2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)

On the Computation of Differentially Flat Inputs

Klemens Fritzsche
Institute of Control Theory
Faculty of Electrical and Computer Engineering
Dresden University of Technology, Germany
klemens.fritzsche@tu-dresden.de

Klaus Röbenack
Institute of Control Theory
Faculty of Electrical and Computer Engineering
Dresden University of Technology, Germany
klaus.roebenack@tu-dresden.de

***Abstract*—Differential flatness and methods based on this property have proved to be very fruitful for tracking controllers and trajectory planning of nonlinear systems. As a dual to the concept of differentially flat outputs, this contribution deals with the computation of flat inputs, and thus with the problem of actuator placement. Given a mathematical description of a system's behavior and a desired output, we propose an algorithm that constructs such inputs for observable systems. We show that there are no integrability problems which are typical for flat output computations, and exemplify this by an example. For non-observable systems, we show that an additional constraint needs to be fulfilled. It is not obvious how to incorporate this into an algorithm.**

***Index Terms*—Nonlinear systems, Differential flatness, Flat inputs, Actuator placement, Unimodular completion, Algorithms.**

I. Introduction

Introduced in the early 1990s [1], [2], the property of differential flatness has gained a lot of attraction in the design of tracking controllers and trajectory planning for nonlinear control systems [3]–[5]. Despite much progress [6]–[13], to date no algorithm can determine whether or not an arbitrary dynamical system is flat, and if so, how to determine the so-called flat output [14], [15]. Although numerous practical examples have been shown to be flat, the scientific demand for a necessary and sufficient flatness condition remains.

There exist mainly two different approaches for systematic flatness analysis, one of which uses nonlinear transformations with the goal to arrive at a representation where the flatness property can be read off [8], [16]. Another very popular approach for flat output computation involves the calculation of unimodular completions of the linearization along an arbitrary trajectory of the system [6], [7], [9]–[13], [17]. In addition, for differential flatness an integrability condition is required to be satisfied.

Investigating the dual perspective of flat outputs, differentially flat inputs were introduced [18], [19], and can be interpreted as an actuator placement problem such that a given system with output becomes flat [20], [21]. In this regard, the design process of dynamical systems can directly benefit from flat input computations. In addition, observer design methods have used the same concept [22], the computation of differential parametrizations can be systemized and thus allow feedforward control and asymptotic tracking of nonlinear systems [21], [23], [24]. Flat inputs have been used in parameter identification [25], [26], and even in problems, such as secure communication [20].

We will render the problem of flat input computation in the unimodularity context and show, that for flat input computation of observable nonlinear systems, the integrability condition is always satisfied. In addition, we propose an algorithm for flat input computation of these systems.

It has been shown that flat inputs may exist for non-observable systems [19]. While a general necessary and sufficient condition for the existence of flat inputs in the non-observable case remains an open problem, this condition has been found for systems whose output has two components [20]. We will reformulate the general non-observable case and investigate sufficient conditions for flat inputs.

II. Preliminaries and problem statement

The field of meromorphic functions in t will be denoted by $\mathfrak{K}$, and – for the sake of readability – the differential operator $\frac{\mathrm{d}}{\mathrm{d}t}$ by λ. Further, for suitable matrices $\mathbf{A}_i \in \mathfrak{K}^{m\times n}$ we set

$$\operatorname{row}(\mathbf{A}_1, \dots, \mathbf{A}_q) := \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_q \end{pmatrix}. \tag{1}$$

Definition 1. *Let $\mathbf{A} \in \mathfrak{K}^{m\times n}$. A matrix $\mathbf{A}^{+\mathsf{R}} \in \mathfrak{K}^{n\times m}$ is called **right pseudo inverse** of $\mathbf{A}$, if $\mathbf{A}\mathbf{A}^{+\mathsf{R}} = \mathbf{I}_m$ holds. In the same manner, $\mathbf{A}^{+\mathsf{L}} \in \mathfrak{K}^{n\times m}$ is called **left pseudo inverse** of $\mathbf{A}$, if $\mathbf{A}^{+\mathsf{L}}\mathbf{A} = \mathbf{I}_n$ holds.*

Definition 2. *Let $\mathbf{A} \in \mathfrak{K}^{m\times n}$ with $\operatorname{rank}\mathbf{A} = r \le m \le n$. A matrix $\mathbf{A}^{\perp\mathsf{R}} \in \mathfrak{K}^{n\times(n-r)}$ is called **right orthogonal complement**, if*

$$\mathbf{A}\mathbf{A}^{\perp\mathsf{R}} = \mathbf{0}_{m,n-r}. \tag{2}$$

*Similarly, let $\mathbf{A} \in \mathfrak{K}^{m\times n}$ with $\operatorname{rank}\mathbf{A} = r \le n \le m$. A matrix $\mathbf{A}^{\perp\mathsf{L}} \in \mathfrak{K}^{(m-r)\times m}$ is called **left orthogonal complement**, if*

$$\mathbf{A}^{\perp\mathsf{L}}\mathbf{A} = \mathbf{0}_{m-r,n}. \tag{3}$$

The set of polynomials in λ with coefficients in $\mathfrak{K}$ constitutes a ring, the so-called Ore polynomial ring which we will denote by $\mathfrak{K}[\lambda]$. The multiplication of elements of $\mathfrak{K}[\lambda]$ is non-commutative and determined by the rule

$$\forall a \in \mathfrak{K} : \lambda a = \dot{a} + a\lambda. \tag{4}$$

In this contribution, we will work with elements from the set $\mathfrak{K}^{m\times n}[\lambda] = (\mathfrak{K}[\lambda])^{m\times n}$, i.e., $m\times n$ matrices with elements in $\mathfrak{K}[\lambda]$.

Definition 3. *A polynomial matrix $\mathbf{A}(\lambda) \in \mathfrak{K}^{n\times n}[\lambda]$ is called **unimodular**, if its inverse $\mathbf{A}^{-1}(\lambda)$ is an element from $\mathfrak{K}^{n\times n}[\lambda]$. We denote the set of unimodular $n \times n$ matrices with $\mathfrak{U}_n[\lambda]$.*

Definition 4. *Let $\mathbf{A}(\lambda) \in \mathfrak{K}^{m\times n}[\lambda]$. The following holds: $\exists \mathbf{L}(\lambda) \in \mathfrak{U}_m[\lambda], \mathbf{R}(\lambda) \in \mathfrak{U}_n[\lambda]$:*

$$\mathbf{L}(\lambda)\mathbf{A}(\lambda)\mathbf{R}(\lambda) = \begin{pmatrix} \boldsymbol{\Delta}(\lambda) & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(n-r)} \end{pmatrix}, \quad (5)$$

*where $\boldsymbol{\Delta}(\lambda) \in \mathfrak{K}^{r\times r}[\lambda]$ with $r \leq \min(m,n)$ denotes a diagonal matrix and the right hand side of (5) is called the **Smith normal form** of $\mathbf{A}(\lambda)$.*

Definition 5. *Let $\mathbf{A}(\lambda) \in \mathfrak{K}^{m\times n}[\lambda]$. $\mathbf{A}(\lambda)$ is called **hyper-regular**, if its Smith normal form yields $\boldsymbol{\Delta}(\lambda) = \mathbf{I}_{\min(m,n)}$.*

Definition 6. *Let*

$$\mathbf{0} = \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, \dots, \mathbf{x}^{(\alpha)}) \quad (6)$$

with $\mathbf{x}(t) \in \mathbb{R}^n$ be an implicit equation. The matrix

$$\mathbf{P}(\lambda) = \left(\sum_{i=0}^{\alpha} \frac{\partial \mathbf{F}}{\partial \mathbf{x}^{(i)}} \lambda^i \right) \quad (7)$$

*is called **generalized Jacobian** or **tangent matrix** of $\mathbf{F}$.*

Definition 7. *Let the behavior of a dynamical system be described by the implicit underdetermined equation*

$$\mathbf{0}_{n-m} = \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}), \quad \mathbf{x}(t) \in \mathbb{R}^n. \quad (8)$$

The system (8) is called (differentially) flat *if there exists an $m-$tuple $\mathbf{y}$ such that*

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \dots, \mathbf{x}^{(\alpha)}) \quad (9)$$
$$\mathbf{x} = \mathbf{g}(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^{(\beta)}) \quad (10)$$

with $\alpha, \beta < \infty$. The tuple $\mathbf{y}$ is then called flat output.

If the dynamical behavior can be manipulated by an input $\mathbf{u}(t) \in \mathbb{R}^m$, we get implicit underdetermined equations of the form (8) by either eliminating $\mathbf{u}$, or by introducing a new state vector consisting of the old state components and the input.

There exist necessary and sufficient conditions for differential flatness [7]. The following condition will be of use in this contribution.

Proposition 1. *The dynamical system with the behavior in implicit form*

$$\mathbf{0}_{n-m} = \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}), \quad \mathbf{x}(t) \in \mathbb{R}^n \quad (11)$$

is flat, if there exists an output

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \dots, \mathbf{x}^{(\alpha)}), \quad \mathbf{y}(t) \in \mathbb{R}^m \quad (12)$$

such that

$$\begin{pmatrix} \mathbf{P}(\lambda) \\ \mathbf{H}(\lambda) \end{pmatrix} := \begin{pmatrix} \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \\ \sum_{i=0}^{\alpha} \frac{\partial \mathbf{h}}{\partial \mathbf{x}^{(i)}}\lambda^i \end{pmatrix} \in \mathfrak{U}_n[\lambda] \quad (13)$$

and

$$\mathrm{d}\left(\mathbf{H}(\lambda)\mathrm{d}\mathbf{x}\right) = \mathbf{0}. \quad (14)$$

Proof. See [7, Theorem. 3]. □

For flat output computation, a necessary condition for Proposition 1 to be applicable, is hyper-regularity of $\mathbf{P}(\lambda)$, which can be interpreted as local controllability [6].

Instead of searching for a flat output, the flatness property may be accounted for in the design process of the system, by asking *where* to influence the behavior of the system. That is, given equations for the behavior of an autonomous first order system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(t) \in \mathbb{R}^n \quad (15)$$

and an output

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \dots, \mathbf{x}^{(\alpha)}), \quad \mathbf{y}(t) \in \mathbb{R}^m, \quad (16)$$

we compute an input-dependent system of the form

$$\dot{\mathbf{x}} = \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u}) \quad (17)$$

such that the given output $\mathbf{y}$ becomes a flat output. The implicit form of (17) reads

$$\mathbf{0} = \hat{\mathbf{F}}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{x}}) := \dot{\mathbf{x}} - \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u}), \quad (18)$$

and the generalized Jacobian of (18) w.r.t row$(\mathbf{x}, \mathbf{u})$ appended by the generalized Jacobian of (16) yields

$$\begin{pmatrix} \hat{\mathbf{P}}(\lambda) \\ \mathbf{H}(\lambda) \end{pmatrix} =: \begin{pmatrix} \frac{\partial \hat{\mathbf{F}}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{x}} & \frac{\partial \hat{\mathbf{F}}}{\partial \mathbf{u}} \\ \sum_{i=0}^{\alpha} \frac{\partial \mathbf{h}}{\partial \mathbf{x}^{(i)}}\lambda^i & \mathbf{0}_{m,m} \end{pmatrix} \quad (19)$$

$$= \begin{pmatrix} \mathbf{I}_n\lambda - \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{x}} & -\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{u}} \\ \sum_{i=0}^{\alpha} \frac{\partial \mathbf{h}}{\partial \mathbf{x}^{(i)}}\lambda^i & \mathbf{0}_{m,m} \end{pmatrix}. \quad (20)$$

Due to Proposition 1, for differential flatness we need to ensure

$$\begin{pmatrix} \hat{\mathbf{P}}(\lambda) \\ \mathbf{H}(\lambda) \end{pmatrix} \in \mathfrak{U}_{n+m}[\lambda] \quad (21)$$

as well as

$$\mathrm{d}\left(\hat{\mathbf{P}}(\lambda)\begin{pmatrix}\mathrm{d}\mathbf{x}\\ \mathrm{d}\mathbf{u}\end{pmatrix}\right) = \mathbf{0}. \quad (22)$$

If both these conditions are satisfied for the input-injected system (17), then the output $\mathbf{y}$ is flat, and $\mathbf{u}$ is called **flat input**.

Usually, flat input computation does not take into account input injection in the output equation. In this contribution, we will require the same and can therefore deduce the following.

Corollary 1. *A necessary condition for the existence of flat inputs is hyper-regularity of the generalized Jacobian of the output equation.*

Proof. Applicability of Proposition 1 involves the unimodularity condition in (13) to be satisfied. An arbitrary hyper-row of a unimodular matrix is hyper-regular. A hyper-regular matrix remains hyper-regular when appending zero columns. If we do not allow flat input injection in the output equation, then the generalized Jacobian of the output is only affected by appending zero columns when injecting a flat input. Therefore, the unimodularity condition of (1) requires the generalized Jacobian of the output equation to be hyper-regular. □

III. Flat input computation for observable systems

The generalized Jacobian of the autonomous system with output of the form

$$\mathbf{F}(\mathbf{x},\dot{\mathbf{x}}) = \dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}) = \mathbf{0}_n \tag{23}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x},\dot{\mathbf{x}},\dots,\mathbf{x}^{(\alpha)}) \tag{24}$$

reads

$$\mathbf{Z}(\lambda) := \sum_{i=0}^{\gamma} \mathbf{Z}_i \lambda^i := \begin{pmatrix} \mathbf{I}_n\lambda - \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \\ \sum_{i=0}^{\alpha} \frac{\partial \mathbf{h}}{\partial \mathbf{x}^{(i)}}\lambda^i \end{pmatrix} \tag{25}$$

with $\mathbf{Z}(\lambda) \in \mathfrak{K}^{(n+m)\times n}[\lambda]$ and $\gamma = \max(1,\alpha)$. Dual to controllability, observability is equivalent to hyper-regularity of $\mathbf{Z}(\lambda)$. Checking hyper-regularity of Ore polynomial matrices with coefficients in $\mathfrak{K}$ can be evaluated symbolically using the Smith normal form, methods based on row and column reduction [27]–[29], or even numerically [30].

The following algorithm proposes a method for the computation of unimodular completions for hyper-regular $(n+m)\times n$ matrices with $\gamma = 1$. For the completion of a hyper-regular matrix of the transposed dimensions with $\gamma > 1$, using integrator chains we can reduce the order of λ to 1 by raising the dimension of the problem. This approach cannot offhandedly be translated here, but since the output components are usually chosen to be physically meaningful, the order of time-derivatives in the output equation should generally be low and restricting to $\gamma \leq 1$ may not be a practical constrain.

Remark 1. *Lie-derivatives may be used to lower the order of time derivatives in the output equation as follows.*

In state space representation, with $\boldsymbol{\varepsilon}(\mathbf{x}) := \mathrm{id}(\mathbf{x})$ *we find*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \frac{\partial \boldsymbol{\varepsilon}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) = L_{\mathbf{f}}\boldsymbol{\varepsilon}(\mathbf{x}).$$

Given $\mathbf{x}^{(k)} = L_{\mathbf{f}}^k\boldsymbol{\varepsilon}(\mathbf{x})$, *by induction we conclude*

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \frac{\partial L_{\mathbf{f}}^k\boldsymbol{\varepsilon}(\mathbf{x})}{\partial t} = \frac{\partial L_{\mathbf{f}}^k\boldsymbol{\varepsilon}(\mathbf{x})}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial t} = \frac{\partial L_{\mathbf{f}}^k\boldsymbol{\varepsilon}(\mathbf{x})}{\partial \mathbf{x}}\mathbf{f}(\mathbf{x}) \\ &= L_{\mathbf{f}}L_{\mathbf{f}}^k\boldsymbol{\varepsilon}(\mathbf{x}) = L_{\mathbf{f}}^{k+1}\boldsymbol{\varepsilon}(\mathbf{x}).\end{aligned} \tag{26}$$

Therefore, an arbitrary output $\mathbf{y} = \mathbf{h}(\mathbf{x},\dot{\mathbf{x}},\dots,\mathbf{x}^{(\alpha)})$ *with* $\alpha < \infty$ *can be represented as*

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, L_{\mathbf{f}}\boldsymbol{\varepsilon}(\mathbf{x}), L_{\mathbf{f}}^2\boldsymbol{\varepsilon}(\mathbf{x}),\dots,L_{\mathbf{f}}^{\alpha}\boldsymbol{\varepsilon}(\mathbf{x})) =: \tilde{\mathbf{h}}(\mathbf{x}),$$

that is, we can always find an equivalent output equation independent of time-derivatives of $\mathbf{x}$.

The following steps describe an iterative algorithm, which is strongly inspired by the algorithm for unimodular row completion described in [10]–[12] for matrices from $\mathfrak{K}^{(n-m)\times n}[\lambda]$ and can thus be seen as the dual version. Due to non-commutativity of $\mathfrak{K}[\lambda]$, in general

$$\mathbf{A}(\lambda) \text{ hyper-regular } \not\Rightarrow \mathbf{A}^{\mathsf{T}}(\lambda) \text{ hyper-regular}$$

holds [30]. This is why we cannot simply complete by taking the transposed matrix and apply the algorithms from [10]–[12], and eventually, why both algorithms differ in some details (whereas the idea remains the same).

A. Algorithm

We will assume $\mathbf{Z}(\lambda) = \mathbf{Z}_1\lambda + \mathbf{Z}_0 \in \mathfrak{K}^{p\times n}[\lambda]$ with $\operatorname{rank}\mathbf{Z}_1 = n$ where $p = n+m$.

First, we set $\mathbf{Z}_{1,[0]} := \mathbf{Z}_1$ and $\mathbf{Z}_{0,[0]} := \mathbf{Z}_0$, where the subscript $[i]$ denotes the iteration cycle. We start with $i = 0$ from the equation

$$\mathbf{0} = \mathbf{v}_{[i]}\left(\mathbf{Z}_{1,[i]}\lambda + \mathbf{Z}_{0,[i]}\right). \tag{27}$$

1) Reduction: We have

$$\mathbf{Z}_{[i]} = \mathbf{Z}_{1,[i]}\lambda + \mathbf{Z}_{0,[i]} \in \mathfrak{K}^{p_{[i]}\times n_{[i]}}[\lambda]$$

with $\operatorname{rank}\mathbf{Z}_{1,[i]} = n_{[i]}$ and $p_{[i]} = n_{[i]} + m_{[i]}$. First, we compute a transformation

$$\mathbf{v}_{[i]} = \left(\mathbf{v}_{[i+1]}, \mathbf{w}_{[i+1]}\right)\begin{pmatrix}\mathbf{Z}_{1,[i]}^{+\mathsf{L}} \\ \mathbf{Z}_{1,[i]}^{\perp\mathsf{L}}\end{pmatrix} \tag{28}$$

which results in

$$\begin{aligned}\mathbf{0} &= \left(\mathbf{v}_{[i+1]}, \mathbf{w}_{[i+1]}\right)\begin{pmatrix}\mathbf{Z}_{1,[i]}^{+\mathsf{L}}\mathbf{Z}_{1,[i]}\lambda + \mathbf{Z}_{1,[i]}^{+\mathsf{L}}\mathbf{Z}_{0,[i]} \\ \mathbf{Z}_{1,[i]}^{\perp\mathsf{L}}\mathbf{Z}_{1,[i]}\lambda + \mathbf{Z}_{1,[i]}^{\perp\mathsf{L}}\mathbf{Z}_{0,[i]}\end{pmatrix} &(29)\\ &= \left(\mathbf{v}_{[i+1]}, \mathbf{w}_{[i+1]}\right)\begin{pmatrix}\mathbf{I}_{n_{[i]}}\lambda + \mathbf{C}_{[i]} \\ \mathbf{B}_{[i]}\end{pmatrix} &(30)\end{aligned}$$

with $\mathbf{B}_{[i]} := \mathbf{Z}_{1,[i]}^{\perp\mathsf{L}}\mathbf{Z}_{0,[i]} \in \mathfrak{K}^{m_{[i]}\times n_{[i]}}$ and $\mathbf{C}_{[i]} := \mathbf{Z}_{1,[i]}^{+\mathsf{L}}\mathbf{Z}_{0,[i]}$. The components $\mathbf{w}_{[i+1]}$ in (30) are purely algebraic, i.e., this step reduces the number of derived components.

If $\mathbf{B}_{[i]} = \mathbf{0}$, there exists a non-observable subsystem, and the matrix $\mathbf{Z}(\lambda)$ cannot be completed because it is not hyper-regular.

If $\operatorname{rank}\mathbf{B}_{[i]} = m_{[i]}$, then the following step (null space decomposition) can be skipped.

2) Null space decomposition: If $\operatorname{rank}\mathbf{B}_{[i]} < m_{[i]}$ in (30), then we adjust the transformation as follows:

$$\mathbf{v}_{[i]} = \left(\mathbf{v}_{[i+1]}, \mathbf{w}_{[i+1]}, \mathbf{z}_{[i+1]}\right)\begin{pmatrix}\mathbf{Z}_{1,[i]}^{+\mathsf{L}} \\ \tilde{\mathbf{Z}}_{1,[i]}^{\perp\mathsf{L}} \\ \mathbf{G}_{[i]}\end{pmatrix} \tag{31}$$

such that $\tilde{\mathbf{B}}_{[i]} = \tilde{\mathbf{Z}}_{1,[i]}^{\perp\mathsf{L}}\mathbf{Z}_{0,[i]}$ has full row rank, and $\mathbf{G}_{[i]}\mathbf{Z}_{0,[i]} = \mathbf{0}$. From both these conditions we can conclude

$$\mathbf{G}_{[i]} = \mathbf{B}_{1,[i]}^{\perp\mathsf{L}}\mathbf{Z}_{1,[i]}^{\perp\mathsf{L}} \tag{32}$$

$$\tilde{\mathbf{Z}}_{1,[i]}^{\perp\mathsf{L}} = \left((\mathbf{B}_{1,[i]}^{\perp\mathsf{L}})^{\perp\mathsf{R}}\right)^{\mathsf{T}}\mathbf{Z}_{1,[i]}^{\perp\mathsf{L}}. \tag{33}$$

The components $\mathbf{z}_{[i+1]}$ are a part of the flat input of the tangent system. We finish this step by replacing $\mathbf{B}_{[i]}$ with $\tilde{\mathbf{B}}_{[i]}$.

3) Elimination: The algorithm terminates in the k-th iteration, if $\mathbf{B}_{[k]}$ has full column rank.

Otherwise, in this step we eliminate the algebraic equations. The matrix $\mathbf{B}_{[i]}$ has full row rank, i.e., there exists a right orthogonal complement which we can multiply (30) with from the right. This yields

$$\mathbf{0} = (\mathbf{v}_{[i+1]}, \mathbf{w}_{[i+1]}) \begin{pmatrix} \mathbf{I}\lambda + \mathbf{C}_{[i]} \\ \mathbf{B}_{[i]} \end{pmatrix} \mathbf{B}_{[i]}^{\perp\mathsf{R}} \tag{34}$$

$$= (\mathbf{v}_{[i+1]}, \mathbf{w}_{[i+1]}) \begin{pmatrix} \lambda\mathbf{B}_{[i]}^{\perp\mathsf{R}} + \mathbf{C}_{[i]}\mathbf{B}_{[i]}^{\perp\mathsf{R}} \\ \mathbf{0} \end{pmatrix} \tag{35}$$

$$= \mathbf{v}_{[i+1]} \left(\mathbf{B}_{[i]}^{\perp\mathsf{R}}\lambda + \mathbf{C}_{[i]}\mathbf{B}_{[i]}^{\perp\mathsf{R}} + \frac{\partial}{\partial t}(\mathbf{B}_{[i]}^{\perp\mathsf{R}}) \right) \tag{36}$$

$$=: \mathbf{v}_{[i+1]} \left(\mathbf{Z}_{1,[i+1]}\lambda + \mathbf{Z}_{0,[i+1]} \right). \tag{37}$$

We obtain an equation of the same structure as in (27) and continue applying the preceding steps.

B. Computing a column completion

After the above steps terminate in the k-th iteration, we are able to compute the unimodular completion of $\mathbf{Z}(\lambda)$ by the relationship $\mathbf{v}_{[i]}\mathbf{Z}_{1,[i]} = \mathbf{v}_{[i+1]}$ which we get from (28) by right multiplication with $\mathbf{Z}_{1,[i]}$. This leads to

$$\mathbf{v}_{[0]} = \mathbf{Z}_{1,[0]}\mathbf{Z}_{1,[1]} \cdots \mathbf{Z}_{1,[k]}\mathbf{v}_{[k]}. \tag{38}$$

For a null space decomposition in step j, we compute the inverse transformation of (31) by

$$\mathbf{z}_{[j+1]} = \mathbf{v}_{[j]}\mathbf{G}_{[j]}^{+\mathsf{R}} \tag{39}$$

where $\mathbf{G}_{[j]}^{+\mathsf{R}}$ also satisfies $\mathbf{Z}_{1,[j]}^{+\mathsf{R}}\mathbf{G}_{[j]}^{+\mathsf{R}} = \mathbf{0}$ and $\tilde{\mathbf{Z}}_{1,[j]}^{\perp\mathsf{L}}\mathbf{G}_{[j]}^{+\mathsf{R}} = \mathbf{0}$ and is therefore unique. This additionally leads to the inverse transformation

$$\mathbf{v}_{[0]} = \mathbf{Z}_{1,[0]}\mathbf{Z}_{1,[1]} \cdots \mathbf{Z}_{1,[j-1]}\mathbf{G}_{[j]}^{+\mathsf{R}}\mathbf{v}_{[k]}. \tag{40}$$

Eventually, the unimodular completion of $\mathbf{Z}(\lambda) = \mathbf{Z}_1\lambda + \mathbf{Z}_0$ is given by

$$\tilde{\mathbf{Q}} \coloneqq \mathbf{Z}_{1,[0]}\mathbf{Z}_{1,[1]} \cdots \mathbf{Z}_{1,[k]}, \tag{41}$$

possibly appended column-wise by

$$\tilde{\mathbf{Q}}_{[j]} \coloneqq \mathbf{Z}_{1,[0]}\mathbf{Z}_{1,[1]} \cdots \mathbf{Z}_{1,[j-1]}\mathbf{G}_{[j]}^{+\mathsf{R}} \tag{42}$$

for every null space decomposition in step j.

Note that the unimodular completion that results from this algorithm is independent of λ.

C. Computing differentially flat inputs using a unimodular completion

If we do not allow input injection into the output equation, then a general unimodular completion needs to be transformed such that the lowest m rows of the completion are zero, i.e.,

$$\mathbf{Q}(\lambda) =: \begin{pmatrix} \mathbf{Q}_1(\lambda) \\ \mathbf{Q}_2(\lambda) \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} \mathbf{Q}_1(\lambda) \\ \mathbf{0}_{m\times m} \end{pmatrix}. \tag{43}$$

In general, the right hyper column of a $(n+m)\times(n+m)$ unimodular matrix can be altered by right multiplication with a unimodular transformation matrix $\mathbf{T}(\lambda) \in \mathfrak{U}_{n+m}[\lambda]$, i.e.,

$$\begin{pmatrix} \mathbf{A}_1(\lambda) & \mathbf{A}_2 \\ \mathbf{A}_3(\lambda) & \mathbf{A}_4 \end{pmatrix} \underbrace{\begin{pmatrix} \mathbf{I}_n & \mathbf{B}_1(\lambda) \\ \mathbf{0} & \mathbf{B}_2(\lambda) \end{pmatrix}}_{=:\mathbf{T}(\lambda)} = \begin{pmatrix} \mathbf{A}_1(\lambda) & \mathbf{C}_1(\lambda) \\ \mathbf{A}_3(\lambda) & \mathbf{C}_2(\lambda) \end{pmatrix} \in \mathfrak{U}_{n+m}[\lambda]$$

with

$$\mathbf{C}_1(\lambda) \coloneqq \mathbf{A}_1(\lambda)\mathbf{B}_1(\lambda) + \mathbf{A}_2\mathbf{B}_2(\lambda) \tag{44}$$

$$\mathbf{C}_2(\lambda) \coloneqq \mathbf{A}_3(\lambda)\mathbf{B}_1(\lambda) + \mathbf{A}_4\mathbf{B}_2(\lambda). \tag{45}$$

Here, $\mathbf{A}_1(\lambda) = \mathbf{I}_n\lambda + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \in \mathfrak{K}^{n\times n}[\lambda]$, $\mathbf{A}_3(\lambda) = \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \in \mathfrak{K}^{m\times n}[\lambda]$ and $\mathrm{row}(\mathbf{A}_2, \mathbf{A}_4)$ is a unimodular completion computed as described in the section above. To ensure unimodularity of the resulting matrix, $\mathbf{T}(\lambda) \in \mathfrak{U}_{n+m}[\lambda]$ implies $\mathbf{B}_2(\lambda) \in \mathfrak{U}_m[\lambda]$.

Due to our assumptions on the output $\mathbf{y}$, there is no feedthrough of the input $\mathbf{u}$. This is why we require

$$\mathbf{C}_2(\lambda) = \mathbf{A}_3(\lambda)\mathbf{B}_1(\lambda) + \mathbf{A}_4\mathbf{B}_2(\lambda) \stackrel{!}{=} \mathbf{0} \tag{46}$$

which we can right multiply by $\mathbf{B}_2^{-1}(\lambda) \in \mathfrak{U}_m[\lambda]$ to get

$$\mathbf{A}_3(\lambda)\mathbf{B}_1(\lambda)\mathbf{B}_2^{-1}(\lambda) + \mathbf{A}_4 = \mathbf{0}. \tag{47}$$

The matrix $\mathbf{A}_3(\lambda)$ is necessarily hyper-regular, i.e., there exists a hyper-regular right inverse $\mathbf{A}_3^{+\mathsf{R}}$. Setting

$$\mathbf{B}_1(\lambda) = -(\mathbf{A}_3(\lambda))^{+\mathsf{R}}\mathbf{A}_4 \quad \text{and} \quad \mathbf{B}_2(\lambda) = \mathbf{I}_m \tag{48}$$

fulfills requirement (46), such that our resulting blocks are of the form

$$\mathbf{C}_1(\lambda) = -\mathbf{A}_1(\lambda)(\mathbf{A}_3(\lambda))^{+\mathsf{R}}\mathbf{A}_4 + \mathbf{A}_2 \tag{49}$$

$$\mathbf{C}_2(\lambda) = \mathbf{0}. \tag{50}$$

Note that in general $\mathbf{C}_1(\lambda)$ depends on the operator λ and thus may introduce time derivatives of the flat input $\mathbf{u}$, which may not be desirable.

Assuming, we have managed to find a λ-independent unimodular completion of the form $\mathbf{Q} = \mathrm{row}(\mathbf{Q}_1, \mathbf{0}_{m,m})$, then the input injection

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{Q}_1(\mathbf{x})\mathbf{u} \tag{51}$$

is flat, if the (now modified) tangent matrix of the implicit version of (51) and the given output

$$\begin{pmatrix} \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{Q}_1\mathbf{u}}{\partial \mathbf{x}} & \mathbf{Q}_1 \\ \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n\lambda - \frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{Q}_1\mathbf{u}}{\partial \mathbf{x}} & \mathbf{Q}_1 \\ \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} & \mathbf{0} \end{pmatrix} \tag{52}$$

still enjoys the unimodularity property. The provided output $\mathbf{y}$ is then a flat output of the system (51).

IV. Example

We compute a flat input for the following academic example system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_2 - x_3 \\ x_1 \\ x_2 x_3 \end{pmatrix}, \quad \mathbf{x}(t) \in \mathbb{R}^3 \tag{53}$$

with the output $\mathbf{y} = (x_3^2, x_1^2)^\mathsf{T}$. From the implicit system $\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}) = \dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}) = \mathbf{0}$ we compute the tangent matrix appended by the generalized Jacobian of the output $\mathbf{y}$

$$\mathbf{Z}(\lambda) := \begin{pmatrix} \mathbf{P}(\lambda) \\ \mathbf{H} \end{pmatrix} = \begin{pmatrix} \lambda & -1 & 1 \\ -1 & \lambda & 0 \\ 0 & -x_3 & \lambda - x_2 \\ 0 & 0 & 2x_3 \\ 2x_1 & 0 & 0 \end{pmatrix}. \tag{54}$$

The algorithm starts with $i = 0$ by setting $\mathbf{Z}_{1,[0]} := \mathbf{Z}_1$. In the reduction step we compute

$$\mathbf{Z}_{1,[0]}^{+\mathsf{L}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{Z}_{1,[0]}^{\perp\mathsf{L}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

which yields

$$\mathbf{B}_{[0]} = \mathbf{Z}_{1,[0]}^{\perp\mathsf{L}} \mathbf{Z}_{0,[0]} = \begin{pmatrix} 0 & 0 & 2x_3 \\ 2x_1 & 0 & 0 \end{pmatrix},$$

$$\mathbf{C}_{[0]} = \mathbf{Z}_{1,[0]}^{+\mathsf{L}} \mathbf{Z}_{0,[0]} = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 0 \\ 0 & -x_3 & -x_2 \end{pmatrix}.$$

The matrix $\mathbf{B}_{[0]}$ has full row rank, i.e., we can skip the null space decomposition and proceed with the elimination step: We calculate

$$\mathbf{Z}_{1,[1]} = \mathbf{B}_{[0]}^{\perp\mathsf{R}} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{Z}_{0,[1]} = \mathbf{C}_{[0]} \mathbf{B}_{[0]}^{\perp\mathsf{R}} = \begin{pmatrix} -1 \\ 0 \\ -x_3 \end{pmatrix}.$$

Note that $\frac{\partial}{\partial t}(\mathbf{B}_{[0]}^{\perp\mathsf{R}}) = \mathbf{0}$ for this particular choice of $\mathbf{B}_{[0]}^{\perp\mathsf{R}}$. Setting $i = 1$, we go back to the reduction step. We compute

$$\mathbf{Z}_{1,[1]}^{+\mathsf{L}} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{Z}_{1,[1]}^{\perp\mathsf{L}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which leads to

$$\mathbf{B}_{[1]} = \mathbf{Z}_{1,[1]}^{\perp\mathsf{L}} \mathbf{Z}_{0,[1]} = \begin{pmatrix} -1 \\ -x_3 \end{pmatrix}, \quad \mathbf{C}_{[1]} = \mathbf{Z}_{1,[1]}^{+\mathsf{L}} \mathbf{Z}_{0,[1]} = \begin{pmatrix} 0 \end{pmatrix}.$$

Here, $\mathbf{B}_{[1]}$ does not have full row rank, i.e., we proceed with a null space decomposition and compute

$$\mathbf{B}_{[1]}^{\perp\mathsf{L}} = \begin{pmatrix} -x_3 & 1 \end{pmatrix}, \quad \mathbf{G}_{[1]} = \mathbf{B}_{[1]}^{\perp\mathsf{L}} \mathbf{Z}_{1,[1]}^{\perp\mathsf{L}} = \begin{pmatrix} -x_3 & 0 & 1 \end{pmatrix}.$$

as well as

$$\tilde{\mathbf{Z}}_{1,[1]}^{\perp\mathsf{L}} = ((\mathbf{B}_{1,[1]}^{\perp\mathsf{L}})^{\perp\mathsf{R}})^\mathsf{T} \mathbf{Z}_{1,[1]}^{\perp\mathsf{L}} = \begin{pmatrix} 1 & 0 & x_3 \end{pmatrix}$$

from which follows

$$\mathbf{B}_{[1]} = \tilde{\mathbf{B}}_{[1]} = \tilde{\mathbf{Z}}_{1,[1]}^{\perp\mathsf{L}} \mathbf{Z}_{0,[1]}^{\perp\mathsf{L}} = \begin{pmatrix} -1 - x_3^2 \end{pmatrix}.$$

The scalar $\mathbf{B}_{[1]}$ has rank 1, and therefore full column rank. Thus, the algorithm terminates. We compute the first part of the completion according to (41)

$$\tilde{\mathbf{Q}} = \mathbf{Z}_{1,[0]} \mathbf{Z}_{1,[1]} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{55}$$

Due to the null space decomposition, we need to compute the remaining columns as described in (42). For this, we compute $\mathbf{G}_{[1]}^{+\mathsf{R}}$, such that $\mathbf{Z}_{1,[1]}^{+\mathsf{R}} \mathbf{G}_{[1]}^{+\mathsf{R}} = \mathbf{0}$ and $\tilde{\mathbf{Z}}_{1,[j]}^{\perp\mathsf{L}} \mathbf{G}_{[j]}^{+\mathsf{R}} = \mathbf{0}$. We get

$$\mathbf{G}_{[1]}^{+\mathsf{R}} = \begin{pmatrix} -\frac{x_3}{x_3^2+1} \\ 0 \\ \frac{1}{x_3^2+1} \end{pmatrix} \tag{56}$$

which results in

$$\tilde{\mathbf{Q}}_{[1]} = \mathbf{Z}_{1,[0]} \mathbf{G}_{[1]}^{+\mathsf{R}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -\frac{x_3}{x_3^2+1} \\ 0 \\ \frac{1}{x_3^2+1} \end{pmatrix} = \begin{pmatrix} -\frac{x_3}{x_3^2+1} \\ 0 \\ \frac{1}{x_3^2+1} \\ 0 \\ 0 \end{pmatrix}.$$

Finally, the described algorithm results in the unimodular completion $\tilde{\mathbf{Q}}$ with

$$\tilde{\mathbf{Q}} := (\tilde{\mathbf{Q}}, \tilde{\mathbf{Q}}_{[1]}) = \begin{pmatrix} 0 & -\frac{x_3}{x_3^2+1} \\ 1 & 0 \\ 0 & \frac{1}{x_3^2+1} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \tag{57}$$

i.e.,

$$\begin{pmatrix} \lambda & -1 & 1 & 0 & -\frac{x_3}{x_3^2+1} \\ -1 & \lambda & 0 & 1 & 0 \\ 0 & -x_3 & \lambda - x_2 & 0 & \frac{1}{x_3^2+1} \\ 0 & 0 & 2x_3 & 0 & 0 \\ 2x_1 & 0 & 0 & 0 & 0 \end{pmatrix} \in \mathcal{U}_5[\lambda]. \tag{58}$$

The lower m rows of the completion $\tilde{\mathbf{Q}}$ are zero already, so we can skip some computations here, that is $\mathbf{Q} = \tilde{\mathbf{Q}}$ and

$$\mathbf{Q}_1 = \begin{pmatrix} 0 & -\frac{x_3}{x_3^2+1} \\ 1 & 0 \\ 0 & \frac{1}{x_3^2+1} \end{pmatrix}.$$

Furthermore, the completion has non-constant coefficients, i.e., the input injected equations have a modified tangent matrix. Thus, we need to make sure the unimodularity condition still holds:

The input injected system in implicit form appended by the output $\mathbf{y}$ reads

$$\begin{pmatrix}\mathbf{0}\\ \mathbf{y}\end{pmatrix} = \begin{pmatrix}\dot{x}_1 - x_2 + x_3\\ \dot{x}_2 - x_1\\ \dot{x}_3 - x_2x_3\\ x_3^2\\ x_1^2\end{pmatrix} + \begin{pmatrix}-\frac{x_3}{x_3^2+1}u_2\\ u_1\\ \frac{1}{x_3^2+1}u_2\\ 0\\ 0\end{pmatrix}. \tag{59}$$

The generalized Jacobian of (59) w.r.t. $\mathrm{row}(\mathbf{x}, \mathbf{u})$ yields

$$\begin{pmatrix}\lambda & -1 & \frac{2x_3^2u_2}{(x_3^2+1)^2} - \frac{u_2}{x_3^2+1} + 1 & 0 & -\frac{x_3}{x_3^2+1}\\ -1 & \lambda & 0 & 1 & 0\\ 0 & -x_3 & \lambda - x_2 - \frac{2x_3u_2}{(x_3^2+1)^2} & 0 & \frac{1}{x_3^2+1}\\ 0 & 0 & 2x_3 & 0 & 0\\ 2x_1 & 0 & 0 & 0 & 0\end{pmatrix} \tag{60}$$

and using methods from [30] we can indeed verify unimodularity. Therefore, the computed input is in fact a flat input and the input injected system reads

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{Q}_1(\mathbf{x})\mathbf{u} = \begin{pmatrix}x_2 - x_3\\ x_1\\ x_2x_3\end{pmatrix} + \begin{pmatrix}0 & -\frac{x_3}{x_3^2+1}\\ 1 & 0\\ 0 & \frac{1}{x_3^2+1}\end{pmatrix}\mathbf{u}$$

with the output $\mathbf{y} = (x_3^2, x_1^2)^\mathsf{T}$ being flat.

V. Direct flat representation

As described in [31], the choice of coordinates of the system's description has an impact on the *visibility* of the flatness property, and as such as well on the computation of flat outputs. This is true for the computation of flat inputs, too:

If the matrix

$$\mathbf{Z}(\lambda) = \begin{pmatrix}\mathbf{P}(\lambda)\\ \mathbf{H}(\lambda)\end{pmatrix} = \begin{pmatrix}\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}\\ \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\end{pmatrix} \in \mathfrak{K}^{(n+m)\times n}[\lambda] \tag{61}$$

contains a $n \times n$ unimodular submatrix, then we can rearrange the rows by left multiplication with a $(n+m)\times(n+m)$ permutation matrix $\mathbf{V}_\pi$ such that

$$\tilde{\mathbf{Z}}(\lambda) := \mathbf{V}_\pi \begin{pmatrix}\mathbf{P}(\lambda)\\ \mathbf{H}(\lambda)\end{pmatrix} =: \begin{pmatrix}\mathbf{A}_1(\lambda)\\ \mathbf{A}_2(\lambda)\end{pmatrix}, \quad \mathbf{A}_1 \in \mathfrak{U}_n[\lambda], \tag{62}$$

then a unimodular completion of $\tilde{\mathbf{Z}}(\lambda)$ can be specified by

$$\hat{\mathbf{Q}} = \begin{pmatrix}\mathbf{0}_{n,m}\\ \mathbf{I}_m\end{pmatrix} \tag{63}$$

and therefore, a unimodular completion of $\mathbf{Z}(\lambda)$ is given by

$$\mathbf{Q} = \mathbf{V}_\pi^\mathsf{T}\hat{\mathbf{Q}} = \mathbf{V}_\pi^\mathsf{T}\begin{pmatrix}\mathbf{0}_{n,m}\\ \mathbf{I}_m\end{pmatrix}. \tag{64}$$

Note that this completion only has constant elements, i.e., $\frac{\partial \mathbf{Q}\mathbf{u}}{\partial \mathbf{x}} = \mathbf{0}$ and the tangent matrix is not influenced by the input injection (compare (52)). Hence, the unimodularity condition fulfilled. Again, if we do not allow input injection in the output equation, then we need to make sure the lowest m rows of this completion is zero. If this is the case, then we get

$$\mathbf{Q}_1 = \begin{pmatrix}\mathbf{I}_n & \mathbf{0}_{n,m}\end{pmatrix}\mathbf{V}_\pi\begin{pmatrix}\mathbf{0}_{n,m}\\ \mathbf{I}_m\end{pmatrix}$$

and the input-injected system reads

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{Q}_1\mathbf{u}. \tag{65}$$

Example 1. *The previously described example with the dynamics* (53)*, the output* $\mathbf{y} = (x_3^2, x_1^2)^\mathsf{T}$ *and the tangent matrix* (54) *has a unimodular submatrix of maximum size, i.e.,*

$$\begin{pmatrix}0 & -x_3 & \lambda - x_2\\ 0 & 0 & 2x_3\\ 2x_1 & 0 & 0\end{pmatrix} \in \mathfrak{U}_3[\lambda] \tag{66}$$

holds. Therefore, a unimodular completion is given by

$$\mathbf{Q} = \begin{pmatrix}\mathbf{I}_2\\ \mathbf{0}_{3,2}\end{pmatrix} \tag{67}$$

and due to all entries of the completion being constants the tangent matrix of the original system remains the same. Hence, the flat input injected system reads

$$\dot{\mathbf{x}} = \begin{pmatrix}x_2 - x_3\\ x_1\\ x_2x_3\end{pmatrix} + \begin{pmatrix}u_1\\ u_2\\ 0\end{pmatrix} \quad \text{with} \quad \mathbf{y} = \begin{pmatrix}x_3^2\\ x_1^2\end{pmatrix} \tag{68}$$

where $\mathbf{y}$ *constitutes a flat output.*

VI. Problem characterization for non-observable systems

According to [19], observability is not necessary for the existence of flat inputs in the MIMO case. This seems somewhat strange at first, but using the above approach it can be seen why this may still allow the computation of flat inputs.

We assume no additional restrictions on $\hat{\mathbf{f}}(\mathbf{x}, \mathbf{u})$, such that we can require the structure

$$\dot{\mathbf{x}} = \hat{\mathbf{f}}(\mathbf{x}, \mathbf{u}) = \mathbf{f}(\mathbf{x}) + \hat{\mathbf{b}}(\mathbf{x}, \mathbf{u}) \tag{69}$$

with

$$\hat{\mathbf{b}}(\mathbf{x}, \mathbf{u}) = -\mathbf{f}(\mathbf{x}) + \mathbf{b}(\mathbf{x}, \mathbf{u}) \tag{70}$$

and input-dependent dynamics $\mathbf{b}(\mathbf{x}, \mathbf{u})$. This results in

$$\dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}, \mathbf{u}), \tag{71}$$

i.e., the following two conditions are sufficient for differential flatness

$$\begin{pmatrix}\mathbf{I}_n\lambda - \frac{\partial \mathbf{b}}{\partial \mathbf{x}} & -\frac{\partial \mathbf{b}}{\partial \mathbf{u}}\\ \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{x}}}\lambda + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} & \mathbf{0}_{m,m}\end{pmatrix} \overset{!}{\in} \mathfrak{U}_{n+m}[\lambda] \tag{72}$$

and

$$\mathrm{d}\left(\begin{pmatrix}\mathbf{I}_n\lambda - \frac{\partial \mathbf{b}}{\partial \mathbf{x}} & -\frac{\partial \mathbf{b}}{\partial \mathbf{u}}\end{pmatrix}\begin{pmatrix}\mathrm{d}\mathbf{x}\\ \mathrm{d}\mathbf{u}\end{pmatrix}\right) \overset{!}{=} \mathbf{0}, \tag{73}$$

where $\mathbf{y} = \mathbf{h}(\mathbf{x}, \dot{\mathbf{x}}, \ldots, \mathbf{x}^{(\alpha)})$ with $\mathbf{y}(t) \in \mathbb{R}^m$ is the given output. The problem here is to find $\mathbf{b}$ which is similar to the problem of computing flat outputs, i.e., a unimodular row-completion with an integrability condition. Here, however, the matrix $(\mathbf{I}_n\lambda, \mathbf{0}_{m,m})$ needs to be added, so there is an additional constraint. The resulting system is in fact observable. It is not straight forward how to incorporate this into an algorithm. Note that for two-output systems, i.e., $\mathbf{y}(t) \in \mathbb{R}^2$, the problem of flat input computation has been solved [20].

We illustrate the above ideas by the an example from [19, ex. 4]:

Example 2. *Given the pair of dynamical system and output*

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{y} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \tag{74}$$

the vector

$$\mathbf{b}(\mathbf{x}, \mathbf{u}) = \begin{pmatrix} u_1 \\ x_3 u_1 \\ u_2 \end{pmatrix} \tag{75}$$

with

$$\frac{\partial \mathbf{b}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & u_1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \frac{\partial \mathbf{b}}{\partial \mathbf{u}} = \begin{pmatrix} 1 & 0 \\ x_3 & 0 \\ 0 & 1 \end{pmatrix}, \tag{76}$$

results in the matrix from (72)

$$\begin{pmatrix} \lambda & 0 & 0 & -1 & 0 \\ 0 & \lambda & -u_1 & -x_3 & 0 \\ 0 & 0 & \lambda & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{77}$$

which is unimodular. Therefore, the input-injected system reads $\dot{\mathbf{x}} = \mathbf{b}(\mathbf{x}, \mathbf{u})$ *with the output* $\mathbf{y} = (x_1, x_2)^\mathsf{T}$.

VII. Conclusion

Dual to the concept of flat outputs, flat inputs can be interpreted as the problem of actuator placement, i.e., given an autonomous system with an output equation, we are interested in an input injected system, such that the output becomes flat. While in general the dynamical system and the output equation can depend on higher derivatives of the state vector, here we focus on first order equations, i.e., only time derivatives of first order are allowed to occur. If the system of interest can be transformed into state space representation, unimodular row completion of higher order equations can be traced back to completions of first order equations using Lie-derivatives. Besides that, output equations are usually chosen to be physically meaningful, which means the order of time derivatives in the output equation should be low. So, restricting to first order equations may not be a practical limitation.

Similar to the computation of flat outputs, we show that for observable systems flat input computation can be associated with unimodular (column) completion of the generalized Jacobian of the implicit dynamical equation appended by a given output. Dealing with Ore polynomials, i.e., mathematical objects with non-commutative multiplication, we propose an algorithm for computing such unimodular completions and show that no integrability condition is required to be satisfied for observable systems – unlike in flat output computation with the same approach. Instead, an additional unimodularity condition needs to be fulfilled.

Since observability is not necessary for the computation of flat inputs, we reformulate the non-observable case in the same framework.

References

[1] P. Martin, "Contribution à l'étude des systèmes différentiellement plats," Ph.D. dissertation, École des Mines de Paris, 1992.

[2] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[3] E. Delaleau and J. Rudolph, "Control of flat systems by quasi-static feedback of generalized states," *International Journal of Control*, vol. 71, pp. 745–765, 1998.

[4] V. Hagenmeyer and E. Delaleau, "Exact feedforward linearization based on differential flatness," *International Journal of Control*, vol. 76, pp. 537–556, 2003.

[5] H. Sira-Ramírez and S. K. Agrawal, *Differentially flat systems*. Control engineering series, 2004.

[6] J. Lévine, *Analysis and Control of Nonlinear Systems*. Springer, Berlin Heidelberg, 2009.

[7] ——, "On necessary and sufficient conditions for differential flatness," *Applicable Algebra in Engineering, Communication and Computing*, vol. 22, no. 1, pp. 47–90, 2011.

[8] M. Schöberl and K. Schlacher, "Construction of flat outputs by reduction and elimination," *Proceedings of 7th IFAC Symposium on Nonlinear Control Systems (NOLCOS), Pretoria*, vol. 40, no. 12, pp. 666–671, 2007.

[9] F. Antritter and J. Lévine, "Towards a computer algebraic algorithm for flat output determination," *Proceedings of 21st International Symposium on Symbolic and Algebraic Computation (ISSAC), Hagenberg*, pp. 7–14, 2008.

[10] M. Franke and K. Röbenack, "On the computation of flat outputs for nonlinear control systems," *European Control Conference (ECC)*, pp. 167–172, 2013.

[11] M. Franke, *Über die Konstruktion flacher Ausgänge für nichtlineare Systeme und zur Polzuweisung durch statische Ausgangsrückführungen*. Diss., Logos-Verlag, Berlin, 2014.

[12] K. Fritzsche, M. Franke, C. Knoll, and K. Röbenack, "On the systematic computation of flat outputs for nonlinear systems with multiple inputs (german)," *at - Automatisierungstechnik*, vol. 64, no. 12, pp. 948–960, 2016.

[13] J. Imae, Y. Akasawa, and T. Kobayashi, "Practical computation of flat outputs for nonlinear control systems," *IOP Conf. Series: Materials Science and Engineering*, vol. 83, 012010, 2015.

[14] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Some open questions related to flat nonlinear systems," in *Open Problems in Mathematical Systems and Control Theory*, ser. Communications and Control Engineering. Springer London, 1999, pp. 99–103.

[15] P. Martin, R. M. Murray, and P. Rouchon, "Flat systems: open problems, infinite dimensional extension, symmetries and catalog," in *Lecture Notes in Control and Information Sciences, Advances in the Control of Nonlinear Systems*. Springer, London, 2007, pp. 33–57.

[16] M. Schöberl and K. Schlacher, "On an implicit triangular decomposition of nonlinear control systems that are 1-flat – a constructive approach," *Automatica*, vol. 50, no. 6, pp. 1649–1655, 2014.

[17] M. Franke and K. Röbenack, "Some remarks concerning differential flatness and tangent systems," *Proceedings in Applied Mathematics and Mechanics (PAMM), Darmstadt*, vol. 12, pp. 729–730, 2012.

[18] S. Waldherr and M. Zeitz, "Conditions for the existence of a flat input," *International Journal of Control*, vol. 81, no. 3, pp. 439–443, 2008.

[19] ——, "Flat inputs in the mimo case," *8th IFAC Symposium on Nonlinear Control Systems, Bologna, Italy, September 1-3*, 2010.

[20] F. Nicolau, W. Respondek, and J.-P. Barbot, "Constructing flat inputs for two-output systems," *The 23rd International Symposium on Mathematical Theory of Networks and Systems*, Jul 2018, Hong Kong, France.

[21] J.-F. Stumper, F. Svaricek, and R. Kennel, "Trajectory tracking control with flat inputs and a dynamic compensator," *Proc. of the European Control Conference 2009, Budapest, Hungary, August 23-26, 2009*, pp. 248–253, 2009.

[22] A. Krener and A. Isidori, "Linearization by output injection and nonlinear observers," *Systems & Control Letters*, vol. 3, pp. 47–52, 1983.

[23] J. Deutscher, F. Antritter, and K. Schmidt, "Feedforward control of nonlinear systems using fictitious inputs," *Proc. of the 44th IEEE Conf. on Decisions and Control and European Control Conf.*, 2005.

[24] F. Antritter and J. Deutscher, "Asymptotic tracking for nonlinear systems using fictitious inputs," *Proc. of the 44th IEEE Conf. on Decision and*

Control, and the European Control Conference 2005, Seville, Spain, December 12-15, 2005, pp. 4022–4027, 2005.

[25] V. Vassilev, M. Gröschel, H.-J. Schmid, W. Peukert, and G. Leugering, "Interfacial energy estimation in a precipitation reaction using the flatness based control of the moment trajectories," *Chemical Engineering Science*, vol. 65, no. 6, pp. 2183–2189, 2010.

[26] R. Schenkendorf and M. Mangold, "Parameter identification for ordinary and delay differential equations by using flat inputs," *Theoretical Foundations of Chemical Engineering*, vol. 48, no. 5, pp. 594–607, 2014.

[27] B. Beckermann, H. Cheng, and G. Labahn, "Fraction-free row reduction of matrices of ore polynomials," *Journal of Symbolic Computation*, vol. 41, pp. 513–543, 2006.

[28] F. Antritter, F. Cazaurang, J. Lévine, and J. Middeke, "On the computation of π-flat outputs for linear time-varying differential-delay systems," *Systems & Control Letters*, vol. 71, pp. 14–22, 2014.

[29] F. Antritter and J. Middeke, "A toolbox for the analysis of linear systems with delays," *CDC-ECC, Orlando, USA*, pp. 1950–1955, 2011.

[30] K. Fritzsche and K. Röbenack, "On hyper-regularity and unimodularity of Ore polynomial matrices," *International Journal of Applied Mathematics and Computer Science*, vol. 28, no. 3, 2018 (accepted).

[31] K. Fritzsche, C. Knoll, M. Franke, and K. Röbenack, "Unimodular completion and direct flat representation in the context of differential flatness," *Proceedings in Applied Mathematics and Mechanics (PAMM)*, vol. 16, no. 1, pp. 807–808, 2016.