2017 21st International Conference on System Theory, Control and Computing (ICSTCC)

Integration of Output Tracking and Trajectory Generation via Analytic Left Inversion

Luis A. Duffaut Espinosa
Department of Electrical and
Biomedical Engineering
University of Vermont
Burlington, VT 05405 USA

W. Steven Gray
Department of Electrical and
Computer Engineering
Old Dominion University
Norfolk, VA 23529 USA

***Abstract*—In this paper, an algorithm for the integration of output tracking and trajectory generation via analytic left inversion is provided. The first step is to identify an output path using a known trajectory generation algorithm, and then a spline approximation of the path is computed. The second step is to solve the output tracking problem by explicitly computing the left inverse of the input-output map of the system to render the Taylor series of the desired input for each polynomial section of the spline approximation. In practice, these infinite series must be truncated and executed over a finite interval of time, so the relationship between truncation, tracking error and execution time is characterized. A case study is presented involving the kinematics of a bi-steerable car.**

***Index Terms*—Output tracking, system inversion, Chen-Fliess series**

I. Introduction

The design of self-driving machines is rapidly evolving due to the removal of the human decision making process and its replacement by a computer control system [14]. For example, a car with independent front and rear axles (bi-steerable) provides improved maneuverability, which would be invaluable in urban settings to improve safety and performance [15], [19], [20]. But such improved performance is predicated on control methodologies that properly combine the true dynamics of the machine with the desired output trajectory [3], [17].

The main goal of this paper is the integration of an output tracking methodology with an out-of-the-box trajectory generation algorithm. The first step is solving the output trajectory generation problem. For instance, one choice for such an algorithm is the rapidly exploring random trees (RRT) method [17]. The second step is solving the output tracking problem. That is, determine the control inputs so that the system's output accurately follows the desired trajectory. Mathematically, this corresponds to computing a left inverse for the system. There is an extensive literature on this topic in the context of geometric nonlinear control theory, see, for example, [5], [7], [16]. This class of techniques generates left inverses implicitly using feedback. While very convenient in some situations, this method has certain limitations, like requiring the system to be minimum phase. There is also no way in this setting to produce an *explicit* expression for the desired control. Another common approach is to use the property of *flatness*, which allows one to explicitly compute a left inverse analytically provided suitable flat outputs can be identified [6], [18]. A final method, which is the one employed in this paper, is to use an analytic expression for the left inverse which is available for systems which have a well defined vector relative degree for the actual outputs of interest [9], [11], [12]. They do *not* have to be flat outputs. The method is based on a Fliess operator representation of the input-output system, and an existing combinatorial Hopf algebra technique that renders an explicit formula for the Taylor series of the left inverse when the output is an analytic function in the range of the input-output map. This single formula can be pre-computed efficiently off-line to arbitrary precision [4], and, for example, could be hardwired on an field-programmable gate array (FPGA) for real-time implementation. However, infinite series Taylor series must be truncated in practice and can only be executed over a finite interval of time. So this will lead to tracking errors. The main goals of this paper are to provide:

i. An algorithm for the integration of output tracking and trajectory generation via analytic left inversion.
ii. An analysis and characterization of the relationship between truncation, tracking error and execution time.
iii. Illustrate the method for the bi-steerable car.

The paper is organized as follows. In Section II, preliminaries concerning Fliess operators and their inverses are briefly summarized. In Section III, new results about error bounds, execution times and the effect of series truncation for Fliess operator are developed. Then, in Section IV, the left inverse computations are applied to the kinematics of the bi-steerable car. This result is then integrated in Section V with the RRT trajectory generation algorithm and the corresponding numerical simulations are presented. Conclusions are provided in the final section.

II. Preliminaries

In this section, some preliminaries concerning the theory of Fliess operators are outlined based on [8], [11], [12].

A. Fliess Operators and Their Interconnections

A set of noncommuting symbols $X = \{x_0, x_1, \ldots, x_m\}$ with $m \geq 0$ finite is called an *alphabet*. Elements of X are referred to as *letters*, and a finite sequence of letters from X, for example, $\eta = x_{i_1} \cdots x_{i_k}$, is called a *word*. Let $|\eta|$ denote the number of letters in η. The collection of words having length k is denoted by X^k, while the set of all words, including the empty word $\emptyset$, is represented by X^*. This set is a monoid under catenation. The symbol ηX^* is used for the set of all words having the prefix η. A mapping $c : X^* \to \mathbb{R}^\ell$ is called a *formal power series*. The value of c for a specific word $\eta \in X^*$ is called the *coefficient* of η in c and usually denoted by (c, η). It is more common to write c as the formal sum $c = \sum_{\eta \in X^*}(c, \eta)\eta$. When the *constant term* $(c, \emptyset) = 0$, the series c is called *proper*. The *support* of c is the set

978-1-5386-3842-2/17/$31.00 ©2017 IEEE

$\mathrm{supp}(c) = \{\eta \in X^* : (c,\eta) \neq 0\}$. The collection of all formal power series over X, namely, $\mathbb{R}^\ell\langle\langle X\rangle\rangle$, forms an associative $\mathbb{R}$-algebra under catenation and a commutative and associative $\mathbb{R}$-algebra under the shuffle product. The latter is the $\mathbb{R}$-bilinear extension of the shuffle product of two words, which is defined inductively by $(x_i\eta) \sqcup\!\sqcup (x_j\xi) = x_i(\eta \sqcup\!\sqcup (x_j\xi)) + x_j((x_i\eta) \sqcup\!\sqcup \xi)$ with $\eta \sqcup\!\sqcup \emptyset = \emptyset \sqcup\!\sqcup \eta = \eta$ for all $\eta, \xi \in X^*$ and $x_i, x_j \in X$. Finally, for any letter $x_i \in X$, the corresponding *left-shift operator* is defined on X^* by $x_i^{-1}(x_i\eta) = \eta$ with $\eta \in X^*$ and zero otherwise. Higher order shifts are defined inductively, and all such operators act linearly on $\mathbb{R}^m\langle\langle X\rangle\rangle$.

A causal m-input, ℓ-output operator, F_c, can be formally identified with any generating series $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$. Fix $\mathfrak{p} \geq 1$ and $t_0 < t_1$. Given a Lebesgue measurable function $u : [t_0, t_1] \to \mathbb{R}^m$, define its norm $\|u\|_\mathfrak{p} = \max\{\|u_i\|_\mathfrak{p} : 1 \leq i \leq m\}$, where $\|u_i\|_\mathfrak{p}$ is the usual $L_\mathfrak{p}$-norm for a measurable real-valued function, u_i, defined on $[t_0, t_1]$. Let $L_\mathfrak{p}^m[t_0, t_1]$ denote the set of all measurable functions defined on $[t_0, t_1]$ having a finite $\|\cdot\|_\mathfrak{p}$ norm and $B_\mathfrak{p}^m(R)[t_0, t_1] := \{u \in L_\mathfrak{p}^m[t_0, t_1] : \|u\|_\mathfrak{p} \leq R\}$. Define inductively for each $\eta \in X^*$ the map $E_\eta : L_1^m[t_0, t_1] \to C[t_0, t_1]$ by setting $E_\emptyset[u] = 1$ and letting

$$E_{x_i\bar{\eta}}[u](t, t_0) = \int_{t_0}^{t} u_i(\tau) E_{\bar{\eta}}[u](\tau, t_0)\, d\tau,$$

where $x_i \in X$, $\bar{\eta} \in X^*$, and $u_0 = 1$. The input-output system corresponding to c is the *Fliess operator*

$$F_c[u](t) = \sum_{\eta \in X^*} (c, \eta)\, E_\eta[u](t, t_0).$$

If there exist real numbers $K_c, M_c > 0$ such that

$$|(c,\eta)| \leq K_c M_c^{|\eta|} |\eta|!, \quad \forall \eta \in X^*,$$

then F_c constitutes a well defined mapping from $B_\mathfrak{p}^m(R)[t_0, t_0 + T]$ into $B_\mathfrak{q}^\ell(S)[t_0, t_0 + T]$ for sufficiently small $R, T > 0$, where the numbers $\mathfrak{p}, \mathfrak{q} \in [1, \infty]$ are conjugate exponents, i.e., $1/\mathfrak{p} + 1/\mathfrak{q} = 1$. (Here, $|z| := \max_i |z_i|$ when $z \in \mathbb{R}^\ell$.) The set of all such *locally convergent* series is denoted by $\mathbb{R}_{LC}^\ell\langle\langle X\rangle\rangle$. On the other hand, if

$$|(c,\eta)| \leq K_c M_c^{|\eta|}, \quad \forall \eta \in X^*,$$

then the operator is well defined over $[0, T]$ for all $R, T > 0$. These are called *globally convergent* series, and the set of all such series is denoted by $\mathbb{R}_{GC}^\ell\langle\langle X\rangle\rangle$. A Fliess operator F_c defined on $B_\mathfrak{p}^m(R)[t_0, t_0 + T]$ is said to be *realized* by a state space realization

$$\dot{z} = g_0(z) + \sum_{i=1}^{m} g_i(z)\, u_i, \quad z(t_0) = z_0, \quad y = h(z),$$

where each g_i is an analytic vector field expressed in local coordinates on some neighborhood $\mathcal{W} \subseteq \mathbb{R}^n$ of z_0, and h is an analytic function on $\mathcal{W}$, if the state equation has a well defined solution $z(t)$, $t \in [t_0, t_0 + T]$ on $\mathcal{W}$ for any given input $u \in B_\mathfrak{p}^m(R)[t_0, t_0 + T]$, and $F_c[u](t) = h(z(t))$, $t \in [t_0, t_0 + T]$. In this case, the coefficients of the i-th component of generating series c are computed by

$$(c_i, \eta) = L_{g_\eta} h_i(z_0), \quad \eta \in X^*, \tag{1}$$

where $L_{g_\eta} h_i := L_{g_{j_1}} \cdots L_{g_{j_k}} h_i$, $\eta = x_{j_k} \cdots x_{j_1}$ is the *iterated Lie derivative* of h_i with respect to $g_{j_1}, \ldots, g_{j_k}$.

It is well known that the interconnection of two Fliess operators in a parallel-product fashion can be described via shuffle product, namely, $F_c F_d = F_{c \sqcup\!\sqcup d}$. The cascade connection $F_c \circ F_d$ also has a Fliess operator representation, $F_{c \circ d}$, where $c \circ d$ is the *composition product* of c and d as described in [11]. In the event that two Fliess operators are interconnected to form a feedback system, the closed-loop system has a Fliess operator representation whose generating series is best characterized in terms of Hopf algebra methods using the operator group $\mathscr{F}_\delta = \{I + F_c \ : \ c \in \mathbb{R}^m\langle\langle X\rangle\rangle\}$ under composition with I representing the unit. The latter is often denoted in terms of the symbol δ, which is the (fictitious) generating series satisfying $F_\delta := I$ so that $I + F_c := F_{\delta + c} = F_{c_\delta}$ with $c_\delta := \delta + c$. The set of all such generating series for $\mathscr{F}_\delta$ is denoted by $\mathbb{R}\langle\langle X_\delta\rangle\rangle$. This set also forms a group under a composition product induced by operator composition, namely, $c_\delta \circ d_\delta := \delta + d + c \,\tilde{\circ}\, d$, where $\tilde{\circ}$ denotes the *modified composition product* [11]. The group $(\mathbb{R}\langle\langle X_\delta\rangle\rangle, \circ, \delta)$ has coordinate functions that form a Faà di Bruno type Hopf algebra. In which case, the group (composition) inverse $c_\delta^{\circ -1}$ can be computed efficiently via the antipode of this Hopf algebra [4], [11].

B. Left Inversion of Multivariable Fliess Operators

When $c \in \mathbb{R}_{LC}^\ell\langle\langle X\rangle\rangle$, the operator F_c is known to map every input u which is analytic at t_0 to an output y which is also analytic at t_0 [21]. The left inverse of F_c can be computed via an explicit formula described next [9], [12]. Assume $t_0 = 0$ without loss of generality, and observe that any c can written as $c = c_N + c_F$, where $c_N := \sum_{k \geq 0} (c, x_0^k) x_0^k$ and $c_F := c - c_N$. The following definition gives a sufficient condition for the left inverse of F_c to exist.

Definition 2.1: Given $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$, let $r_i \geq 1$ be the largest integer such that $\mathrm{supp}(c_{F,i}) \subseteq x_0^{r_i - 1} X^*$, where $i = 1, 2, \ldots, m$. Then the component series c_i has *relative degree* r_i if the linear word $x_0^{r_i - 1} x_j \in \mathrm{supp}(c_i)$ for some $j \in \{1, \ldots, m\}$, otherwise it is not well defined. In addition, c has *vector relative degree* $r = [r_1\ r_2\ \cdots\ r_m]$ if each c_i has relative degree r_i and the $m \times m$ matrix

$$A = \begin{bmatrix} (c_1, x_0^{r_1-1}x_1) & (c_1, x_0^{r_1-1}x_2) & \cdots & (c_1, x_0^{r_1-1}x_m) \\ (c_2, x_0^{r_2-1}x_1) & (c_2, x_0^{r_2-1}x_2) & \cdots & (c_2, x_0^{r_2-1}x_m) \\ \vdots & \vdots & \vdots & \vdots \\ (c_m, x_0^{r_m-1}x_1) & (c_m, x_0^{r_m-1}x_2) & \cdots & (c_m, x_0^{r_m-1}x_m) \end{bmatrix}$$

has full rank. Otherwise, the vector relative degree of c is not defined.

It can be shown directly that vector relative degree as defined above is consistent with the geometric notion given in the state space setting [16].

Next note that any series $C \in \mathbb{R}^{m \times m}\langle\langle X\rangle\rangle$ with $\det((C, \emptyset)) \neq 0$ has a shuffle inverse given by

$$C^{\sqcup\!\sqcup -1} = ((C, \emptyset)(I - C'))^{\sqcup\!\sqcup -1} = (C')^{\sqcup\!\sqcup *} (C, \emptyset)^{-1},$$

where $C' = I - (C, \emptyset)^{-1} C$ is proper, i.e., $(C', \emptyset) = 0$, and $(C')^{\sqcup\!\sqcup *} := \sum_{k \geq 0} (C')^{\sqcup\!\sqcup k}$. The relationship between $C^{\sqcup\!\sqcup -1}$ and the multiplicative inverse operator $(F_C)^{-1}$, that is, $F_C (F_C)^{-1} = (F_C)^{-1} F_C = I$, is $(F_C)^{-1} = F_{C^{\sqcup\!\sqcup -1}}$. Let $X_0 := \{x_0\}$, and $\mathbb{R}^m[[X_0]]$ denotes the set of all commutative series over X_0. When $c \in \mathbb{R}^m[[X_0]]$, $F_c[u](t)$ is equivalent to the Taylor series $\sum_{k \geq 0} (c, x_0^k) E_{x_0^k}[u](t) = \sum_{k \geq 0} (c, x_0^k) t^k / k!$. The formula for left inversion is given next.

Theorem 2.1: Suppose $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$ has vector relative degree r. Let y be analytic at $t = 0$ with generating series $c_y \in$

$\mathbb{R}^m_{LC}[[X_0]]$ satisfying $(c_{y_i}, x_0^k) = (c_i, x_0^k)$, $k = 0, 1, \ldots, r_i - 1$, $i = 1, 2, \ldots, m$. Then the input

$$u(t) = \sum_{k=0}^{\infty} (c_u, x_0^k) \frac{t^k}{k!}, \tag{2}$$

is the unique real analytic solution to $F_c[u] = y$ on $[0, T]$ for some $T > 0$, where

$$c_u = \left(\left[C^{\sqcup\!\sqcup\, -1} \sqcup\!\sqcup\, (x_0^r)^{-1}(c - c_y) \right]^{\circ -1} \right)_N, \tag{3}$$

the i-th row of $(x_0^r)^{-1}(c - c_y)$ is $(x_0^{r_i})^{-1}(c_i - c_{y_i})$, and the (i, j)-th entry of C is $(x_0^{r_i - 1} x_j)^{-1}(c_i)$.

III. Analysis of Truncation Error and Execution Times for Fliess Operators

Let the truncation of $F_c[u](t)$ to order N be

$$F_c^N[u](t) := \sum_{j=0}^{N} \sum_{\eta \in X^j} (c, \eta) E_\eta[u](t).$$

Then the truncation error is

$$\left| F_c[u](t) - F_c^N[u](t) \right| = \left| \sum_{j=N+1}^{\infty} \sum_{\eta \in X^j} (c, \eta) E_\eta[u](t) \right|.$$

If the applied input u is bounded in the sense that there is a constant $M_u > 0$ such that $|u(t)| \leq M_u$, then $|E_\eta[u](t)| \leq M_u^{|\eta|} t^{|\eta|} / |\eta|!$, $\forall \eta \in X^*$. When $c \in \mathbb{R}^\ell_{LC}\langle\langle X \rangle\rangle$, it follows from [10, Theorem 6] that the output error is bounded as:

$$\left| F_c[u](t) - F_c^N[u](t) \right| \leq e(N) := \frac{s^{N+1}}{1 - s}, \tag{4}$$

where $s := M_c M_u t (m + 1) < 1$. Denoting by ϵ the desired upper bound, a maximum execution time t_ϵ can be obtained from the smallest positive root, say s_ϵ, of the polynomial $K_c s^{N+1} + s\epsilon - \epsilon = 0$, namely,

$$t_\epsilon = \frac{s_\epsilon}{(M_c M_u (m + 1))}. \tag{5}$$

By Descartes' rule of signs this polynomial always has a positive real root. A similar analysis holds for the case of globally convergent series. Specifically, if $c \in \mathbb{R}^\ell_{GC}\langle\langle X \rangle\rangle$, then from [10, Theorem 7] it follows that

$$\left| F_c[u](t) - F_c^N[u](t) \right| \leq e(N) := K_c \, \mathrm{e}^s (1 - \Gamma(N + 1, s)), \tag{6}$$

where $\Gamma(a, b) := \int_b^\infty t^{a-1} \mathrm{e}^{-t} \, dt / \Gamma(a)$ is the (upper) incomplete Gamma function. Denoting again the desired error bound by $\epsilon \leq K_c$ and defining s as in the locally convergent case, one now has to solve the transcendental equation

$$K_c \, \mathrm{e}^s (1 - \Gamma(N + 1, s)) - \epsilon = 0 \tag{7}$$

to determine s_ϵ. Let N be fixed and $G(s) := K_c \, \mathrm{e}^s (1 - \Gamma(N + 1, s))$ restricted to $[0, \infty]$. For $s \in [0, \infty]$, the function $(1 - \Gamma(N + 1, s))$ decreases smoothly and monotonically from 1 to 0. Now, there are always $s_2 > s_1 > 0$ such that $G(s_1) > \epsilon$ and $G(s_2) < \epsilon$. Since $G(s)$ is smooth, then there exists a positive solution s_ϵ of (7). With the help of Mathematica function `FindRoot`, one can find this positive real solution s_ϵ of (7). The execution time is then computed from (5).

The next theorem describes exactly how much of the Taylor series of u is needed to exactly produce $F_c^N[u]$ for any fixed $N \geq 1$.

Theorem 3.1: For $c \in \mathbb{R}^\ell_{LC}\langle\langle X \rangle\rangle$ and $N \geq 1$, it follows for any input u which is analytic at $t = 0$ that $F_c^N[u] = F_c^N[u^{N-1}]$ on some interval $[0, T]$ with $T > 0$, where $u^{N-1}(t) := \sum_{n=0}^{N-1} (c_u, x_0^n) \, t^n / n!$.

The proof follows directly from results in [13], which are summarized next, in light of the fact that $y = F_c[u]$ is equivalent to $c_y = c \circ c_u$ for analytic inputs.

Lemma 3.1: Let $c \in \mathbb{R}^\ell\langle\langle X \rangle\rangle$ and $c_u \in \mathbb{R}^m[[X_0]]$. Then for any $n \geq 0$

$$(c \circ c_u, x_0^n) = (c, x_0^n) + \sum_{k=1}^{n} \sum_{\substack{i_1, \ldots, i_k = 1 \\ j_1, \ldots, j_k = 0}}^{m, n} \left(c, \bar{P}^{j_1 \cdots j_k}_{i_1 \cdots i_k}(n) \right) (c_{u_{i_1}}, x_0^{j_1}) \cdots (c_{u_{i_k}}, x_0^{j_k}),$$

where c_{u_i} is the i-th component series of c_u and

$$\bar{P}^{j_1 \cdots j_k}_{i_1 \cdots i_k}(n) = \sum_{n_0, \ldots, n_k \geq 0} \chi^{j_1 \cdots j_k}_{n_0 \cdots n_k}(n) \, x_0^{n_k} x_{i_k} \cdots x_0^{n_1} x_{i_1} x_0^{n_0}$$

is a polynomial with coefficients

$$\chi^{j_1 \cdots j_k}_{n_0 \cdots n_k}(n) = \Big(x_0^{n_k + 1} [x_0^{j_k} \sqcup\!\sqcup\, [x_0^{n_{k-1}+1} [x_0^{j_{k-1}} \sqcup\!\sqcup \cdots x_0^{n_1 + 1} [x_0^{j_1} \sqcup\!\sqcup\, x_0^{n_0}] \cdots]]], x_0^n \Big).$$

An alternative identity for $(c \circ c_u, x_0^n)$ can be deduced from Lemma 3.1 by introducing an ordering on the coefficients $(c_{u_{i_1}}, x_0^{j_1}) \cdots (c_{u_{i_k}}, x_0^{j_k})$. For each $k \geq 1$ define the set of $2 \times k$ matrices

$$S_k = \left\{ \begin{pmatrix} j_1 & j_2 & \cdots & j_k \\ i_1 & i_2 & \ldots & i_k \end{pmatrix} : 1 \leq i_l \leq m, \; j_l \geq 0, \; (1, 0) \leq (i_1, j_1) \leq \cdots \leq (i_k, j_k) \right\},$$

where "$\leq$" denotes the lexicographic order on the set $\{(i, j) : i, j \in \mathbb{N}\}$. Define the positive integers $s_1, \ldots, s_p$ for a given element of S_k by

$$\begin{pmatrix} j_1 & \cdots & j_k \\ i_1 & \ldots & i_k \end{pmatrix} = \begin{pmatrix} \beta_1 \cdots \beta_1 & \cdots & \beta_p \cdots \beta_p \\ \underbrace{\alpha_1 \cdots \alpha_1}_{s_1} & \cdots & \underbrace{\alpha_p \cdots \alpha_p}_{s_p} \end{pmatrix}.$$

Using this ordering, the lemma below follows naturally.

Lemma 3.2: Let $c \in \mathbb{R}^\ell\langle\langle X \rangle\rangle$ and $c_u \in \mathbb{R}^m[[X_0]]$. Then

$$(c \circ c_u, x_0^n) = (c, x_0^n) + \sum_{k=1}^{n} \sum_{S_k} \frac{1}{s_1! \cdots s_p!} \left(c, P^{j_1 \cdots j_k}_{i_1 \cdots i_k}(n) \right) (c_{u_{i_1}}, x_0^{j_1}) \cdots (c_{u_{i_k}}, x_0^{j_k}),$$

where the inner sum is taken over all elements of S_k such that $k + j \leq n$ and

$$P^{j_1 \cdots j_k}_{i_1 \cdots i_k}(n) := \sum_{\sigma \in \Pi_k} \bar{P}^{j_{\sigma(1)} \cdots j_{\sigma(k)}}_{i_{\sigma(1)} \cdots i_{\sigma(k)}}(n),$$

where Π_k denotes the permutation group on $\{1, 2, \ldots, k\}$.

A more compact form of the identity above is possible if one associates a family of polynomials in $\mathbb{R}\langle X \rangle$ with each $c_u \in \mathbb{R}^m[[X_0]]$:

$$P_{c_u}(n) := x_0^n + \sum_{k=1}^{n} \sum_{S_k} \frac{1}{s_1! \cdots s_p!} P^{j_1 \cdots j_k}_{i_1 \cdots i_k}(n) (c_{u_{i_1}}, x_0^{j_1}) \cdots (c_{u_{i_k}}, x_0^{j_k}),$$

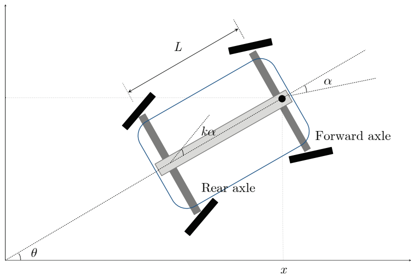

Fig. 1. Bi-steerable car

$n \geq 0$. Therefore, $c_y = c \circ c_u = \sum_{n=0}^{\infty} (c, P_{c_u}(n))\, x_0^n$, where $\deg(P_{c_u}(n)) = n$ and $(c \circ c_u, x_0^n) = (c, P_{c_u}(n))$, for $n \geq 0$.

The main theorem of the section is presented next. The proof is omitted due to space constraints, nevertheless the proof follows from combining the results above.

Theorem 3.2: Suppose $y = F_c[u]$ and $N \geq 1$ is fixed.

1. If $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$, u is analytic at $t = 0$, and $\epsilon > 0$ is sufficiently small then $y^N := F_c^N[u^{N'}]$ satisfies $\left|y^N(t) - y(t)\right| \leq \epsilon$ on $[0, t_\epsilon]$ for $N' \geq N - 1$, where t_ϵ is given by (5), and s_ϵ is the smallest positive real solution of (4).
2. If $c \in \mathbb{R}^{\ell}_{GC}\langle\langle X\rangle\rangle$, u is entire, and $\epsilon > 0$ is arbitrary sufficiently small then $y^N := F_c^N[u^{N'}]$ satisfies $\left|y^N(t) - y(t)\right| \leq \epsilon$ on $[0, t_\epsilon]$ for $N' \geq N - 1$, where t_ϵ is given by (5), and s_ϵ is the smallest positive real solution of (7).

IV. Left Inverse of Bi-steerable Car Kinematics

Consider the bi-steerable car shown in Figure 1. For simplicity the car is assumed to have zero mass and move in the plane with the speed of the car $u_1 = \sqrt{\dot{x}^2 + \dot{y}^2}$ and the front axle steering angular velocity $u_2 = \alpha$ as inputs. The kinematics of the system are therefore

$$\dot{x} = u_1 \cos(\theta + \alpha), \quad \dot{y} = u_1 \sin(\theta + \alpha),$$
$$\dot{\theta} = u_1 \frac{\sin(\alpha - f(\alpha))}{L \cos(f(\alpha))}, \quad \dot{\alpha} = u_2,$$

where $f(\alpha) := k\alpha$ with $k \in \mathbb{R}$. These dynamics assume the usual constraint of rolling without slippage of the wheels. Setting $z_1 = x$, $z_2 = y$, $z_3 = \theta$, $z_4 = \alpha$, the outputs are picked to be the coordinates of the front axle center $y_i = z_i$, $i = 1, 2$. The corresponding two-input, two-output state space realization is

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{pmatrix} = \begin{pmatrix} \cos(z_3 + z_4) \\ \sin(z_3 + z_4) \\ \frac{\sin((1-k)\alpha)}{L\cos(k\alpha)} \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \tag{8a}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}. \tag{8b}$$

Hereafter, the focus is on the $k = -0.7$ case, which as explained in [19] is sufficient for the existence of flat outputs. However, this fact is inconsequential here since flat outputs can not be directly combined with the trajectory generation algorithm. The outputs in (8), on the other hand, have physical meaning, but they are *not* flat. The generating series, c, can be computed directly from (1) using the vector fields and output function given in (8), however, one can check that the system does *not* have a well defined vector relative degree. Therefore, applying dynamic extension on u_1 yields the augmented system

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \end{pmatrix} = \begin{pmatrix} z_5 \cos(z_3 + z_4) \\ z_5 \sin(z_3 + z_4) \\ z_5 \frac{\sin((1-k)\alpha)}{L\cos(k\alpha)} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_2 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \bar{u}_1, \tag{9}$$

where the new input $\bar{u}_1$ is taken as the derivative of the original input u_1. This system has vector relative degree $r = [r_1\ r_2] = [2\ 2]$ with $r_1 + r_2 = 4 < 5 = n$ and generating series

$$\begin{aligned} c_1 &= z_{1,0} + z_{5,0} \,\mathrm{co}\, x_0 + z_{5,0} \,\mathrm{si}\, x_0 x_1 + \mathrm{co}\, x_0 x_2 \\ &\quad - z_{5,0}^2 \frac{\sin(1.7 z_{4,0})}{\cos(0.7 z_{4,0})} \,\mathrm{si}\, x_0^2 + \cdots \\ c_2 &= z_{2,0} + z_{5,0} \,\mathrm{si}\, x_0 + z_{5,0} \,\mathrm{co}\, x_0 x_1 + \mathrm{si}\, x_0 x_2 \\ &\quad + z_{5,0}^2 \frac{\sin(1.7 z_{4,0})}{\cos(0.7 z_{4,0})} \,\mathrm{co}\, x_0^2 + \cdots, \end{aligned}$$

where $\mathrm{co} := \cos(z_{3,0} + z_{4,0})$ and $\mathrm{si} := \sin(z_{3,0} + z_{4,0})$. In this case, $c \in \mathbb{R}^2_{GC}\langle\langle X\rangle\rangle$ with $X = \{x_0, x_1, x_2\}$ since all sines and cosines in the numerators are bounded by one. Thus, the output $y = F_c[u]$ is well defined on $[0, \infty)$ for all $u \in B_1^m[0,T](R)$ and any finite $R, T > 0$. One way to get additional insight into the convergence characteristics of the generating series for the augmented system is to estimate the geometric growth constants for $c_N := \sum_{k \geq 0} (c, x_0^k) x_0^k$ by fitting $\ln((c_N, \eta))$ as a linear function of $|\eta|$ [8]. This analysis yields the following estimates for the growth constants: $\hat{K}_c = \mathrm{e}^{2.25}$ and $\hat{M}_c = \mathrm{e}^{2.24}$. From the form of the coefficients of c and the fact that they are computed using (1), a conservative choice for the growth constants is $K_c = |\max\{z_{1,0}, z_{2,0}, z_{5,0}\}|$ and $M_c = 2.4\,|z_{5,0} \sec(0.7 z_{4,0})|$. Note that $K_c = 9.5 \approx \hat{K}_c = \mathrm{e}^{2.5}$ and $M_c = 45.77 > \hat{M}_c = \mathrm{e}^{2.24} = 9.39$. The parameter $M_c > \hat{M}_c$ will produce execution times that are shorter (but safer) than what the truncated system allows.

Proceeding with the left-inverse computation, the decoupling matrix for the augmented system is

$$A = (C, \emptyset) = \begin{pmatrix} z_{5,0}\,\mathrm{si} & \mathrm{co} \\ z_{5,0}\,\mathrm{co} & \mathrm{si} \end{pmatrix},$$

which is clearly nonsingular if $z_{5,0} \neq 0$. For a a given output

$$y(t) = \sum_{k=0}^{\infty} (c_y, x_0^k) \frac{t^k}{k!}$$

with generating series $c_y = [c_{y_1}, c_{y_2}]^T$, the left inverse $c_u = [c_{\bar{u}_1}, c_{u_2}]^T$ is computed directly from (2)-(3). Degree three polynomial outputs are sufficient for this application, so let $(c_{y_j}, x_0^i) = v_{ji}$ for $i = 0, 1, 2, 3$ and $j = 1, 2$. The series d is

$$d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} := C^{\sqcup\!\sqcup\, -1} \sqcup\!\sqcup\, (x_0^r)^{-1} (c - c_y),$$

where, for example,

$$\begin{aligned} d_1 &= \frac{v_{12}\,\mathrm{si}}{z_{5,0}} - \frac{v_{22}\,\mathrm{co}}{z_{5,0}} + z_{5,0} \frac{\sin(1.7 z_{4,0})}{\cos(0.7 z_{4,0})} \\ &\quad + \left(\left(\frac{v_{13}}{z_{5,0}} + v_{22} \frac{\sin(1.7 z_{4,0})}{\cos(0.7 z_{4,0})} \right) \mathrm{si} \right. \\ &\quad \left. + \left(v_{12} \frac{\sin(1.7 z_{4,0})}{\cos(0.7 z_{4,0})} - \frac{v_{23}}{z_{5,0}} \right) \mathrm{co} \right) x_0 + \cdots, \end{aligned}$$

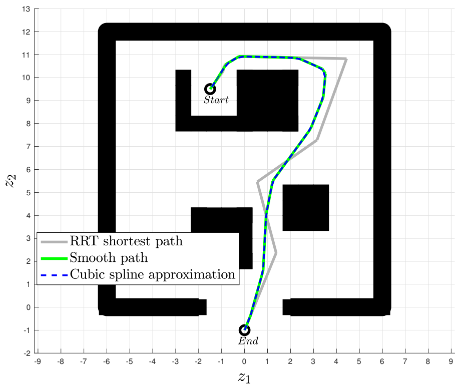

Fig. 2. Shortest, smoothed and spline approximated paths

The final step is to apply the recursive method in [4], [11] to determine $d^{\circ -1}$. The expression for $c_{\bar{u}_1}$, for example, is

$$c_{\bar{u}_1} = \left(d_1^{\circ -1}\right)_N = \frac{v_{22}\mathrm{si}}{z_{5,0}} - \frac{v_{12}\mathrm{co}}{z_{5,0}} - z_{5,0}\frac{\sin(1.7z_{4,0})}{\cos(0.7z_{4,0})}$$
$$+ \frac{1}{z_{5,0}^2}\Bigg(- 2v_{12}v_{22}\cos(2(z_{3,0}+z_{4,0})) + \cdots$$
$$- z_{5,0}\frac{\sin(1.7z_{4,0})}{\cos(0.7z_{4,0})}(v_{12} + 0.7v_{22}\tan(0.7z_{4,0}))\Bigg)\Bigg) x_0 + \cdots .$$

Some key features of the composition inverse of d are:

i. The formulas are exact if not truncated.

ii. The formulas only need to be computed once. Then based on measurements of the current position and steering angle, the input for tracking the next section of the desired path can be quickly computed by just numerically evaluating the formula.

iii. Flat outputs are not required for computing these left inverses.

iv. One can increase the degree of approximation of the output tracking by including more input and output terms if the computing power is available.

In practice, only truncations of $c_{\bar{u}_1} = (d_1^{\circ -1})_N$ and $c_{u_2} = (d_2^{\circ -1})_N$ can be utilized. Let c be the generating series of system (9). Given an output error bound $\epsilon > 0$ and a polynomial $c_y = \sum_{k=0}^{N}(c_y, x_0^k)x_0^k$ in the range of the operator F_c, the corresponding input (2) is truncated to degree N' as in Theorem 3.1. For $\eta \in X^*$, the i-th coordinate function applied to $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$ is $a^i_\eta(c) := (c_i, \eta)$. A direct evaluation of the antipode formula in [4], [11] gives

$$a^i_{\emptyset}(d^{\circ -1}) = (Sa^i_{\emptyset})(d) = -(d_i, \emptyset)$$
$$a^i_{x_0}(d^{\circ -1}) = (Sa^i_{x_0})(d) = -(d_i, x_0) + \sum_{\ell=1}^{2}(d_i, x_\ell)(d_\ell, \emptyset)$$
$$\vdots$$

where S denotes the antipode operator. Now, due to the restriction given by the range of the operator F_c in Theorem 2.1 and the fact that the vector relative degree of c is $r = [2, 2]$, the following conditions must hold for the initial conditions $(z_{1,0}, z_{2,0}, z_{3,0}, z_{4,0}, z_{5,0})$ of (9) in order for $F_c[u]$ to track the desired output y:

$$(c_1, \emptyset) = (c_{y_1}, \emptyset),\ (c_2, \emptyset) = (c_{y_2}, \emptyset), \tag{10}$$
$$(c_1, x_0) = (c_{y_1}, x_0),\ \text{and}\ (c_2, x_0) = (c_{y_2}, x_0). \tag{11}$$

The conditions in (10) are easily satisfied since one can choose to start tracking at the initial conditions defined for the system. This is equivalent to

$$(c_1, \emptyset) = (c_{y_1}, \emptyset) = z_{1,0},\ (c_2, \emptyset) = (c_{y_2}, \emptyset) = z_{1,0}. \tag{12}$$

On the other hand, thanks to the dynamic extension procedure applied to the system, the initial conditions $z_{4,0}$ and $z_{5,0}$ constitute free parameters that can be chosen so that (11) always hold. This reduces to $(c_1, x_0) = z_{5,0}\,\mathrm{co}$, and $(c_2, x_0) = z_{5,0}\,\mathrm{si}$. Assuming $z_{3,0}$ is given, this system of equations can always be solved for $z_{4,0}$ and $z_{5,0}$ since it is the same as computing an inverse polar coordinates transformation. Thus,

$$z_{4,0} = \arctan\left(\frac{(c_2, x_0)}{(c_1, x_0)}\right) - z_{3,0} \tag{13}$$
$$z_{5,0} = \sqrt{(c_1, x_0)^2 + (c_2, x_0)^2}. \tag{14}$$

Finally, a direct application of Theorem 3.2 gives $N' = N - 1$ as the minimum value that can make $|F_c^N[u^{N'}](t) - y(t)| \leq \epsilon$ for $t \in [0, t_\epsilon]$, where $t_\epsilon = s_\epsilon / 3M_cM_u$, and s_ϵ is the smallest positive real solution of (6).

V. A CASE OF STUDY OF THE BI-STEERABLE CAR

A recent description of the state of the art in using flatness for wheeled vehicles can be found in [1]. Unfortunately, flat outputs are sometimes not the most desirable from a physical point of view. This turns out to be the case for the bi-steerable car, where the flat outputs do not correspond to simple physical variables like a center point on the vehicle, its speed or orientation [15], [19], [20]. This then makes integrating the tracking problem with the trajectory generation problem more complex. In this section, an algorithm for integrating the truncated left-inverse of the bi-steerable car model with the trajectory generation provided by the RRT procedure is described. The successful application of the left-inverse formula relies on the fact that when physical variables are used as outputs, the restrictions on the range of the Fliess operator representing the system are directly used in the algorithm design. The simulation time is set to 1 since the model is dimensionless. The steps of the algorithm are:

i. Map the area and obstacles where the bi-steerable car moves and provide a *start* location and an *end* location.

ii. The RRT trajectory generation algorithm described in [17] computes the shortest path between the start and end locations.

iii. The path is smoothed and approximated by a cubic spline.

iv. Each spline section, which is a cubic polynomial, is fed to the left inversion formula in order to compute the corresponding control inputs $\bar{u}_1$ and u_2.

v. Truncated versions of $\bar{u}_1$ and u_2 are applied to the car.

Consider the following specific example:

i. Consider the area shown in Figure 2. Pick the start and end locations to be: $Start: (-1.5, 9.5),\ \ End: (0, -1)$.

ii. The shortest path found by RRT is shown in Figure 2.

iii. The path is then smoothed and approximated by cubic splines as shown in Figure 2. The smoothing process is based on a divide and conquer routine that takes in consideration a circle of radius $r = 0.3$ around the position of the car so that the smoothed path does not violate the boundaries defined by the obstacles [2].

iv. The control inputs are now computed for each spline piece using left inversion. The first spline section has initial

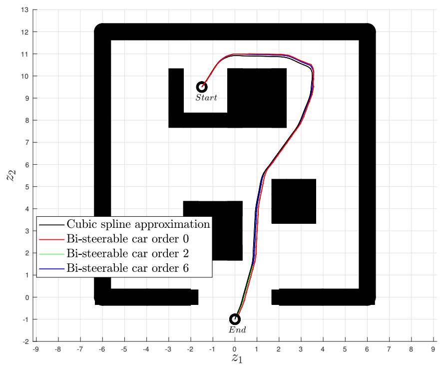

Fig. 3. Cubic spline approximated path and bi-steerable car trajectory for piecewise constant and degree six cases

conditions $z_{1,0} = -1.5$, $z_{2,0} = 9.5$, $z_{3,0} = 1.5$, and the desired outputs of the path are

$$c_{y_1} = -1.5 + 10.75x_0 - 28.15x_0^2 + 185.15x_0^3$$
$$c_{y_2} = 9.5 + 14.40x_0 + 13.07x_0^2 - 24.89x_0^3.$$

The fact that the constant terms for each spline section gives the actual position of the bi-steerable car is represented by condition (12), which is key for the direct integration of the left inverse and the trajectory generation procedure. Also, (13) and (14) ensure that $(c_1, x_0) = (c_{y_1}, x_0) = 10.75$ and $(c_2, x_0) = (c_{y_2}, x_0) = 14.40$, which gives $z_{4,0} = -4.00028$, and $z_{5,0} = -17.97$. The left inverse to degree two, u^2, is

$$\bar{u}_1^2(t) = 7.74 - 195.5t + 1592.2t^2,$$
$$u_2^2(t) = 6.37 - 142.17t + 386.63t^2.$$

The corresponding output error functions are

$$y_1(t) - y_1^2(t) = -6940.62t^5 + 144186t^6 - 612522t^7 + 1.33246 \times 10^6 t^8 + \cdots, \quad (15a)$$
$$y_2(t) - y_2^2(t) = 9315.36t^5 - 165629.t^6 + 1.84881 \times 10^6 t^7 - 7.44982 \times 10^6 t^8 + \cdots. \quad (15b)$$

From Theorem 3.2, an estimate of the execution time for u^2 can be computed using $K_c = 2.25$, $N = 3$, $\epsilon = 0.2$, and $\hat{M}_c = \mathrm{e}^{2.24} = 9.39$. Thus, $t_\epsilon = s_\epsilon/(3M_cM_u) = 0.015$. While conservative, this estimate ensures that the system does not produce an error beyond $\epsilon = 0.2$ for *every section*. On the other hand, one can compute an empirical execution time directly from (15) for the *first section* as $\hat{t}_\epsilon = 0.49$, which is much greater than t_ϵ. A pragmatic solution for the simulation time was found to be $t_{sim} = 0.02$. Employing the sixth order truncation of u in (2) gives an empirical execution time of $\hat{t}_\epsilon = 0.711$. As expected, higher degree of approximation gives longer empirical execution times. However, the average residual errors of the overall simulation (next step) does not significantly differ from the quadratic case.

v. The truncated control input $u^N = [\bar{u}_1^N, u_2^N]^T$ for $N = 0, 2, 6$ was used to drive the bi-steerable car as shown in Figure 3. The corresponding average errors against the cubic spline approximation of the path produced by the RRT algorithm are 0.2007, 0.144 and 0.141, respectively. Thus, the most appropriate input for the bi-steerable car is u^2 as it gives a good balance between accuracy and execution time, as expected from Theorem 3.2. That is, u^6 produces only a small improvement in tracking performance for the additional computation expense.

VI. Conclusions

A methodology for integrating a Fliess operator based left-inverse output tracking with an out-of-the-box trajectory generation algorithm was provided for generating series having well defined vector relative degree. A characterization of errors due to series truncation was provided, which allows one to compute execution times for truncated Fliess operators. A case study was presented for the bi-steerable car.

References

[1] B. d'Andréa-Novel, L. Menhour, M. Fliess and H. Mounier, Some remarks on wheeled autonomous vehicles and the evolution of their control design, *IFAC-PapersOnLine - 9th IFAC Symposium on Intelligent Autonomous Vehicles*, 49 (2016) 199–204.

[2] S. Carpin and G. Pillonetto, Motion planning using adaptive random walks, *IEEE Transactions on Robotics*, 21 (2005) 129–136.

[3] A. de Luca, G. Oriolo, and M. Vendittelli, Control of wheeled mobile robots. Experimental overview, in *RAMSETE: Articulated and Mobile Robots for SErvices and TEchnologies*, S. Nicosia, B. Siciliano, A. Bicchi, and P. Valigi, Eds., Springer, London, 2001, pp. 181–226.

[4] L. A. Duffaut Espinosa, K. Ebrahimi-Fard, and W. S. Gray, A combinatorial Hopf algebra for nonlinear output feedback control systems, *Journal of Algebra*, 453 (2016) 609–643.

[5] H. W. Engl and P. Kügler, Nonlinear inverse problems: Theoretical aspects and some industrial applications, in *Multidisciplinary Methods for Analysis, Optimization and Control of Complex Systems*, V. Capasso and J. Periaux, Eds., Springer, Heidelberg, 2005 pp. 3–48.

[6] M. Fliess, J. Lévine, P. Martine and P. Rouchon, Flatness and defect of non-linear systems: introductory theory and examples, *International Journal of Control*, 61 (1995) 1327–1361.

[7] N. H. Getz, *Dynamic Inversion of Nonlinear Maps with Applications to Nonlinear Control and Robotics*, Ph.D. dissertation, UC Berkeley, Berkeley, CA, 1996.

[8] W. S. Gray and L. A. Duffaut Espinosa, A Faà di Bruno Hopf algebra for analytic nonlinear feedback control systems, in *Faà di Bruno Hopf Algebras, Dyson-Schwinger Equations, and Lie-Butcher Series*, K. Ebrahimi-Fard and F. Fauvet, Eds., IRMA Lect. Math. Theor. Phys., Eur. Math. Soc., Zürich, Switzerland, 2015.

[9] W. S. Gray, L. A. Duffaut Espinosa and K. Ebrahimi-Fard, Analytic left inversion of multivariable Lotka-Volterra models, *Proc. 54th IEEE Conf. on Decision and Control*, Osaka, Japan, 2015, pp. 6472–6477.

[10] ——, Discrete-time approximations of Fliess operators, *Numerische Mathematik*, 137 (2017) 35–62.

[11] ——, Faà di Bruno Hopf algebra of the output feedback group for multivariable Fliess Operators, *Systems & Control Letters*, 74 (2014) 64–73.

[12] W. S. Gray, L. A. Duffaut Espinosa, and M. Thitsa, Left inversion of analytic nonlinear SISO systems via formal power series methods, *Automatica*, 50 (2014) 2381–2388.

[13] W. S. Gray and Y. Wang, Formal Fliess operators with applications to feedback interconnections, *Proc. 18th International Symposium on the Mathematical Theory of Networks and Systems*, Blacksburg, Virginia, 2008.

[14] A. Hars, Autonomous cars: The next revolution looms, *Inventivio Innovation Briefs 2010-01*, Nuremberg, 2010, www.inventivio.com/innovationbriefs/2010-01.

[15] J. Hermosillo and S. Sekhavat, Feedback control of a bi-steerable car using flatness application to trajectory tracking, *Proc. 2003 American Control Conference*, Denver, CO, 2003, pp. 3567–3572.

[16] A. Isidori, *Nonlinear Control Systems*, 3rd Ed., Springer-Verlag, London, 1995.

[17] S. LaValle, *Planning Algorithms*, Cambridge University Press, New York, 2006.

[18] J. Lévine, *Analysis and Control of Nonlinear Systems: A Flatness-based Approach*, Springer, Berlin, 2009.

[19] S. Sekhavat and J. Hermosillo, Cycab bi-steerable cars: a new family of differentially flat systems, *Advanced Robotics*, 16 (2002) 445–462.

[20] S. Sekhavat, P. Rouchon, and J. Hermosillo, Computing the flat outputs of Engel differential systems, the case study of the bi-steerable car, *Proc. 2001 American Control Conference*, Arlington, VA, 2001, pp. 3576–3581.

[21] Y. Wang, *Differential Equations and Nonlinear Control Systems*, Ph.D. dissertation, Rutgers University, New Brunswick, NJ, 1990.