2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

On the Calculation of the Equilibrium Points of a Nonlinear System via Exact Quadratization

Francesco Carravetta

***Abstract*— We show that, for quadratizable dynamic nonlinear systems, all the equilibrium points satisfy an augmented system of nonlinear equations obtained by applying exact quadratization to a suitably modified dynamic system. For autonomous polynomial dynamic systems (i.e. with no input) this method can be viewed as a general method of solving systems of polynomial equations, which has a lower computational complexity with respect to the classical one consisting of the Buchberger's algorithm (that searches for a Groebner basis) followed by variable elimination. As a matter, we show that the augmented system of polynomials obtained by quadratization is always a Groebner basis for the ideal associated to the originary problem. This allows skipping the computationally heavy Buchberger's algorithm, and applying directly elimination theory in the solution-searching algorithm.**

I. INTRODUCTION

Let us consider a nonlinear system:

$$\dot{x} = f(x,u), \qquad f \in \mathcal{F}^n_{n+p}, \quad n,p \in \mathbb{N} \tag{1}$$

where $\mathcal{F}^n_{n+p}$ is a set of $\Re^n$-valued functions, defined each on some open domain $D_f \subset \Re^{n+p}$ (which depends of f in general), and such that existence of a solution $t \mapsto (x(t),u(t))$ passing through any point $(\bar{x},\bar{u}) \in D_f$ is assured for any t in an interval $I \subset \Re$ around the origin, as well as unicity of the map $t \mapsto x(t)$ for any map $t \mapsto u(t)$ in some subset of functions $u : I \to \Re^p$, namely $\mathcal{U}_p$, such that $(x(I),u(I)) \subset D_f$. We can, for instance, consider the set $\mathcal{C}^n_{n+p} = \mathcal{C}_{n+p} \times \overset{\cdot}{\cdot}\cdot \times \mathcal{C}_{n+p}$ (n times), where $\mathcal{C}_{n+p}$ is the set of all real functions $f : D_f \to \Re$ differentiable on D_f, and for $\mathcal{U}_p = \mathcal{C}^p_1$, with $0 \in D_\phi \subset \Re$ for any $\phi \in U_p$. Then, of course, we have $\mathcal{C}^n_{n+p} \subset \mathcal{F}^n_{n+p}$. The set of all equilibrium points of system (1), and the corrisponding inputs u (necessarily constant in time) is the set of all real solutions of the system of equations:

$$f(x,u) = 0. \tag{2}$$

By defining $\mathbf{x} = (x,u)$, that is: $\mathbf{x}_i = x_i$, for $i = 1,\dots,n$, and $\mathbf{x}_{n+j} = u_j$, for $j = 1,\dots,p$, eq. (2) rewrites

$$f(\mathbf{x}) = 0, \tag{3}$$

and searching for the pairs $(\bar{x},\bar{u})$, giving equilibrium points and the associated (constant) inputs, can be viewed as the problem of finding the hypersurfaces in $\mathbb{R}^{n+p}$ described by functions of the type $\psi : U \to \Re^n$ (there may be indeed many of such functions ψ for a given f), such that $D'_f = (U,\psi(U)) \subset D_f$, and (3) is satisfied for any $\mathbf{x} \in D'_f$. In Algebraic Geometry, for $\mathcal{F}^n_{n+p} = \Re[\mathbf{x}]$, i.e. for polynomials in $\mathbf{x}$ with coefficients in $\Re$, the union of all hypersurfaces in $\Re^{n+p}$ that are zeros of (3) is named: a (classic) *algebraic variety*[1], and is denoted: $\mathcal{V}(S)$, where $S = \{f_1,\dots,f_n\}$, is the *generator* of the variety, and $f_i \in \Re[\mathbf{x}]$ are the components of f in (3). As well known (and not difficult to show) $\mathcal{V}(S) = \mathcal{V}(\mathcal{I}_S)$ where $\Re[\mathbf{x}] \supset \mathcal{I}_S = (S)$, with (S) denoting the ideal generated by S in $\Re[\mathbf{x}]$, i.e. the set of all linear compositions of the elements of S, with coefficients in $\Re[\mathbf{x}]$. Moreover, by the Hilbert's basis Theorem, $\Re[\mathbf{x}]$ is a Noetherian ring, that is to say: all ideals in $\Re[\mathbf{x}]$ are finitely generated (they have a finite generator). This entails the existence of minimals generators, say σ, for any ideal $\mathcal{I} \subset \Re[x]$, i.e. $\mathcal{I} = (\sigma)$, which have the property: any element in σ cannot be written as a linear composition (with coefficients in $\Re[x]$) of the other elements of σ (otherwise, denoting by f such element, and defining $\sigma' = \sigma \setminus \{f\}$, we would have $(\sigma) = (\sigma')$). In the particular case of $p = 0$ (i.e. no input) we have $\mathbf{x} = x$, and (3) is a system of n equations in n indeterminates. This still doesn't mean neither that a solution of (3) exists nor that it is unique, nonetheless, for $n = 1$, the set of all solutions is always finite, i.e. consists of r points in $\Re$ with $r \in \mathbb{N}$, $r \leq n$, ($r = 0$ means no real solutions) [2] which means that the hypersurfaces determined by ψ_i are single points, that are the *irreducible* components of the algebraic variety defined by (3). Since $(S) = (S,g)$, if g is a linear composition of elements in S, one can use linear compositions in order to change the leading terms of the polynomials of the generator, in a way that we will briefly recall in the next section. This leads to the so called *elimination theory*, which is the core of computer algebra methods for solving systems of polynomial equations, and is in fact an extension of the Gauss-elimination method for linear systems. However, elimination theory has a limitation: it only applies to a *particular kind* of generators that are known in Algebra as *Groebner basis*, whose definition we will recall in the next section, as well as the more known method for testing whether a given generator is a Groebner basis, namely: the Buchberger's algorithm, which works in such a way to modify and/or add new polynomials to the initial set of generators in such a way that finally (the

F. Carravetta is with Institute of Systems Analysis and Computer Science 'A. Ruberti', National Research Council, Roma 00185, Italy. francesco.carravetta@iasi.cnr.it

[1] In some textbooks of Algebraic Geometry, such sets are named *algebraic sets*, whereas the name 'variety' is reserved to *irreducible* algebraic sets, i.e., roughly speaking, sets of zeroes constituting a *single* hypersurface.

[2] We limit our discussion to real solutions, which are those of interest for control systems. Nonetheless, there would be no problems in considering system (1) in $\mathbb{C}$.

algorithm is proven to terminate in a finite number of steps) always outputs a set of generators that constitute a Groebner basis. The Buchberger's algorithm may be computational expensive (it may be super-exponental in the number of indeterminates), and is a part of all the classic algorithms for solving polynomial systems of equations.

The present paper situates within an algebraic approach to problems arising in systems and control theory, and in particular uses some elementary concept of algebraic geometry and polynomial algebra, that can be found, for instance, in [1], [2], [3]. In a wider sense the paper context is represented by the differential-algebraic approaches to system theory that can be found in [4], or in [5].

In this paper we show that, for polynomial systems, i.e. for f in (1) given by a vector of polynomials in $\Re[x, u]$, the calculation of all equilibrium points can be performed by replacing the Buchberger algorithm with a much less computationally heavy algorithm based on *Exact Quadratization* (EQ) of the original system (1), a kind of systems immersion that has been introduced in [6], in the context of control systems, further studied in [7] as a general property of ordinary differential equations (ODE), and recently used in the numerical integration of ODE [9]. This algorithm is the main result of the paper, and is described in §III. Numerical details, and/or a comparison with other existing methods, as well as computer simulations, are beyond the scope of the present paper: they will be presented in future publications. We present here a principle scheme for EQ-based equilibrium points calculation, that can be applied for a more general $f \in \mathcal{Q}^n_{n+p} \subset \mathcal{F}^n_{n+p}$, where $\mathcal{Q}^n_{n+p} \supset (\Re[x, u])^n$, is a more general class of *quadratizable* functions (§III-A). Moreover, we will illustrate how to apply the algorithm when the main interest is just to solve a system of polynomial equations not necessarily related to some dynamic system (§IV-A). We finally argue that control problems formulated as stabilization problems around a fixed equilibrium point can be studied and applied on a quadratization of the original system, in that any regulator of the former is a regulator for the latter as well (§IV-C).

To start with, in §II we provide, for the reader ease, some of the basic algebraic notion used in the paper. The details of the Buchberger's algorithm can be found in [2], or in [3], and are here omitted: as a matter, what is important in the paper is the definition of Groebner basis, and just to know that the Buchberger's algorithm is the usual tool used to find it. We also briefly explain (§II-B) EQ, and how it is used in the algorithm. Some geometric argument will be used in order to illustrate how EQ works in the algorithm, for which we refer readers to classic textbook of differential geometry, such as [11]–[13]. For the differential-geometric insight of EQ we also refer readers to [8], and [10].

II. Preliminaries

A. Some basic algebraic notions

We briefly recall some notions of polynomial Algebra. Details and proofs can be found in [2]. Given a polynomial $f \in \Re[x]$, the degree of f is the degree of the monomial having maximum degree out of all the monomials of f. If $m = cx_1^{p_1} \cdots x_n^{p_n}$, with $c \in \Re$ is a monomial of f, the degree of m is $p_1 + \ldots + p_n$. m is said to be *monic* if $c = 1$. Let us denote by $\mathcal{M}(x)$ the set of all monomials in the indeterminates $x_1, \ldots, x_n$. Then, any ordering of the indeterminates: $x_{i_1} > x_{i_2} > \cdots > x_{i_n}$ induces the following total ordering (which is said: *lexicographic*) in $\mathcal{M}(x)$:

$$x_{i_1}^{p_1} \cdots x_{i_n}^{p_n} > x_{i_1}^{q_1} \cdots x_{i_n}^{q_n}, \quad \text{iff} \quad p_i > q_i, \quad \forall i > j; \quad (4)$$

where j is such that $p_i = q_i \ \forall i \leq j$. Given an ordering of the indeterminates we define $LT(f)$, i.e. the *Leading Term* of f, as the monomial of f of highest lexicographic order. Let us consider $f, g_1, \ldots, g_L \in \Re[x]$, we can always divide f by $g_1, \ldots, g_L$ through the *polynomial division algorithm* in order to obtain the following decomposition:

$$f = q_1 g_1 + \ldots + q_L g_L + r$$

for suitable *quotients* $q_1, \ldots, q_l \in \Re[x]$, and *remainder* $r \in \Re[x]$, where any monomial of r is divisible by $LT(g_i)$ for $i = 1, \ldots, L$. However, the remainder r is not unique in general.

If $\mathcal{I}$ is an ideal in $\Re[x]$, $LT(\mathcal{I})$ shall denote the ideal generated by the leading terms of the elements of $\mathcal{I}$, i.e. $LT(\mathcal{I}) = (\{LT(f) : f \in \mathcal{I}\})$. A generator $g_1, \ldots, g_L$ of $\mathcal{I}$ is said to be a *Groebner basis* of $\mathcal{I}$ if $LT(\mathcal{I}) = (LT(g_1), \ldots, LT(g_L))$. A Groebner basis $g_1, \ldots, g_L$ is said to be *reduced* if all g_i's are monic, and all monomials of g_i are not divisible for $LT(g_j)$, for any $i \neq j$. Any ideal $\mathcal{I}$ in $\Re[x]$ has a Groebner basis, and any Groebner basis $\{g_1, \ldots, g_L\}$ can be replaced by a reduced Groebner basis.

The most important property of Groebner basis $g_1, \ldots, g_L$ is that, for any $f \in \Re[x]$ the division algorithm of f by $g_1, \ldots, g_L$ yields a zero remainder if and only if $f \in \mathcal{I} = (g_1, \ldots, g_L)$. As a matter of fact, we stress that if $g_1, \ldots, g_L$ is not a Groebner basis, even if $f \in (g_1, \ldots, g_L)$ the division algorithm of f by $g_1, \ldots, g_L$ could terminate with $r \neq 0$.

The *Buchberger's algorithm*, is the most known algorithm to test whether a given set of generators is a Groebner basis. If not, the Buchberger's algorithm increase the number of generators until they form a Groebner basis. We will return on the Buchberger's algorithm, in particular as for its computational complexity, later on in §IV-B. The Buchberger's algorithm is the basis by which computer algebra programs attempts to solve systems of polynomial equations. As a matter of fact, these programs basically implement steps from *elimination theory*, which is a generalization to polynomial of the well known linear algebra methods based on Gauss-elimination. These algorithms works on reduced Groebner basis, thus, as the Buchberger's algorithm yields a Groebner basis their first step is to get a reduced basis. Next, elimination theory modifies the generator of the ideal associated to the system of equations trying to perform a sort of 'triangularization' that allegedly allow to solve the system by successive substitution.

B. Exact quadratization

Let us consider the system function $f : \mathcal{D}_f \subset \mathbb{R}^{n+p} \to \Re^n$ in (1), and assume $f \in \mathcal{C}^n_{n+p}$. Moreover assume that $\mathcal{D}_f = \mathcal{D}_{f,x} \times \mathcal{D}_{f,u}$, with $\mathcal{D}_{f,u}, \mathcal{D}_{f,u}$, open sets in $\Re^n$, and $\Re^p$ respectively. Then, f is said to be *quadratizable* on the (open) domain $\mathcal{D}'_{f,x} \subset \mathcal{D}_{f,x}$, if there exists an integer $\mu \geq n$, an injective differentiable map $\mathbf{\Phi} : \mathcal{D}'_{f,x} \to \Re^\mu$, and a map

$$B(\cdot,\cdot,\cdot) : \mathcal{D}'_{f,x} \times \mathcal{D}'_{f,x} \times \mathcal{D}_{f,u} \to \Re^\mu, \tag{5}$$

bilinear with respect the first two arguments, such that, if $\Re \supset I \ni t \mapsto (x(t), u(t))$ is a solution of (1) for some interval I, there is another interval $I' \subset I$, such that the differentiable map $I' \ni t \mapsto \Phi(x(t)) = \zeta(t) \to \mathbb{R}^\mu$ satisfies:

$$\dot{\zeta} = B(\zeta, \zeta, u), \tag{6}$$

which is said *a quadratization* of (1). We denote by $\mathcal{Q}^n_{n+p} \subset \mathcal{F}^n_{n+p}$ the class of all quadratizable functions. The meaning of EQ comes from the fact that $\mathbf{\Phi}(\mathcal{D}'_\mathbf{f})$ is a differentiable submanifold of $\Re^\mu$ (in the usual topology), of dimension n, and that the map $\mathbf{\Phi} : \mathcal{D}'_\mathbf{f} \to \mathbf{\Phi}(\mathcal{D}'_\mathbf{f})$ is invertible, thus, any solution $I \ni t \mapsto x(t)$ of (1) can be recovered (at least: partially) from a solution $I' \ni \zeta \mapsto \zeta(t)$ of the quadratization (6), provided that $\zeta(0) = \mathbf{\Phi}(x(0))$, and in fact $x(t) = \mathbf{\Phi}^{-1}(\zeta(t))$, $\forall t \in I'$.

Even though the characterization of the class $\mathcal{Q}^n_{n+p}$ is still an open problem, in [6] a very large subclass $\bar{\mathcal{Q}}^n_{n+p} \subset \mathcal{Q}^n_{n+p}$ has been defined, including all the most known and usual differentiable functions, and such that, for any $f \in \bar{\mathcal{Q}}^n_{n+p}$ the quadratizing map $\mathbf{\Phi}$ can be decomposed as $\mathbf{\Phi} = \mathbf{\Phi}' \circ \mathbf{\Psi}$ where for the generic i-th component $z_i = \mathbf{\Psi}_i(x)$ we have:

$$\dot{z}_i = \sum_{j=1}^{\nu_i} v_{ij}(u) X_{ij}(z); \quad X_{ij}(z) = z_1^{p_{ij1}} \cdots z_\alpha^{p_{ij\alpha}}, \tag{7}$$

where $v_{ij} \in \mathcal{C}^1_p$, ν_i is the number of monomials X_{ij} in the i-th equation, α is the dimension of the codomain ($\Re^\alpha$) of the map $\mathbf{\Psi}'$, and the exponents $p_{ij\alpha}$ are real numbers. This means that all functions in the class $\bar{\mathcal{Q}}^n_{n+p}$ are quadratizable through the intermediate representation (7). This is the reason why hereafter we focus on systems of the type (7), namely *formally polynomial* systems. Thus, let $\mathbf{\Psi}$ be the identity on $\mathcal{D}_f$, and let $\mathbf{\Phi} = \mathbf{\Phi}'$ be the quadratizing map, defined on $\mathcal{D}'_f$, for the system:

$$\dot{x}_i = \sum_{j=1}^{\nu_i} v_{ij}(u) X_{ij}(x); \quad X_{ij}(x) = x_1^{p_{ij1}} \cdots x_n^{p_{ijn}}, \tag{8}$$

where all symbols are the same as in (7) but x replaces z. Then, denoting $m = \sum_{i=1}^n \nu_i$ (the number of monomials) in (8), the basic theorem on EQ [6], [7] states that a quadratization for (8) exists, such that:

i) $\mathcal{D}'_f$ is *dense* in $\mathcal{D}_f$;

ii) $\mu = n + m$, and $\mathbf{\Phi}$ is the aggregate vector $\mathbf{\Phi}(x) = (x, \Phi(x))$, with $\Phi = (\Phi_{ij}, i = 1, \ldots, n, j = 1, \ldots, \nu_i)$ where:

$$\Phi_{ij}(x) = x_i^{-1} X_{ij}(x), \tag{9}$$

iii) denoting, for $v, w \in \Re^\alpha$, $v \circ w = (v_1 w_1, \ldots, v_\alpha, w_\alpha)$, the map B in (6) is given by

$$B(\zeta, \zeta, u) = \zeta \circ V\zeta, \tag{10}$$

where, by *ii)*, $\zeta = (x, Z)$, $Z = \Phi(x)$, and the square matrix $V \in \Re^{\mu \times \mu}$, said *the frame* (of the quadratization), is given by:

$$V = \begin{bmatrix} 0 & d \\ 0 & D \end{bmatrix} \tag{11}$$

with $d \in \Re^{n \times m}$, $D \in \Re^{m \times m}$ given by

$$d = \begin{bmatrix} v_{11} \ldots v_{1\nu_1} & 0 & \ldots & 0 \\ & \ddots & & \\ 0 & \ldots & 0 & v_{n1} \ldots v_{n\nu_n} \end{bmatrix} \tag{12}$$

$$D = \begin{bmatrix} \pi_{111} v_{11} \ldots \pi_{111} v_{1\nu_1} & \ldots & \pi_{11n} v_{n1} \ldots \pi_{11n} v_{n\nu_n} \\ & \vdots & \\ \pi_{n\nu_n 1} v_{11} \ldots \pi_{n\nu_n 1} v_{1\nu_1} & \ldots & \pi_{n\nu_n n} v_{n1} \ldots \pi_{n\nu_n n} v_{n\nu_n} \end{bmatrix} \tag{13}$$

where $\pi_{ijk} = p_{ijk} - \delta_{ik}$ (δ is the Kronecker symbol: $\delta_{ik} = 1$ if $i = k$, and 0 otherwise). The quadratization

$$\dot{\zeta} = \zeta \circ V\zeta, \tag{14}$$

is said *canonical quadratization* of system (8).

III. The proposed new algorithm

A. The general idea of the algorithm

Let us gather all the coefficients $v_{ij}(u)$ into a vector $v = v(u) \in \Re^m$, then, provided that $\mathcal{D}_f = \mathcal{D}'_f$, i.e. a quadratization exists on the whole domain of f, all the curves (in $\Re^{\mu+m}$) $t \mapsto (\zeta(t), v(t))$, $(v(t) = v(u(t)))$, solutions of (8) passing each (for a fixed value of v, say $\bar{v}$) through a fixed point $(\bar{x}, \mathbf{\Phi}(\bar{x}), \bar{v})$ lie entirely in the submanifold $\mathbf{\Phi}(\mathcal{D}_f) \times \Re^m$, and the original solution $t \mapsto (x(t), v(t))$ is entirely recovered directly by projecting ζ on the hyperplane $\Re^{n+m}$ of the first $\mu = n + m$ coordinates ((x, v)). If we solve the system of equations $\zeta \circ V\zeta$ we obtain a set of hyphersurfaces, i.e. the algebraic variety $\mathcal{V}(\zeta \circ V\zeta)$, consisting of all equilibrium points of the canonical quadratic system $\dot{\zeta} = \zeta \circ V\zeta$, and the corresponding (necessarily constant in time) v values. The equilibrium points of the original system are precisely the subset of $\mathcal{V}(\zeta \circ V\zeta)$ that lie in $\mathbf{\Phi}(\mathcal{D}_f) \times \Re^m$, and, hence, we can get all of them by using (9), and solving in $\zeta = (x, Z, v)$, the joint system of equations:

$$\zeta \circ V(v)\zeta = 0, \tag{15}$$

$$Z - \Phi(x) = 0, \tag{16}$$

where we have made explicit the dependence of V by v (which is linear, as we can see in (12), and (13)). Provided that the solution-set for (15), (16) is not empty, it is a union of hyphersurfaces in $\Re^{\mu+m}$ that is described by the union of the graphs of maps of the type $v \mapsto \Psi_k(v)$, for $k \in K$ in some finite set K, and v in some subset of $\Re^m$. Thus, for any admissible $\bar{v}$, the set $\{\Psi_k(\bar{v}) : k \in K\}$ is the set of

equilibrium points of the original system corresponding to $\bar{v}$. Finally, in order to find the equilibrium inputs we have to solve $v(u) - \bar{v} = 0$ with respect to u, however note that finding the u's from the v's is a separated problem with respect to the main (and more difficult) problem of finding the maps $v \to \Psi_k(v)$, and in fact, for $u \mapsto v(u)$ analytic, it amounts to find, in the (v,u)-space $\Re^{m+p}$, the hyphersurfaces given by $\det(\partial v/\partial u) = 0$: off these surfaces the theorem of the inverse function can be applied to obtain the equilibrium input $\bar{u}$ from $\bar{v}$, whereas, on these surfaces, we directly can read the values of $\bar{u}$ (that may be more than one) from $\bar{v}$. If $p_{ijk} \in \mathbb{N}$ then (8) is a polynomial system. With the further hypothesis that

$$p_{iji} \geq 1, \quad \forall i,j, \tag{17}$$

looking at the structure of the function Φ given in (9), we realize that the subsystem of equations (16) is polynomial as well, in the indeterminates (x, Z, v). In the next section we will see that hypothesis (17) is not really necessary, and in fact we will devise a method for equilibria calculation which is effective for any polynomial systems.

B. The adjoint system

The algorithm we are going to present for calculating the equilibrium points (hereafter EP) of a general polynomial system of the type (8), with $p_{ijk} \in \mathbb{N}$, is based on the following observation. For any nonlinear system $\dot{x} = f(x,u)$, let us consider the system $\dot{x}_i = f_i(x,u) \circ x$, that we name *adjoined system*. Notice that, for any $\bar{x} \in \mathcal{D}_f$, these systems give two different solutions passing through $\bar{x}$, even when the same input is applied, nonetheless, if $\bar{x}$ is an EP and $\bar{u}$ is (a/the) corresponding input for the first system, i.e. $f(\bar{x},\bar{u}) = 0$, then $f(\bar{x},\bar{u}) \circ \bar{x} = 0$, thus the pair $(\bar{x},\bar{u})$ is an EP for the adjoined system as well. In other words, the EP-set of the adjoined system include the one of the original system. The adjoined equilibrium points that are not EP of the original system are coordinate hyperplanes in $\Re^{n+p}$ of the type $x_i = 0$, for $i \in \aleph_0$, where $\aleph_0 \subset \aleph = \{1,\ldots,n\}$. To see this, let $(\bar{x},\bar{u}) \in \Re^{n+p}$ be not an EP for the original system, then define $\aleph_0 \subset \aleph$ (possibly empty), such that $f_i(\bar{x},\bar{u}) \neq 0$, then there are two cases: *(i)* $\bar{x}_i = 0$ for $i \in \aleph_0$, and *(ii)* there is a $j \in \aleph_0$ such that $\bar{x}_j \neq 0$. In the case *(i)* $(\bar{x},\bar{u})$ is a point of the coordinate hyperplane

$$H_{\aleph_0} = \{(x,u) \in \Re^{n+p} : x_i = 0, \quad i \in \aleph_0\},$$

as well as an EP for the adjoined system. In the case *(ii)* $(\bar{x},\bar{u})$ is neither a point of $H_{\aleph_0}$ nor an EP for the adjoined system. With that being said, it is clear that, in order to search for the EP of the original system, we can search for the EP of the adjoined system instead, and then, for any $\aleph_0 \subset \aleph$, discard the points in $U_{\aleph_0}$ that are not EP of the original system. We will see that, for polynomial systems, this can be performed quite easily.

C. The general scheme of the algorithm

Let us consider a polynomial system of the type (8) with $p_{ijk} \in \mathbb{N}$. According to §III-A, the calculation of the equilibrium inputs u is performed after the calculation of the equilibrium values of v, thus, hereafter, we consider the v's as 'inputs', and p as the number of v_{ij} in (8). Note that, for polynomial systems, the adjoined system can be simplified as follows, we set: $\dot{x}_i = f_i(x,v)x_i$ for $i \in \mathcal{A} \subset \aleph_0$, and leave the original equation $\dot{x}_i = f(x,v)$, for $i \in \aleph_0 \setminus \mathcal{A}$, where

$$\mathcal{A} = \{i \in \aleph : \quad p_{iji} = 0, \quad \text{for some} \quad j = 1,\ldots,\nu_i\}$$

that is to say, the set of i that index an equation where at least one monomial X_{ij}, for some $j = 1,\ldots,\nu_i$, does not include x_i (i.e. include a term x_i^0). Notice that all points in $H_{\mathcal{A}_0}$, with $\mathcal{A}_0 = \aleph \setminus \mathcal{A}$, are EP of the original system, therefore we have to find just the EP in $\Re^{n+p} \setminus H_{\mathcal{A}_0}$. Accounting of that, we modify the general scheme of the algorithm as follows: search for the EP of the adjoined system, then for any $\aleph_0 \subset \mathcal{A}$ (which replaces: $\aleph_0 \subset \aleph$) discard the points in $H_{\aleph_0}$ that are not EP of the original system. The following recursive algorithm consists of two macro-steps, and performs all tasks described above:

Macro-step 1. Set $S = \Re^{n+p}$, calculate $\mathcal{A}$, and include all points of $H_{\mathcal{A}_0}$ in the set of EP of the original system, then, build up the adjoined system (if $\mathcal{A} = \emptyset$ the adjoined system is equal to the original system). Calculate all EP of the adjoined system in $S \setminus H$, $H = \bigcup_{\aleph_0 \subset \aleph} H_{\aleph_0}$ i.e. off all hyperplanes of S where $x_i = 0$ for at least one $i \in \aleph$. All of such EP are in fact EP of the original system as well.

Macro-step 2. In the original system set $x_i = 0 \ \forall i \in \mathcal{A}_0$, rename the indeterminates $\{x_i : i \notin \mathcal{A}_0\}$, as $x_1,\ldots,x_{n'}$. If $n' = n$ the algorithm terminates, otherwise (i.e. $n' < n$), rename n' as n, and repeat macro-step 1.

In macro-step 2, the original system is replaced by a simpler one, where all monomials that include x_i^h with $h > 0$ and $i \in \mathcal{A}_0$ vanish (and, hence, possibly a subset of equations vanishes). This new original system is then solved in the subspace $H_{\mathcal{A}_0}$, identified with a *new* $\Re^{n+p}$, where now $x_1,\ldots,x_n$ are the old variables $\{x_i : i \notin \mathcal{A}_0\}$, renumerated.

D. Details of Macro-Step 1

Macro-Step 1 calculates all EP of the adjoined system in $S \setminus H$, that is to say all EP of the type $(\bar{x},\bar{v})$ with $\bar{x}_i \neq 0$ $\forall i = 1,\ldots,n$. This is performed by solving, with respect to (ζ, v), with $\zeta = (x,Z)$, the polynomial system of equations (15), (16), which, on account of the structure of V given in (11), can be rewritten:

$$x \circ d(v)Z = 0 \tag{18}$$

$$Z \circ D(v)Z = 0, \tag{19}$$

$$Z - \Phi(x) = 0. \tag{20}$$

Since $x_i \neq 0$ for $i = 1,\ldots,n$, eq. (18) is equivalent to solve the equation $d(v)Z = 0$. Moreover, $Z_{ij} = 0$ entails $\Phi_{ij}(x) = x_1^{p_{ij1}} \cdots x_n^{p_{ijn}} = 0$, thus, either it is verified with some $x_i = 0$, or it is not verified (if $p_{ijk} = 0$ for all $k =$

$1, \ldots, n$, i.e. the monomial is constant). Therefore, $Z_{ij} \neq 0$ for all i, j, which makes eq. (19) as well equivalent to the equation $D(v)Z = 0$, then (18)–(20) entails $d(v)Z = D(v)Z = Z - \Phi(x) = 0$, which is an hyphersurface in $\Re^{n+m+p}$, i.e. the algebraic variety $V(\mathcal{I})$, where

$$\mathcal{I} = (d(v)Z, D(v)Z, Z - \Phi(x)), \tag{21}$$

Now, let us set the following ordering, that we name the *canonical ordering*, in the set of indeterminates (x, Z, v): $y_k > y_s$ if $k < s$; $x_i > x_j$ if $i < j$; $v_k < x_i < Z_{hl}$, $\forall k = 1, \ldots, p$, $i = 1, \ldots, n$, $h = 1, \ldots, n$, $l = 1, \ldots, \nu_h$; $Z_{hl} > Z_{h'l'}$ if $h < h'$, and, $Z_{hl} > Z_{hl'}$ if $l < l'$. We can prove the following theorem.

Theorem 1 *The generators of the ideal $\mathcal{I}$ in (21) form a Groebner basis for $\mathcal{I}$ with respect to the canonical ordering*

(Proof omitted).

IV. Some remarks and further developments

A. Solving systems of polynomial equations

The algorithm described in §III can be used for solving any system of n polynomial equations in n indeterminates $x = (x_1, \ldots, x_n)$. Indeed, by gathering the polynomials in a vector we can write the system of equations as $f(x) = 0$, finding the solution of which, is equivalent to finding the EPs of the autonomous dynamic system $\dot{x} = f(x)$. Suppose, to exclude trivial cases, that $(f) \neq \Re[x]$, and all polynomials in f are irreducible, i.e. we cannot divide any of them by the others with a zero remainder (in algebraic terms: they form a minimal generator for the ideal (f)), thus $(f) \neq \{0\}$. Then, the associated algebraic variety $\mathcal{V}(f)$ in $\Re^n$, is a non-empty, finite set of points in $\Re^n$ (cf. discussions in §I, and §III-A). To apply our algorithm we build up an adjoint system, following the guidelines of §III-B, and apply exact quadratization to the adjoint system in order to obtain the augmented system of equations associated to the ideal $\mathcal{I}$ in (21), where $v \in \Re^p$ is a vector of known constants. The latter consists of $n + 2m$ equations in the $n + m$ indeterminates $\zeta = (x, Z)$, out of which $n + m$ are *linear* equations in the m indeterminates Z. Clearly, denoted q the number of columns linearly independent of V (note, looking at the structure of V given in (11), that $\mathrm{maxrank}(V) = m$, thus $q \leq m$), the solution Z describes an hypherplane in $\Re^m$ of dimension $m - q$, which is the kernel of the matrix $[d^T : V^T]^T \in \Re^{(n+m)\times m}$. We can find this kernel through an ordinary linear algebra algorithm, which amounts to express q out of the Z-variables as linear functions of the other $m-q$ variables (let us name the latter set Z'). By eliminatng these q linear maps we finally get a polynomial system of m polynomial equations in the $n + m - q$ variables x, Z'. By Theorem 1, all generators of the ideal $\mathcal{I}$ form a Groebner basis for $\mathcal{I}$, thus we can apply the substitution theory directly in order to find the solution. Geometrically, the solution is the intersection of the hyphersurface in $\Re^{n+m}$ defined by $Z - \Phi(x)$, with the hypherplane, in $\Re^{n+m}$, obtained by extending the kernel in $\Re^m$ as a constant with respect to the other n variables x (and in fact the equation $V\zeta = 0$ shows that Z is constant with respect to x).

B. The computational advantage of our algorithm

The computational complexity of the Buchberger's algorithm is very difficult to determine, nonetheless it has been proven that it is bounded by $O(g^{2^n})$, where n is the number of indeterminates, and g is the maximum degree of the system of equations, i.e. the degree of the monomials having maximum degree out of all the monomials of any polynomial in the system. As we have seen, our algorithm allows to avoid the application of the Buchberger's algorithm, because it moves the problem to solving another system of equations, i.e. (18)–(20), whose polynomials constitute always a Groebner basis (with respect to the canonical ordering) of the ideal they generate. The new system is obtained by appling exact quadratization to a suitable dynamic system (the adjoint, system) directly obtained from the original system, and in fact we can say that, essentially, our algorithm consists in replacing the Buchberger's algorithm, always present in the classic method for solving polynomial equations, with the exact quadratization. However, notice that applying exact quadratization is just the underlying idea upon which the method is based: as a matter of fact, from a computational point of view, exact quadratization amounts to nothing else than building up the two matrices $d(v)$ and $D(v)$ defined in (12), and (13). As for $d(v)$, this matrix is built up with a negligible computational effort, since its entries are just coefficients of the original system polynomials. Therefore, all the computational burden of the quadratization consists in building up the sole matrix $D(v)$, which amounts, looking at the structure of $D(v)$ given in (13), to perform m^2 multiplications, where m is the total number of monomials in the original system. Overall, the comparison is between the computational complexity of the quadratization, which is upperbounded by $O(m^2)$, and the upperbound $O(g^{2^n})$ of the Buchberger's algorithm, thus, even if $m \geq n$, polynomial vs exponential complexity, which definitely shows the computational advantage of the method here presented.

C. Future perspectives: regulators design via exact quadratization

We conclude the paper by pointing out an approach, which is suggested by the EPs calculation method we have here described, for nonlinear control problems involving quadratizable systems. Suppose that, for a dynamic nonlinear, quadratizable, system of the type (1), we have found the set of all EPs, and the result is the finite set of $\mu \in \mathbb{N}$ points in $\Re^{n+p}$: $\mathcal{E} = \{(\bar{x}^1, \bar{u}^1), \ldots, (\bar{x}^\mu, \bar{u}^\mu)\}$. We can next ask to find, for each $(\bar{x}^i, \bar{u}^i) \in \mathcal{E}$ the family of sets $\mathcal{U}_i = \{U_i^K \subset \Re^{n+p} : K \in \Re^-\}$ defined as follows: for any $K \in \Re^-$, $U_i^K \subset \Re^{n+p}$ is the largest set such that $(\bar{x}^i, \bar{u}^i) \in U_i^K$, and for any $(x', u') \in U_i^K$ there is $\lambda \in \Re^+$, $\lambda \geq |\|\bar{x}^i\|^2 + \|\bar{u}^i\|^2 - (\|\bar{x}'\|^2 + \|\bar{u}'\|^2)|$, and a map:

$\Re \ni [0, +\infty) \mapsto (x(t), u(t)) \in \Re^{n+p}$, solution of (1) such that $x(0) = x'$, $u(0) = u'$, and

$$|\|x(t)\|^2 + \|u(t)\|^2 - (\|\bar{x}^i\|^2 + \|\bar{u}^i\|^2)| \leq \lambda \exp(Kt).$$

Since $\mathcal{E}$ is the set of EQs for system (1), $\mathcal{U}_i \neq \emptyset$, as it includes the set $\{(\bar{x}^i, \bar{u}^i)\}$ (indeed, under the constant input $u(t) \equiv \bar{u}^i$ we have $(x(t), u(t)) \equiv (\bar{x}^i, \bar{u}^i)$). We say that $(\bar{x}^i, \bar{u}^i)$ is a SES (Strongly Exponentially Stable) EP for system (1) if $\mathcal{U}_i$ includes a neighbourhood[3] of $(\bar{x}^i, \bar{u}^i)$.

Clearly, finding which points of $\mathcal{E}$ are SES, and the associated families $\mathcal{U}_i$, is an instance of a *regulation problem* (to be understood in both senses of open- or closed-loop[4]) for the nonlinear control system (1). Now, let us suppose that system (1) is the formally polynomial system (8) (the same argument can be extended to any quadratizable system either), as argued in §II-B, and §III-A, all trajectories in $\Re^{n+m}$ (and *a fortiori* all the EPs) of system (8) are the projections on the hypherplane, $\mathcal{H} = \Re^{n+m}$, of the variables (x, v), of the trajectories (and of the EPs) of the canonical quadratization (14) that lie in the manifold $\mathcal{M} = \mathbf{\Phi}(\mathcal{D}_\mathbf{f}) \times \Re^m$, the latter being a sub-manifold (of dimension $n + m$) of $\Re^{n+2m}$. In other words, the original system is *equivalent* to a quadratic system *evolving on the manifold* $\mathcal{M}$: even though system (14) has trajectories in all of $\Re^{n+2m}$, there are trajectories all included in $\mathcal{M}$, and in fact every trajectory that intersects $\mathcal{M}$ *lies entirely in* $\mathcal{M}$. As a matter of fact, any trajectory of the quadratization (14) that lies on $\mathcal{M}$, once projected on $\mathcal{H}$, yields a trajectory of the original system, and vice-versa: any original trajectory is obtained by projection on $\mathcal{H}$ of a trajectory of the quadratization lying in $\mathcal{M}$.

This means that, if we are able to solve the regulation problem for the quadratization (14), we get the solution as well of the regulation problem for the original system, because the sets U_i^K for the original problem are nothing else than the projection on $\mathcal{H}$ of the intersections with the manifold $\mathcal{M}$ of the U_i^K obtained solving the regulation problem for the quadratization (14). In particular, any SES EP, P, of the quadratization lying on $\mathcal{M}$, projected on $\mathcal{H}$, is a SES EP for the original system: indeed, naming P' the projection on $\mathcal{H}$ of P, the projection on $\mathcal{H}$ of a neighbourhood of P, namely $U_i^K \cap \mathcal{M}$, is a neighbourhood of P' in $\mathcal{H}$, thus P' is a SES EP for the original system. In conlcusion, regulation problems for nonlinear systems of a very general type (quadratizable systems form indeed a *huge* class) can be all studied as regulation problems for a system of the type (14), i.e. a dynamic system homogeneous and quadratic (with respect to ζ), and cubic with respect to the aggregated variable (ζ, v).

[3]We comply with the classic notion according to which a neighbourhood of a point P in a topologic space T is a subset of T that includes an open set that includes P

[4]In closed-loop amounts to say that a subset of equations in the system (1), namely $\dot{x}_i = f_i(x, u)$ for $i \in \mathcal{S} \subset \{1, \ldots, n\}$, represent a (fixed) *dynamic feedback controller*, i.e. x_i for $i \notin \mathcal{S}$, represent the state-variables of an open-loop system, and $x_i \in \mathcal{S}$ are actually inputs for such open-loop system, whereas u is an exogenous input.

V. Conclusion

Exact Quadratization applied to a polynomial dynamic system in $\Re^\mu$ does not lead, in general, to a quadratization on the whole system domain, because the quadratizing map (9) could be undefined on some coordinate hypherplane of $\Re^\mu$. The new method for EP calculation we have here proposed uses exact quadratization relying on the observation that it is always possible to build up an *adjoint system* (defined in §III-B) whose EPs are the same as the original system, and can be quadratized on the whole space either. The algorithm, described in §III-C, and §III-D, consists of two macro-steps: in the first one, the EPs are calculated off all coordinate hyperplanes, whereas the second macro-step first projects the system on certain maximal hypherplanes and call recursively the first macro-step in order to repeat the calculation on smaller and smaller subspaces. The system of equations (15), (16), whose solution set is the set of EPs of a quadratization of the original system, is polynomial if the original system is polynomial, and in this case, as proven in Theorem 1, it is always a Groebner basis of the ideal it generates in $\Re[x]$. This fact is the main result of the paper, because allows to skip the computationally heavy Buchberger's algorithm in the subsequent solution procedure. In §IV-B we have given a (coarse) quantification of the computational advantage of our algorithm. Finally, in §IV-C we have pointed out a possible development of the present research in the field of control systems.

References

[1] A. Gathmann, *Algebraic Geometry*, Class Notes TU Kaiserslautern, 2014.

[2] D.S. Dummit, and R.M. Foote, *Abstract Algebra*, John Wiley & Sons, 3nd edition, 2004.

[3] L. Menini, C. Possieri, and A. Tornambè, *Algebraic Geometry for Robotics and Control Theory*, World Scientific, 2021.

[4] G. Conte, C.H. Moog, and A.M. Perdon, *Algebraic Methods for Nonlinear Control Systems*, Communication and Control Engineering Series, Springer Verlag, 2nd edition, 2006.

[5] M. Fliess, J. Lévine, Ph. Martin, P. Rouchon, Sur les systèmes non linéaires différentiellement plats, *C. R. Acad. Sci.* Paris, Ser. I 315, 1992, pp 619–624.

[6] F. Carravetta, Global exact quadratization of continuous-time nonlinear control systems. *SIAM J. Cont. & Opt.*, vol. 53, 2015 ,pp 235–261.

[7] F. Carravetta, On the Solution Calculation of Nonlinear Ordinary Differential Equations via Exact Quadratization. *J. Differ. Equ.*, 269, 2020, pp. 11328-11365..

[8] F. Carravetta, "A brief and some further insight on the exact quadratization of nonlinear control systems", *in Proc. of 55th IEEE Conference on Decision and Control*, Las Vegas, USA, Dec. 12–14, 2016

[9] A. Borri, F. Carravetta, and P. Palumbo, *Quadratized Taylor series methods for ODE numerical integration*, Applied Mathematics and Computation 458, 2023, 128237

[10] F. Carravetta, M. A. Sarafrazi, Z. Bartosiewicz, and Ü. Kotta, "Testing Generic Strong Accessibility of Nonlinear Control Systems via Polynomial (Quadratic) Immersion" *in Proc. of 61th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 6–9, 2022

[11] B.I. Dundas, *A Short Course in Differential Topology*, Cambridge Mathematical Textbooks, Cambridge University Press, 2018.

[12] W.M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Pure and Applied Mathematics, Academic Press, 1975

[13] S. Helgason, *Differential Geometry, Lie Groups, and Symmetric Spaces*, Graduate Studies in Mathematics, American Mathematical Society, 1978.