf{z}$ coordinates as given in (11).
  - Use the projection map to define $\hat{\mathbf{x}} \triangleq Pr(\mathbf{z})$ and obtain $\dot{\hat{\mathbf{x}}}$ as shown in (12) and (13).
  - Keep the diffeomorphism $\mathbf{\Phi}_p(\mathbf{x})$ and the resulting $\mathbf{z}$-system for the backward process. The $\mathbf{z}$-system is of the form:
$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \hat{f}_z(\hat{\mathbf{x}}, z_p), \quad (34) \\ \dot{z}_p &= f_{z_p}(\hat{\mathbf{x}}, z_p) + \alpha(\hat{\mathbf{x}}, z_p)u, \quad (35) \end{aligned}$$
    where $f_{z_p}(\hat{\mathbf{x}}, z_p)$ and $\alpha(\hat{x}, z_p)$ are the p^{th} component of $f_z(\mathbf{z})$ and $g_z(\mathbf{z})$ given in (11), respectively.
  - Compute the separation given by (14). The dynamical system (14) has the same structure as the system (8), but it is of reduced order $p-1$. This becomes the input to the next iteration.
- **Termination:** The aforementioned steps are repeated until a single-dimensional system is obtained or further separation according to (14) is no longer possible.

Several remarks are in order.

Remark 1: If $\hat{f}_z(\hat{\mathbf{x}}, z_p)$ is linearly dependent on z_p, then

$$\begin{aligned} g_{\hat{x}}(\hat{x}) &= Pr(\frac{\partial \mathbf{\Phi}_* f)}{\partial z_p}) \\ &= -\frac{1}{\alpha(\mathbf{z})} Pr([\mathbf{\Phi}_* f, \mathbf{\Phi}_* g]) \text{ from (17)} \\ &= -\frac{1}{\alpha(\mathbf{z})} Pr(\mathbf{\Phi}_*[f,g]). \end{aligned}$$

This defines a direct relationship between the $g_{\hat{x}}(\hat{x})$ and $[f,g]$. By using the Lie-bracket-like operation given in Definition 3 of [11], all the higher-order Lie brackets can be obtained for the reduced system. This is where the

distinction between FBL systems and non-FBL systems is visible. Clearly, in case of non-FBL systems, $\hat{f}_z(\hat{\mathbf{x}}, z_p)$ is not linearly dependent on z_p for all iterations. That is, non-FBL systems will necessarily have an iteration where a linear relationship cannot be established. Hence, in this case, the algorithm needs to be modified as illustrated in [12] and [13].

Remark 2: This algorithm can transform the system into a chain of integrators by choosing $\gamma(\mathbf{x}) = L_f\phi_i$ such that $L_gL_f\phi_i \neq 0$ at every iteration. The existence of such a ϕ_i for FBL systems can be proved using the following lemma.

Lemma 5: If the vector fields g and $[f, g]$ are linearly independent in the neighborhood of $\mathbf{x}^0 \in \mathcal{D}$, then we can find $\phi_i(\mathbf{x})$, $i = 1, \ldots, n-1$, such that (i)

1) $\phi_i \neq 0$ and $\frac{\partial \phi_i}{\partial \mathbf{x}}$ are linearly independent rows that span the null space of $g(\mathbf{x})$, i.e. $L_g\phi_i = 0$, and
2) there exists at least one ϕ_i such that $L_gL_f\phi_i \neq 0$.

Proof: Since $span\{g\}$ is nonsingular and is an involutive distribution of dimension 1 in the neighborhood of $\mathbf{x}^0 \in \mathcal{D}$, then, by Frobenius theorem, there exist $\phi_1(\cdot), \ldots, \phi_{n-1}(\cdot)$ such that $\phi_i(\cdot) : \mathbb{R}^n \to \mathbb{R}$ and the following properties hold $\forall \mathbf{x} \in \mathcal{D}$ and for $i = 1, \ldots, n-1$:

- $\phi_i(\mathbf{x}) \neq 0$,
- $Rank \begin{pmatrix} \frac{\partial \phi_1(\mathbf{x})}{\partial \mathbf{x}} \\ \vdots \\ \frac{\partial \phi_{n-1}(\mathbf{x})}{\partial \mathbf{x}} \end{pmatrix} = n-1$,
- $L_g\phi_i(\mathbf{x}) = 0$.

The proof proceeds by contradiction. Let us assume that the second part of the lemma is not true, that is, $L_gL_f\phi_i = 0$ for all $i = 1, \ldots, n-1$. It is known that

$$L_{[f,g]}\phi_i = L_fL_g\phi_i - L_gL_f\phi_i = 0, \quad \forall \ \ 1 \leq i \leq n-1. \tag{36}$$

It follows from $L_g\phi_i = 0$ that $L_fL_g\phi_i = 0$. Since, by assumption, $L_gL_f\phi_i = 0$, it follows from (36) that $\frac{\partial \phi_i}{\partial \mathbf{x}}$ are $n-1$ linearly independent rows that also span the null space of $[f, g]$. Since the vector fields g and $[f, g]$ are linearly independent, which means that there are at most $n-2$ row vectors that span the null space of span$\{g, [f, g]\}$. However $\frac{\partial \phi_i}{\partial \mathbf{x}}$, $i = 1, \ldots n-1$, being $n-1$ linearly independent rows that span the null space of span$\{g, [f, g]\}$ contradicts the assumption that $L_gL_f\phi_i = 0$ for all $i = 1, \ldots, n-1$. Hence, there must exist at least one ϕ_i such that $L_gL_f\phi_i \neq 0$. ■

However, transforming a system to a chain of integrators may introduce undesirable singularities. In such a situation, the liberty to chose $\gamma(\mathbf{x})$ must be exploited in order to avoid these singularities.

B. Backward stage for controller design

In this section, the cascade form obtained at every step during the forward stage will be exploited to design an asymptotically stabilizing controller. Corollary 1 is applied iteratively at every step of the backward process. To illustrate the steps of the algorithm, consider a system obtained during the p^{th} step of the forward process given by (34) and (35) The algorithm proceeds as follows.

Algorithm for controller design

- **Initialization**: The algorithm starts by designing an asymptotically stabilizing controller for the smallest cascade system obtained during the last iteration of the forward process. For example, if the forward process successfully proceeds until a single-dimensional system is derived, a system of the form

$$\dot{x}_1 = f_1(x_1, x_2) \tag{37}$$

is obtained. Then, taking x_2 as the input and solving

$$-kx_1 = f_1(x_1, x_2) \tag{38}$$

for x_2 gives an asymptotically stabilizing controller for (37). This value of x_2 becomes the desired value $x_{2,d}(x_1)$.
- **Induction**: At the start of the $(p-1)^{st}$ iteration (p varies from 2 to n), the desired value $x_{p,d}(x_1, ..., x_{p-1})$ that asymptotically stabilizes the $(p-1)$-dimensional system whose state is $\hat{x} = (x_1, \ldots, x_{p-1})$ is known. Also, from the forward iteration, a p-dimensional system of the form (34) and (35) and the diffeomorphism $\mathbf{z} = \mathbf{\Phi}_p(\mathbf{x})$ is known. Corollary 1 is then used to design $x_{(p+1),d}$. The following relationships exist between the equations available in Corollary 1 and those available at the current iteration:
  - The system state $\mathbf{x}$ in (2) corresponds to $\hat{\mathbf{x}}$,
  - f_x in (2) corresponds to $\hat{f}_z(\hat{\mathbf{x}}, z_p)$ in (34),
  - ξ corresponds to z_p,
  - $\xi_d(\mathbf{x})$ corresponds to $x_{p,d}(x_1, ..., x_{p-1})$,
  - $f_\xi(\mathbf{x}, \xi)$ in (3) corresponds to $f_{z_p}(\hat{\mathbf{x}}, z_p)$ from (35),
  - $g(\mathbf{x}, \xi)$ in (3) corresponds to $\alpha(\hat{x}, z_p)$ from (35), and
  - u corresponds to x_{p+1}.

  Finally, (4) is used to obtain $x_{(p+1),d}$. The change of coordinates $\mathbf{z} = \mathbf{\Phi}_p(\mathbf{x})$ is used to change the coordinates of $x_{(p+1),d}$ from $\mathbf{z}$ to $\mathbf{x}$. Then, $x_{(p+1)d}$ can be used as the input in the next iteration.
- **Last iteration**: In the last iteration (iteration number $n-1$), x_{n+1} becomes the sought input u for the system (8).

IV. Example of a DC Motor

The example illustrates how the possibility of choosing $\gamma(\mathbf{x})$ helps avoid the singularity that arises due to the particular choice of $\gamma(\mathbf{x})$ required for feedback linearization. This section presents an application of the algorithm to a FBL system. However, the algorithm is not restricted to FBL systems, and application of the quotient method to non-FBL system are illustrated in [12] and [13].

A field-controlled DC motor with negligible shaft damping is considered in [1]. The system can be described by

$$\begin{aligned} v_f &= R_f i_f + L_f \frac{di_f}{dt}, \\ v_a &= c_1 i_f \omega + L_a \frac{di_a}{dt} + R_a i_a, \\ J\frac{d\omega}{dt} &= c_2 i_f i_a. \end{aligned}$$

The first equation represents the field circuit, with v_f, i_f, R_f, and L_f being the voltage, current, resistance and inductance, respectively. The variables v_a, i_a, R_a, and L_a are the corresponding variables for the armature circuit described by the second equation. The term $c_1 i_f \omega$ is the back e.m.f. induced in the armature circuit. The third equation is the equation of motion for the shaft, with the rotor inertia J and the torque $c_2 i_f i_a$ produced by the interaction of the armature current with the field circuit flux. The voltage v_a is held constant, and control is achieved by varying v_f. The system is represented by the third-order model

$$\dot{\mathbf{x}} = f(\mathbf{x}) + gu,$$

with states $x_1 = i_f, x_2 = i_a, x_3 = \omega$, the input $u = \frac{v_f}{L_f}$,

$$f(\mathbf{x}) = \begin{bmatrix} -ax_1 \\ -bx_2 + \rho - cx_1x_3 \\ \theta x_1 x_2 \end{bmatrix}, \qquad g = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

and positive constants $a = R_f, b = R_a/L_a, c = c_1/L_a, \theta = c_2/J, \rho = V_a/L_a$. The open-loop system has its equilibrium at $x_1 = 0$ and $x_2 = \rho/b$. The aim is to design a controller that drives the system from any initial condition to the desired operating point $x^* = (0, \rho/b, \omega_0)$, where ω_0 is the desired set point for the angular velocity x_3.
The forward stage gives the simple cascade structure,

$$\begin{aligned} \dot{y_1} &= -y_2\theta(by_2 + \rho), \\ \dot{y_2} &= -by_2 + g_1(y_1, y_2)z_3, \\ \dot{z_3} &= -az_3 + u, \end{aligned} \tag{39}$$

where

$$g_1(y_1, y_2) = -(c^2\omega_0^2 + 2cy_1 - 2\theta c\rho y_2 - bc\theta y_2^2)^{\frac{1}{2}}$$

with the diffeomorphism

$$\mathbf{\Phi} = \begin{pmatrix} \frac{-\theta\rho^2 - b^2c\omega_0^2 + b^2\theta x_2^2 + b^2cx_3^2}{2b^2} \\ x_2 - \rho/b \\ x_1 \end{pmatrix}. \tag{40}$$

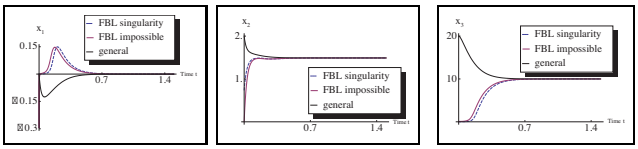
The first function in this diffeomorphism is always the static feedback linearizing output function (Propositions 3 and 4 of [11]). The second stage computes an asymptotic control for (39) given by

$$\begin{aligned} u = -k_3(y_3 - y_{3,d}) + \frac{\partial y_{3,d}}{\partial y_1}(-y_2\theta(by_2 + \rho)) \\ + \frac{\partial y_{3,d}}{\partial y_2}(-by_2 + g_1(y_1, y_2)z_3) + az_3, \end{aligned} \tag{41}$$

where

$$\begin{aligned} y_{3,d} &= \frac{-k_2(y_2 - y_{2,d}) + by_2 + \frac{\partial y_{2,d}}{\partial y_1}(-y_2\theta(by_2 + \rho))}{g_1(y_1, y_2)}, \\ y_{2,d} &= \frac{-\theta\rho + \sqrt{\theta^2\rho^2 + 4\theta bk_1y_1}}{2\theta b}. \end{aligned}$$

It is possible to obtain a chain of integrators by choosing $\gamma(\mathbf{x}) = L_f\phi_i$ such that $L_gL_f\phi_i \neq 0$ at every iteration. The existence of such a ϕ_i is guaranteed through Lemma 5, which results in the unique diffeomorphism [1]:



(a) x_1 (b) x_2 (c) x_3

Fig. 3. DC motor controlled with the quotient method. The state behavior is depicted for three different initial conditions. The thin line represents a general case. The dashed line corresponds to a point of singularity for feedback linearization. The thick line represents an initial condition that is impossible to control using feedback linearization.

$$\Phi_{FBL} = \begin{pmatrix} \theta x_2^2 + cx_3^2 - \theta(\rho/b)^2 - c\omega_0^2 \\ 2\theta x_2(\rho - bx_2) \\ -2\theta(\rho - 2bx_2)(-bx_2 + \rho - cx_1x_3) \end{pmatrix}.$$

However, this transformation introduces a singularity at $x_2 = \rho/2b$ in addition to the system singularity at $x_3 = 0$. In contrast, the quotient approach provides a degree of freedom through the choice of $\gamma(\mathbf{x})$ at every iteration. This degree of freedom allows circumventing this singularity by using the diffeomorphism given in (40). This fact is clearly seen in the simulation results presented in Figure 3. The simulations are carried out using the parameters of a DC motor [16] with three different initial conditions $(0, \rho/(2b), 0.01$; *FBL singularity*), $(0, 0, 0.1$; *FBL impossible*) and $(0, 2, 20$; *general*). The first (FBL singularity) and second (FBL impossible) initial conditions are outside the domain of attraction of any controller designed using feedback linearization [1] due to the presence of singularity at $x_2 = \rho/2b$. The controller designed using feedback linearization works only for $x_2 > \rho/2b$, whereas the controller designed using the quotient method does not have any such restriction. Hence, upon using the quotient method, a larger domain of attraction can be achieved.

V. Possibility of using Backstepping

Backstepping requires the system to be in strict-feedback form. Strict-feedback form can be obtained from the forward stage of the algorithm by constructing the diffeomorphism

$$\mathbf{\Phi}(\mathbf{x}) = \mathbf{\Phi}_2(\mathbf{x}) \perp \mathbf{\Phi}_3(\mathbf{x}) \perp \ldots \perp \mathbf{\Phi}_{n-1}(\mathbf{x}) \perp \mathbf{\Phi}_n(\mathbf{x}), \tag{42}$$

where $\perp$ is an operation defined as

$$\begin{pmatrix} \phi_{L_1}(\mathbf{x}) \\ \vdots \\ \phi_{L_p}(\mathbf{x}) \end{pmatrix} \perp \begin{pmatrix} \phi_{R_1}(\mathbf{x}) \\ \vdots \\ \phi_{R_p}(\mathbf{x}) \\ \phi_{R_{(p+1)}}(\mathbf{x}) \\ \vdots \\ \phi_{R_k}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \phi_{L_1}(\mathbf{z}) \\ \vdots \\ \phi_{L_{(p)}}(\mathbf{z}) \\ \phi_{R_{(p+1)}(\mathbf{x})} \\ \vdots \\ \phi_{R_k}(\mathbf{x}) \end{pmatrix},$$

and

$$\mathbf{z} = \begin{pmatrix} \phi_{R_1}(\mathbf{x}) \\ \vdots \\ \phi_{R_p}(\mathbf{x}) \end{pmatrix}.$$

However, in order to apply backstepping to a strict-feedback form and achieve global stability, it is essential that $\mathbf{\Phi}(\mathbf{x})$ in (42) be a global diffeomorphism. Moreover, in order to avoid singularity, it is necessary that, in (35), $\alpha(\hat{x}, z_p) \neq 0$ globally. If both conditions are satisfied, a guaranteed globally stabilizing controller can be obtained by replacing the controller design stage by backstepping.

The difference in the two approaches is illustrated by straightforward application of both methods on the following system:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\xi, \quad (43)$$

$$\dot{\xi} = u. \quad (44)$$

Let us assume that there exists a $\xi_d(\mathbf{x})$ that stabilizes (43). This means that there exists a corresponding Lyapunov function $V(\mathbf{x})$ such that

$$\dot{V}(\mathbf{x}) = \frac{\partial V(\mathbf{x})}{\partial \mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x})\xi_d(\mathbf{x})) = -W(\mathbf{x}),$$

where $W(\mathbf{x})$ is a positive definite function. Now, by creating a new Lyapunov function $V_\xi = V(\mathbf{x}) + \frac{1}{2}(\xi - \xi_d(\mathbf{x}))^2$ and assigning

$$\dot{V}_\xi = -W(\mathbf{x}) - k(\xi - \xi_d(\mathbf{x}))^2,$$

a backstepping control law can be computed [4]:

$$u = -k(\xi - \xi_d(\mathbf{x})) + \frac{\partial \xi_d(\mathbf{x})}{\partial \mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x})\xi) - \frac{\partial V(\mathbf{x})}{\partial \mathbf{x}} g(\mathbf{x}). \quad (45)$$

On the other hand, by defining $e = \xi - \xi_d(\mathbf{x})$ and assigning $\dot{e} = -ke$, yields the control law as:

$$u = -k(\xi - \xi_d(\mathbf{x})) + \frac{\partial \xi_d(\mathbf{x})}{\partial \mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x})\xi). \quad (46)$$

Upon comparing the two control laws, one notices the absence of the term $-\frac{\partial V}{\partial \mathbf{x}}(\mathbf{x})g(\mathbf{x})$ in (46). When larger systems are dealt with, this term would get accumulated at every stage and might turn out to be very complex. However, depending on the system, there might exist Lyapunov functions for which either the use of this term can be avoided or the term can be used so as to cancel other larger ones. In the case of non-FBL systems, since a strict feedback form could be obtained only through approximation, using backstepping does not result in a globally stabilizing controller. Hence, when the backstepping method fails to achieve global stability, the method proposed in this paper might be more appropriate because of the reduced complexity implied by (46) as compared with (45).

VI. Conclusions

A method for constructing stabilizing controllers for single-input FBL systems has been proposed. The algorithm is based on an iterative decomposition of the original system into cascade form. The idea is to isolate the effect of the control input on one state and then use that state as the control input for the remaining states. This paper has provided additional results required to incorporate the control design in the algorithm presented in [11].

The control law is designed iteratively based on the cascade forms obtained at every iteration of the forward process. The control law given in Corollary 1 is used iteratively to construct the controller. The structure needed for Corollary 1 can only be guaranteed for feedback-linearizable systems. This is a consequence of Lemma 2 and Lemma 3. However, even if the distribution generated by the original system is not involutive, the approach can still be applied, as illustrated in [12] and [13].

The backward stage could be replaced by backstepping, which is the method of choice when $\Phi(\mathbf{x})$ in (42) is a global diffeomorphism and $\alpha(\hat{x}, z_p) \neq 0$ globally at each stage. In all other cases, the backward stage is advantageous over backstepping. The main advantage stems from the absence of the term $-\frac{\partial V}{\partial \mathbf{x}}(\mathbf{x})g(\mathbf{x})$, which simplifies things considerably, especially for large systems. It has also been shown through the example of a field-controlled DC motor (a FBL system) that the quotient approach intelligently avoids the singularity introduced during feedback linearization by properly choosing $\gamma(\mathbf{x})$ at every stage.

Acknowledgments: Support by the Swiss National Science Foundation under Grant **FN 200021-126916/1** is gratefully acknowledged.

References

[1] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Englewood Cliffs, N.J., 3rd edition, 2002.

[2] A. Isidori. *Nonlinear Control Systems*. Springer Verlag, Berlin, Heidelberg, New York, second edition, 1989.

[3] H. Nijmeijer and A. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer Verlag, New York, 1990.

[4] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, Inc., New York, NY, USA, 1995.

[5] R. Ortega, A.J. Van Der Schaft, I. Mareels, and B. Maschke. Putting energy back in control. *Control Systems Magazine, IEEE*, 21(2):18–33, Apr 2001.

[6] V. I. Utkin. *Sliding Modes in Control Optimization*. Springer Verlag, Moscow, 1992.

[7] P. A. Businger. Reducing a matrix to Hessenberg form. *Math. Comp.*, 23:819–821, 1969.

[8] G. S. Miminis and C. C. Paige. An algorithm for pole assignment of time invariant linear systems. *Int. J. Control*, 35(2):341–354, 1982.

[9] C. C. Paige. Properties of numerical algorithms related to computing controllability. *IEEE Trans. on Automatic Control*, 26(1):130–138, February 1981.

[10] P. M. Van Dooren. The generalized eigenstructure problem in linear system theory. *IEEE Trans. on Automatic Control*, AC-26(1):111–129, 1981.

[11] Ph. Mullhaupt. Quotient submanifolds for static feedback linearization. *Systems & Control Letters*, 55:549–557, 2006.

[12] S. S. Willson, P. Mullhaupt, and D. Bonvin. Quotient method for controlling the acrobot. *48th IEEE Conference on Decision and Control*, 2009.

[13] D. Ingram, S. S. Willson, P. Mullhaupt, and D. Bonvin. Stabilization of the cart-pendulum system through approximate manifold decomposition. *18th IFAC World Congress*, 2011.

[14] I. A. Tall. State linearization of control systems: An explicit algorithm. *48th IEEE Conference on Decision and Control*, 2:7448–7453, 2009.

[15] I. A. Tall. State and feedback linearization of single-input control systems. *Systems & control letters*, 59:429–441, 2010.

[16] Yung-chii Liang and V.J. Gosbell. DC machine models for spice2 simulation. *IEEE Trans. on Power Electronics*, 5(1):16–20, January 1990.