Dissipative quadratizations of polynomial ODE systems *

Yubo Cai[1(✉)] and Gleb Pogudin[2]

[1] École Polytechnique, Institute Polytechnique de Paris, Palaiseau, France
`yubo.cai@polytechnique.edu`
[2] LIX, CNRS, École Polytechnique, Institute Polytechnique de Paris, Palaiseau, France `gleb.pogudin@polytechnique.edu`

Abstract. Quadratization refers to a transformation of an arbitrary system of polynomial ordinary differential equations to a system with at most quadratic right-hand side. Such a transformation unveils new variables and model structures that facilitate model analysis, simulation, and control and offer a convenient parameterization for data-driven approaches. Quadratization techniques have found applications in diverse fields, including systems theory, fluid mechanics, chemical reaction modeling, and mathematical analysis.

In this study, we focus on quadratizations that preserve the stability properties of the original model, specifically dissipativity at given equilibria. This preservation is desirable in many applications of quadratization including reachability analysis and synthetic biology. We establish the existence of dissipativity-preserving quadratizations, develop an algorithm for their computation, and demonstrate it in several case studies.

Keywords: differential equations · quadratization · stability · variable transformation

1 Introduction

Systems of ordinary differential equations (ODEs) are the standard choice when it comes to modeling processes happening in continuous time, for example, in the sciences and engineering. For a given dynamical process, one can derive different ODE models, in particular, by choosing different sets of variables. It has been observed in a variety of areas and contexts that these choices may have a significant impact on the utility and relevance of the resulting model, and a number of different types of variable transformations have been studied.

In this paper, we will study one such transformation, *quadratization*, which aims at transforming an ODE system to a system where the right-hand side

consists of polynomials of degree at most two. Let us illustrate this transformation on a toy example: we start with a scalar ODE $x' = x^3$ in a single variable $x = x(t)$ with cubic right-hand side. If we now augment the state space with an additional coordinate $y = x^2$, we can write the original equation as $x' = xy$ with quadratic right-hand side, and we can do the same for y':

$$y' = 2xx' = 2x^4 = 2y^2.$$

So, the transformation in this case is the following:

$$x' = x^3 \quad \rightarrow \quad \begin{cases} x' = xy, \\ y' = 2y^2. \end{cases}$$

It turns out that every polynomial ODE system can be similarly lifted to an at most quadratic one: this fact has been established at least 100 years ago [2,27] and has been rediscovered several times since then [8,10,11,17,24]. In the recent years quadratization has been used in a number of application areas including model order reduction [5,6,17,25,26], synthetic biology [13,20,21], numerical integration [16,18,19], and reachability analysis [14]. While it has been shown in [21] that the problem of finding the minimal number of extra variables necessary for quadratization is NP-hard, at least two practically useful software packages have been developed for performing quadratization: BioCham [21] and QBee [7].

In the majority of the applications mentioned above, the constructed quadratic ODE model is further used in the context of *numerical* simulations. It is, therefore, a natural question whether one can not only guarantee that the transformed model is at most quadratic, but also that it preserves some desirable dynamical/numerical properties of the original ODE system. To the best of our knowledge, this question has not been studied systematically, and in this paper, we initiate this line of research by studying *dissipativity-preserving quadratizations*.

We will say that an ODE system is *dissipative* at an equilibrium point if the real parts of the eigenvalues of the linearization of the system around this point are negative. In particular, dissipativity implies that the system is *asymptotically stable* at this point [23, Theorem 8.2.2]. The main contribution of the paper is two-fold. First, we prove that, for every polynomial system dissipative at several equilibrium points, there exists a quadratization which is also dissipative at all these points. Second, we design and implement an algorithm to search automatically for such quadratization attempting to minimize the dimension. Our algorithm is based on a combinatorial condition on the new variables which is sufficient to guarantee that the resulting quadratic model can be made dissipative as well. This combinatorial condition can be viewed as a generalization and formalization of the artificial stabilization used in [26, Section 4.1]. We implemented the new algorithm and we illustrate it in several case studies including an application for reachability analysis (in combination with the algorithm from [14]). Our implementation together with the examples from this paper is available at [1].

The rest of the paper is organized as follows. In Section 2, we introduce the main notions, quadratization, and dissipativity, and show that quadratization

performed straightforwardly may not preserve dissipativity (and, thus, render the model into a numerically unstable one). Section 3 contains the statement and the proof of the main theoretical result of the paper (Theorem 1) that there always exists a dissipativity-preserving quadratization for any collection of dissipative equilibria. Based on the ideas from the proof, we give an algorithm (Algorithm 2) for constructing such a quadratization in Section 4. We showcase our implementation of this algorithm on several case studies in Section 5. Concluding remarks are contained in Section 6.

2   Preliminaries

Throughout this section, we will consider a polynomial ODE system, that is, a system of differential equations

$$\mathbf{x}' = \mathbf{p}(\mathbf{x}), \tag{1}$$

where $\mathbf{x} = \mathbf{x}(t) = (x_1(t), \ldots, x_n(t))$ is a vector of unknown functions and $\mathbf{p} = (p_1, \ldots, p_n)$ is a vector of n-variate polynomials $p_1, \ldots, p_n \in \mathbb{R}[\mathbf{x}]$.

Definition 1 (Quadratization). *For a system* (1), *quadratization is a pair consisting of*

- *a list of new variables*

$$y_1 = g_1(\mathbf{x}), \ldots, y_m = g_m(\mathbf{x})$$

- *and two lists*

$$\mathbf{q}_1(\mathbf{x}, \mathbf{y}) = (q_{1,1}(\mathbf{x}), \ldots, q_{1,n}(\mathbf{y})) \quad and \quad \mathbf{q}_2(\mathbf{x}, \mathbf{y}) = (q_{2,1}(\mathbf{x}, \mathbf{y}), \ldots, q_{2,m}(\mathbf{x}, \mathbf{y}))$$

of $m + n$-variate polynomials in $\mathbf{x}$ and $\mathbf{y} = (y_1, \ldots, y_m)$

such that the degree of each of of $\mathbf{q}_1$ and $\mathbf{q}_2$ is at most two and

$$\mathbf{x}' = \mathbf{q}_1(\mathbf{x}, \mathbf{y}) \quad and \quad \mathbf{y}' = \mathbf{q}_2(\mathbf{x}, \mathbf{y}). \tag{2}$$

If all the polynomials $g_1, \ldots, g_m$ are monomials, the quadratization is called monomial quadratization.

Note that unlike, for example, [7, Definition 1], by *quadratization* we mean not just the set of new variables but also the quadratic ODE system (2). The reason for this is that, for a fixed set of new variables, there may be many different systems of the shape (see Example 1) prescribed by (2) exhibiting different numerical behaviors (see Example 2).

Example 1 (Quadratization). Consider the following scalar ODE

$$x' = -x + x^3.$$

Here we have $n = 1$ and $p_1(x) = -x + x^3$. Consider $y = g_1(x) = x^2$. Then we can write

$$x' = -x + x^3 = -x + xy,$$
$$y' = 2xx' = -2x^2 + 2x^4 = -2y + 2y^2.$$

Therefore, one possible quadratization is given by

$$g_1(x) = x^2, \quad q_{1,1}(x, y) = -x + xy, \quad q_{2,1}(x, y) = -2y + 2y^2.$$

As we have mentioned above, there may be different $\mathbf{q}$'s corresponding to the same $\mathbf{g}$. In this example, we could take, for example, $q_{2,1} = -2y + 2y^2 + 2(y - x^2) = -2x^2 + 2y^2$ or $q_{2,1} = y - 3x^2 + 2y^2$. As we will see in Example 2, such choices may have a dramatic impact on the numerical properties of the resulting ODE system.

Definition 2 (Equilibrium). *For a polynomial ODE system* (1), *a point* $\mathbf{x}^* \in \mathbb{R}^n$ *is called an* equilibrium *if* $\mathbf{p}(\mathbf{x}^*) = 0$.

Definition 3 (Dissipativity). *An ODE system* (1) *is called* dissipative *at an equilibrium point* $\mathbf{x}^*$ *if all the eigenvalues of the Jacobian* $J(\mathbf{p})|_{\mathbf{x}=\mathbf{x}^*}$ *of* $\mathbf{p}$ *and* $\mathbf{x}^*$ *have negative real part.*

It is known that a system which is dissipative at an equilibrium point $\mathbf{x}^*$ *is* asymptotically stable *at* $\mathbf{x}^*$ *[23, Theorem 8.2.2], that is, any trajectory starting in a small enough neighborhood of* $\mathbf{x}^*$ *will converge to* $\mathbf{x}^*$ *exponentially fast.*

Note that if $\mathbf{x}^* = 0$, *then the Jacobian at this point is simply the matrix of the linear part of* $\mathbf{p}(\mathbf{x})$.

Assume that $\mathbf{x}^* \in \mathbb{R}^n$ is an equilibrium of $\mathbf{x}' = \mathbf{p}(\mathbf{x})$, and consider a quadratization of this system as in Definition 1. Then a direct computation shows that $(\mathbf{x}^*, \mathbf{g}(\mathbf{x}^*))$ is an equilibrium point of the resulting quadratic system (2).

Definition 4 (Dissipative quadratization). *Assume that a system* (1) *is dissipative at an equilibrium point* $\mathbf{x}^* \in \mathbb{R}^n$. *Then a quadratization given by* $\mathbf{g}, \mathbf{q}_1$ *and* $\mathbf{q}_2$ *(see Definition 1) is called* dissipative *at* $\mathbf{x}^*$ *if the system*

$$\mathbf{x}' = \mathbf{q}_1(\mathbf{x}, \mathbf{y}), \quad \mathbf{y}' = \mathbf{q}_2(\mathbf{x}, \mathbf{y})$$

is dissipative at a point $(\mathbf{x}^*, \mathbf{g}(\mathbf{x}^*))$.

The following example shows that, even for the same new variables $\mathbf{y} = \mathbf{g}(\mathbf{x})$, different quadratizations may have significantly different stability properties.

Example 2 (Stable and unstable quadratizations). Consider the scalar ODE $x' = -x + x^3$ from Example 1. We have already found a quadratization for it using a new variable $y = g_1(x) = x^2$ with the resulting quadratic system being

$$x' = -x + xy \quad \text{and} \quad y' = -2y + 2y^2. \tag{3}$$

We notice that we can add/subtract $y - x^2$ with any coefficients from the right-hand sides of the system. For example, we can obtain:

$$x' = -x + xy \quad \text{and} \quad y' = -2y + 2y^2 + 12(y - x^2) = 10y - 12x^2 + 2y^2. \quad (4)$$

Both systems above are quadratizations of the original system and, thus, mathematically, for any initial condition (x_0, y_0) satisfying $y_0^2 = x_0^2$, they must follow the same trajectory. However, (3) is stable at $(0,0)$ while (4) is not. By numerically integrating them, we can observe in Figure 1 that in practice (3) reflects the dynamics of the original equation accurately and (4) heavily suffers from numerical instability.
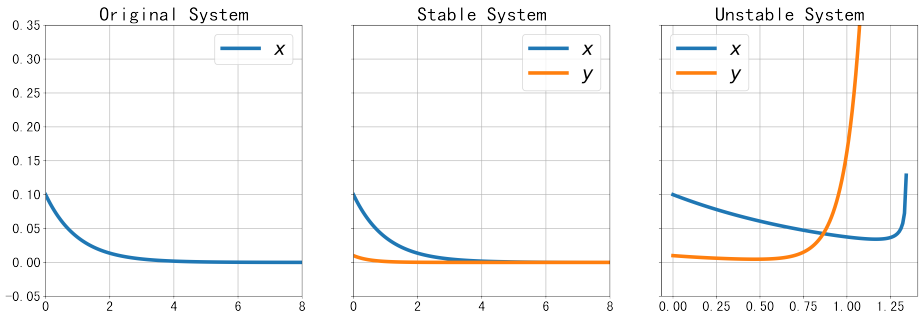

Fig. 1: Plot of the original equation, (3), and (4) with initial condition $\mathcal{X}_0 = [x_0, y_0 = x_0^2] = [0.1, 0.01]$. Numerical method: "LSODA" (uses hybrid Adams/BDF method with automatic stiffness detection) in scipy.integrate.solve_ivp package [22,29].

3   Existence of dissipativity-preserving quadratizations

The main result of this section is the following theorem. Its proof is constructive and is used to design an algorithm in Section 4.

Theorem 1. *For every polynomial ODE system* $\mathbf{x}' = \mathbf{p}(\mathbf{x})$, *there exists a quadratization that is dissipative at all the dissipative equilibria of* $\mathbf{x}' = \mathbf{p}(\mathbf{x})$.

Remark 1. In fact, the key ingredients of the proof, Propositions 1 and 2, imply a stronger statement: for every set of finitely many equilibria, there is a quadratization such that the number of nonnegative eigenvalues of the Jacobian of the quadratic system at these points is the same as for the original system.

The rest of the section will be devoted to proving Theorem 1. The main technical notion will be an *inner-quadratic* set of polynomials.

Definition 5 (Inner-quadratic set). *As finite set* $g_1(\mathbf{x}), \dots, g_m(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ *of nonconstant polynomials in* $\mathbf{x} = (x_1, \dots, x_n)$ *is called* inner-quadratic *if, for every* $1 \leqslant i \leqslant m$, *there exist (not necessarily distinct)* $a, b \in \{x_1, \dots, x_n, g_1, \dots, g_m\}$ *such that* $g_i = ab$.

A quadratization will be called inner-quadratic *if the set of new variables* **g** *is inner-quadratic. We will also always assume that the new variables are sorted by degree, that is,* $\deg g_1 \leqslant \deg g_2 \leqslant \ldots \leqslant \deg g_m$.

The rationale behind the notion of inner-quadratic quadratization is that at most quadratic relations between the new variables give us the flexibility to "tune" the right-hand side of the resulting quadratic system in the same fashion as we added a multiple of $y - x^2$ in Example 2. These additional terms force the trajectory to stay on the image of the map $\mathbf{x} \to (\mathbf{x}, \mathbf{y})$, on which the properties of the original dynamics (such as dissipativity) are preserved. The following definition formalizes this observation.

Definition 6 (Stabilizers). *Consider a polynomial ODE system* $\mathbf{x}' = \mathbf{p}(\mathbf{x})$ *and its inner-quadratic quadratization given by m new variables* $\mathbf{y} = \mathbf{g}(\mathbf{x})$ *and right-hand side* $\mathbf{q}_1(\mathbf{x}, \mathbf{y}), \mathbf{q}_2(\mathbf{x}, \mathbf{y})$ *of the resulting quadratic system as in Definition 1. For every $1 \leqslant i \leqslant m$, by the definition of inner-quadratic set, there exist $a_i, b_i \in \{\mathbf{x}, y_1, \ldots, y_{i-1}\}$ such that the equality $y_i = a_i b_i$ holds if we replace each y_j with $g_j(\mathbf{x})$. We define the i-th stabilizer by $h_i(\mathbf{x}, \mathbf{y}) := y_i - a_i b_i$.*

Since each stabilizer is at most quadratic and vanishes under the substitution $\mathbf{y} = \mathbf{g}(\mathbf{x})$, *adding any stabilizer to any of* $\mathbf{q}_1, \mathbf{q}_2$ *still yields a quadratization of* $\mathbf{x}' = \mathbf{p}(\mathbf{x})$.

Example 3 (Stabilizers). Let us give an example of the stabilizers. Consider a system:

$$x_1' = -3x_1 + x_2^4, \quad x_2' = -2x_2 + x_1^2.$$

By applying **Algorithm 1**, we introduce the following new variables to obtain an inner-quadratic quadratization:

$$y_1 = x_1^2, \quad y_2 = x_2^2, \quad y_3 = x_1 x_2, \quad y_4 = x_2^3 = x_2 y_2.$$

Then the corresponding stabilizers, according to the definition above, will be:

$$h_1(\mathbf{x}, \mathbf{y}) = y_1 - x_1^2, \qquad h_2(\mathbf{x}, \mathbf{y}) = y_2 - x_2^2,$$
$$h_3(\mathbf{x}, \mathbf{y}) = y_3 - x_1 x_2, \qquad h_4(\mathbf{x}, \mathbf{y}) = y_4 - x_2 y_2$$

Theorem 1 follows directly from the following two properties of inner-quadratic quadratizations:

- every polynomial ODE system has an inner-quadratic quadratization (Proposition 1);
- for any inner-quadratic quadratization, one can modify the right-hand sides of the quadratic system (but not the new variables) using the stabilizers in order to obtain a dissipativity-preserving quadratization (Proposition 2).

Proposition 1. *Every polynomial ODE system* $\mathbf{x}' = \mathbf{p}(\mathbf{x})$ *admits an inner-quadratic quadratization. Furthermore, it can be chosen to be a monomial quadratization.*

Proof. We will show that the quadratization which is typically used to prove the existence of a quadratization for every polynomial ODE system (see, e.g. [9, Theorem 1]) is in fact inner-quadratic. For every $1 \leqslant i \leqslant n$, we introduce $d_i = \max\limits_{1 \leqslant j \leqslant n} \deg_{x_i} p_j$. Then it is proven in [9, Theorem 1] that the following set of new variables yields a quadratization of $\mathbf{x}' = \mathbf{p}(\mathbf{x})$:

$$\mathcal{M} = \{x_1^{i_1} \ldots x_n^{i_n} \mid \forall j\colon 0 \leqslant i_j \leqslant d_j, \ \sum i_j > 1\}.$$

Let $g \in \mathcal{M}$. Then there exists $1 \leqslant j \leqslant n$ such that $\deg_{x_j} g > 0$. Then we can write $g = (g/x_j) \cdot x_j$, where g/x_j is either in $\mathcal{M}$ or belongs to $\{x_1, \ldots, x_n\}$. Thus, $\mathcal{M}$ is an inner-quadratic set.

Proposition 2. *Consider a system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$ and its inner-quadratic quadratization defined by new variables $\mathbf{g}(\mathbf{x})$ and the new right-hand side $\mathbf{q}_1, \mathbf{q}_2$ as in Definition 1. Let $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$ be a finite subset of the equilibria of the system. Then there exist vectors of quadratic polynomials $\mathbf{r}_1(\mathbf{x}, \mathbf{y}), \mathbf{r}_2(\mathbf{x}, \mathbf{y})$ such that $\mathbf{g}, \mathbf{r}_1, \mathbf{r}_2$ define a quadratization for which the eigenvalues of the Jacobian at each equilibrium point of the form $(\mathbf{x}_i^*, \mathbf{g}(\mathbf{x}_i^*))$ are the union of the eigenvalues of $J(\mathbf{p})|_{\mathbf{x}=\mathbf{x}_i^*}$ and a set of numbers with negative real part.*

Corollary 1. *Consider a system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$ and its inner-quadratic quadratization defined by new variables $\mathbf{g}(\mathbf{x})$ and the new right-hand side $\mathbf{q}_1, \mathbf{q}_2$ as in Definition 1. Then there exist vectors of quadratic polynomials $\mathbf{r}_1(\mathbf{x}, \mathbf{y}), \mathbf{r}_2(\mathbf{x}, \mathbf{y})$ such that $\mathbf{g}, \mathbf{r}_1, \mathbf{r}_2$ define a quadratization which is dissipative at every dissipative equilibrium of $\mathbf{x}' = \mathbf{p}(\mathbf{x})$.*

Proof (Proof of Corollary 1). Since each dissipative equilibrium of the system is an isolated root of the polynomial system obtained by equating the right-hand side to zero, there are only finitely many of them. So we apply Proposition 2 to this finite set of equilibria and obtain the desired quadratization.

Before proving Proposition 2, we establish a useful linear-algebraic lemma.

Lemma 1. *Let $A \in \mathbb{R}^{n \times n}$ be a square matrix and $B \in \mathbb{R}^{n \times n}$ be an upper triangular matrix with ones on the diagonal. Then there exists $\lambda_0 \in \mathbb{R}$ such that, for every $\lambda > \lambda_0$, the real parts of all the eigenvalues of $A - \lambda B$ are negative.*

Proof. Consider the characteristic polynomial of $A - \lambda B$. It can be written as

$$\det(A - \lambda B - tI) = (-\lambda)^n \det(B - A/\lambda + (t/\lambda I)).$$

We set $T = t/\lambda$ and rewrite the latter determinant as $Q(T, 1/\lambda) := \det((B + T \cdot I) - A/\lambda)$. Since Q is the determinant of a matrix with the entries linear in T and $1/\lambda$, it is a bivariate polynomial in T and $1/\lambda$ of total degree at most n. Furthermore, if we set $\lambda = \infty$ (equivalently, if we set $1/\lambda = 0$), we have $Q(T, 0) = \det(B + T \cdot I)$. Since B is upper-triangular with ones on the diagonal, this determinant is equal to $\det(B + T \cdot I) = (T + 1)^n$, so $Q(T, 1/\lambda)$ can be written as $Q(T, 1/\lambda) = (T + 1)^n + \frac{1}{\lambda} p\left(T + 1, \frac{1}{\lambda}\right)$, where p is a bivariate polynomial of

the total degree at most $n-1$ in $T+1$ and $\frac{1}{\lambda}$. Let C be an upper bound for the absolute value of the coefficients of p. Then, for $\lambda > 1$, we can bound:

$$\left| p\left(T+1, \frac{1}{\lambda}\right) \right| < Cn^2 \max(|T+1|^{n-1}, 1).$$

Let T_0 be any root of $Q(T, 1/\lambda)$. Then we have

$$|T_0 + 1|^n \leqslant \frac{1}{\lambda} Cn^2 \max(|T_0 + 1|^{n-1}, 1).$$

Let us take $\lambda > Cn^2$. Then

$$|T_0 + 1|^n < \max(|T_0 + 1|^{n-1}, 1) \implies |T_0 + 1| < 1.$$

So, in this case, the real part of any root of Q will be negative. Then the same is true for the characteristic polynomial of $A - \lambda B$ because these two polynomials differ by scaling by a positive number λ. Therefore, λ_0 can be taken to be $\max(1, Cn^2)$.

We will also use the following folklore analytic lemma.

Lemma 2. *Let* $\mathbf{x}' = \mathbf{f}(\mathbf{x})$ *a system of polynomial differential equations of dimension n with an equilibrium point $\mathbf{x}^*$. Let $\varphi \colon \mathbb{R}^n \to \mathbb{R}^n$ be an invertile change of coordinates, and let $\mathbf{y}' = \mathbf{g}(\mathbf{y})$ be the image of the system under the coordinate change. Then the matrices $J_{\mathbf{x}}(\mathbf{f})|_{\mathbf{x}=\mathbf{x}^*}$ and $J_{\mathbf{y}}(\mathbf{g})|_{\mathbf{y}=\varphi(\mathbf{x}^*)}$ are conjugate. In particular, they have the same eigenvalues.*

Proof. By the chain rule, we have

$$\mathbf{y}' = (\varphi(\mathbf{x}))' = J_{\mathbf{x}}(\varphi)\mathbf{x}' = J_{\mathbf{x}}(\varphi)|_{\mathbf{x}=\varphi^{-1}(\mathbf{y})} f(\varphi^{-1}(\mathbf{y})) = \mathbf{g}(\mathbf{y}).$$

Then we can write $J_{\mathbf{y}}(\mathbf{g})$ as

$$J_{\mathbf{x}}(\varphi)|_{\mathbf{x}=\varphi^{-1}(\mathbf{y})} J_{\mathbf{x}}(f)|_{\mathbf{x}=\varphi^{-1}(\mathbf{y})} J_{\mathbf{y}}(\varphi^{-1}) + \sum_{i=1}^{n} A_i f_i(\varphi^{-1}(\mathbf{y})),$$

where A_i is the Jacobian of the i-th column of $J_{\mathbf{x}}(\varphi)|_{\mathbf{x}=\varphi^{-1}(\mathbf{y})}$. If we plug $\varphi(\mathbf{x}^*)$ for $\mathbf{y}$, since $\mathbf{f}(\mathbf{x}^*) = 0$, the latter sum will vanish, so we get

$$J_{\mathbf{x}}(\varphi)|_{\mathbf{x}=\mathbf{x}^*} J_{\mathbf{x}}(f)|_{\mathbf{x}=\mathbf{x}^*} J_{\mathbf{y}}(\varphi^{-1})|_{\mathbf{y}=\varphi(\mathbf{x}^*)}.$$

By the chain rule, the matrices $J_{\mathbf{x}}(\varphi)|_{\mathbf{x}=\mathbf{x}^*}$ and $J_{\mathbf{y}}(\varphi^{-1})|_{\mathbf{y}=\varphi(\mathbf{x}^*)}$ are inverses to each other, so the Jacobians are indeed conjugated.

Proof (Proof of Proposition 2). Before starting the proof, we would like to point at Example 6 in our extended version of the paper [3] which illustrates the main steps of the proof.

---

[3] https://arxiv.org/pdf/2311.02508.pdf

We define a map $\varphi\colon \mathbb{R}^{n+m} \to \mathbb{R}^{n+m}$ from a space with coordinates $(\mathbf{x}, \mathbf{y})$ to a space with coordinates $(\mathbf{x}, \mathbf{z})$, where $\mathbf{z} = (z_1, \ldots, z_m)$, by

$$\varphi_i(\mathbf{x}, \mathbf{y}) = x_i \quad \text{for } 1 \leqslant i \leqslant n,$$
$$\varphi_{n+j}(\mathbf{x}, \mathbf{y}) = y_j - g_j(\mathbf{x}) \quad \text{for } 1 \leqslant j \leqslant m.$$

This map is invertible with the inverse given by $(\varphi^{-1})_i(\mathbf{x}, \mathbf{z}) = x_i$ for $1 \leqslant i \leqslant n$ and $(\varphi^{-1})_{n+j}(\mathbf{x}, \mathbf{z}) = z_j + g_j(\mathbf{x})$ for $1 \leqslant j \leqslant m$, so φ is bijective. Note that $\varphi(\mathbf{x}^\circ, \mathbf{g}(\mathbf{x}^\circ)) = (\mathbf{x}^\circ, \mathbf{0})$ for every $\mathbf{x}^\circ \in \mathbb{R}^n$. We apply a change of coordinates defined by φ to the quadratic system $\mathbf{x}' = \mathbf{q}_1(\mathbf{x}, \mathbf{y})$, $\mathbf{y}' = \mathbf{q}_2(\mathbf{x}, \mathbf{y})$ and obtain a (not necessarily quadratic) system of the form:

$$\mathbf{x}' = \widetilde{\mathbf{q}}_1(\mathbf{x}, \mathbf{z}) \quad \text{and} \quad \mathbf{z}' = \widetilde{\mathbf{q}}_2(\mathbf{x}, \mathbf{z}), \tag{5}$$

where $\widetilde{q}_1 = q_1(\mathbf{x}, \mathbf{z} + \mathbf{g}(\mathbf{z}))$ and $\widetilde{q}_2$ can be found using the chain rule as follows:

$$\mathbf{z}' = (\mathbf{y} - \mathbf{g}(\mathbf{x}))' = q_2(\mathbf{x}, \mathbf{z} + \mathbf{g}(\mathbf{x})) - J_{\mathbf{x}}(\mathbf{g})\mathbf{x}' = q_2(\mathbf{x}, \mathbf{z} + \mathbf{g}(\mathbf{x})) - J_{\mathbf{x}}(\mathbf{g})q_1(\mathbf{x}, \mathbf{z} + \mathbf{g}(\mathbf{x})). \tag{6}$$

Since the variety $\{(\mathbf{x}^\circ, \mathbf{g}(\mathbf{x}^\circ)) \mid \mathbf{x}^\circ \in \mathbb{R}^n\}$ was an invariant variety of $\mathbf{x}' = \mathbf{q}_1(\mathbf{x}, \mathbf{y})$, $\mathbf{y}' = \mathbf{q}_2(\mathbf{x}, \mathbf{y})$ by construction, the linear space $\{(\mathbf{x}^\circ, \mathbf{0}) \mid \mathbf{x}^\circ \in \mathbb{R}^n\}$ is invariant for (5) and the restriction of (5) to this space coincides with the original system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$. This implies the following constraints on $\widetilde{\mathbf{q}}_1$ and $\widetilde{\mathbf{q}}_2$:

- $\widetilde{\mathbf{q}}_1(\mathbf{x}, \mathbf{z}) = \mathbf{p}(\mathbf{x}) + \mathcal{O}(\mathbf{z})$, where $\mathcal{O}(\mathbf{z})$ stands for a polynomial with each monomial containing at least one of the $\mathbf{z}$;
- $\widetilde{\mathbf{q}}_2(\mathbf{x}, \mathbf{z}) = \mathcal{O}(\mathbf{z})$.

Due to these constraints, for every $\mathbf{x}^\circ \in \mathbb{R}^n$, the Jacobian of $(\widetilde{\mathbf{q}}_1, \widetilde{\mathbf{q}}_2)$ at $(\mathbf{x}^\circ, \mathbf{0})$ is of the form

$$J_{\mathbf{x}, \mathbf{z}}(\widetilde{\mathbf{q}}_1, \widetilde{\mathbf{q}}_2)|_{\mathbf{x} = \mathbf{x}^\circ, \mathbf{z} = \mathbf{0}} = \begin{pmatrix} J_{\mathbf{x}}(\mathbf{p})|_{\mathbf{x} = \mathbf{x}^\circ} & * \\ 0 & J_{\mathbf{z}}(\widetilde{\mathbf{q}}_2)|_{\mathbf{x} = \mathbf{x}^\circ, \mathbf{z} = \mathbf{0}} \end{pmatrix} \tag{7}$$

Let $h_1(\mathbf{x}, \mathbf{y}), \ldots, h_m(\mathbf{x}, \mathbf{y})$ be the stabilizers of the quadratization (see Definition 6). We take an arbitrary parameter $\lambda \in \mathbb{R}$ and consider $\mathbf{q}_{2,\lambda}(\mathbf{x}, \mathbf{y})$ defined by

$$\mathbf{q}_{2,\lambda}(\mathbf{x}, \mathbf{y}) = \mathbf{q}_2(\mathbf{x}, \mathbf{y}) - \lambda \mathbf{h}(\mathbf{x}, \mathbf{y}). \tag{8}$$

Since the h_i's are stabilizers, $\mathbf{g}, \mathbf{q}_1, \mathbf{q}_{2,\lambda}$ is a quadratization of the original system for any value of λ (see Definition 6). By using $q_{2,\lambda}$ instead of q_2 in (6), we obtain $\widetilde{\mathbf{q}}_{2,\lambda} = \widetilde{\mathbf{q}}_2 - \lambda \mathbf{h}(\mathbf{x}, \mathbf{z} + \mathbf{g}(\mathbf{x}))$. Then, as in (7), we get

$$J_{\mathbf{x}, \mathbf{z}}(\widetilde{\mathbf{q}}_1, \widetilde{\mathbf{q}}_{2,\lambda})|_{\mathbf{x} = \mathbf{x}^\circ, \mathbf{z} = \mathbf{0}} = \begin{pmatrix} J_{\mathbf{x}}(\mathbf{p})|_{\mathbf{x} = \mathbf{x}^\circ} & * \\ 0 & (J_{\mathbf{z}}(\widetilde{\mathbf{q}}_2) - \lambda J_{\mathbf{z}}(\mathbf{h}))|_{\mathbf{x} = \mathbf{x}^\circ, \mathbf{z} = \mathbf{0}} \end{pmatrix} \tag{9}$$

Observe that, since every g_i is of the form z_i plus polynomial in $\mathbf{x}$ and z's with smaller indices, $J_{\mathbf{z}}(\mathbf{h})$ is a lower-triangular matrix with ones on the diagonal.

Having such a convenient expression for the Jacobian, we consider the given equilibria $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$. For any $\mathbf{x}^\circ \in \{\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*\}$, the eigenvalues of the Jacobian (9) are the union of the eigenvalues of $J_{\mathbf{x}}(\mathbf{p})|_{\mathbf{x} = \mathbf{x}^\circ}$ and the eigenvalues of $(J_{\mathbf{z}}(\widetilde{\mathbf{q}}_2) - \lambda J_{\mathbf{z}}(\mathbf{h}))|_{\mathbf{x} = \mathbf{x}^\circ, \mathbf{z} = \mathbf{0}}$. Applying Lemma 1 to ℓ pairs of matrices

$A = J_{\mathbf{z}}(\widetilde{\mathbf{q}}_2)|_{\mathbf{x}=\mathbf{x}_i^*, \mathbf{z}=0}$ and $B = J_{\mathbf{z}}(\mathbf{h})|_{\mathbf{x}=\mathbf{x}_i^*, \mathbf{z}=0}$, we choose λ to be larger than any of the λ_0's provided by the lemma. Then all the eigenvalues of this block will also have negative real parts. By Lemma 2, the same true for the Jacobian of $\mathbf{x}' = \mathbf{q}_1(\mathbf{x}, \mathbf{y})$, $\mathbf{y}' = \mathbf{q}_{2,\lambda}(\mathbf{x}, \mathbf{y})$.

4   Algorithms

Based on the proof of Theorem 1, finding a dissipativity-preserving quadratization can be done in two following steps:

(Step 1) finding an inner-quadratic quadratization
(Step 2) modifying the corresponding quadratic system to achieve dissipativity at the given equilibria.

In this section, we give algorithms for both steps. Section 4.1 shows how to modify the quadratization algorithm from [7] to search for inner-quadratic quadratizations. Using this algorithm as a building block, we give a general algorithm for computing dissipativity-preserving quadratizations in Section 4.2.

4.1   Computing inner-quadratic quadratization

Our algorithm follows the general *Branch-and-Bound (B&B)* paradigm [28] and is implemented based on the optimal monomial quadratization algorithm from [7, Section 4]. Therefore, we will describe the algorithm briefly, mainly focusing on the differences with the algorithm from [7].

We define each subproblem [7, Definition 3.3] as a set of new monomial variables $\{y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x})\}$, and the subset of the *search space* [7, Definition 3.1] for the subproblem will be the set of all *quadratizations* including these new variables. To each subproblem $\{y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x})\}$, the algorithm from [7] assigns a set of generalized variables V (new variables, $\mathbf{x}$'s, and 1) and a set of nonsquares NS (monomials in the right-hand side which are not quadratic in the generalized variables [7, Definition 4]). Additionally, we define the set of non-inner-quadratic new variables NQ which consist of all the monomials among $y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x})$ which are not quadratic in $\{y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x}), x_1, \ldots, x_n\}$. In particular, a subproblem is an inner-quadratic quadratization *if and only if* NS $= \varnothing$ and NQ $= \varnothing$. Note that NS and NQ are disjoint since NQ $\subseteq$ V and V $\cap$ NS $= \varnothing$.

Example 4. The notation previously introduced will now be demonstrated through the system $x' = x^4 + x^3$ (taken from [7, Example 4], to display the difference between two algorithms). We consider a subproblem with one already added new variable $y_1(x) := x^3$ (so, $y_1' = 3x^2 x' = 3x^6 + 3x^5$). In this case, we have

$$V = \{1, x, x^3\}, \quad V^2 = \{1, x, x^2, x^3, x^4, x^6\}, \quad NS = \{x^5\}, \quad \underline{NQ = \{x^3\}}.$$

The algorithm starts from the subproblem $\varnothing$. For every iteration, we select one element m from NS $\cup$ NQ (using a heuristic score function [7, Section 4.1])

and compute all the decompositions of the form $m = m_1 m_2$, where m_1 and m_2 are monomials. If $m \in$ NS, for every such decomposition, we create a new subproblem by adding the elements of $\{m_1, m_2\} \setminus V$ and at least one new variable will be added due to the property of NS. If $m \in$ NQ, we only do this for the decompositions with $m_1 \neq 1$ and $m_2 \neq 1$.

We apply this operation recursively and stop when NS $\cup$ NQ $= \varnothing$ for each branch. Therefore, we can find all the possible inner-quadratic quadratization of the system. To improve the efficiency, we do not consider branches with more new variables than in already found answers and use versions of domain-specific pruning rules from [7]. The algorithm is summarized as Algorithm 1.

Algorithm 1: Computing optimal inner-quadratic quadratization

Input
- polynomial ODEs system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$.
- a set of already chosen new variables $y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x})$ (at the first call, $\varnothing$).
- the order N of the smallest inner-quadratic quadratization found so far (at the first call, $N = \infty$).

Output a more optimal inner-quadratic quadratization containing $y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x})$ if such quadratization exists.

(Step 1) If $y_1(\mathbf{x}), \ldots, y_\ell(\mathbf{x})$ is a inner-quadratic quadratization, that is, NS $= \varnothing$ and NQ $= \varnothing$, and $\ell < N$, **return** $y_1, \ldots, y_\ell$.

(Step 2) Select the element $m \in$ NS $\cup$ NQ with the smallest *score*, compute all the decompositions $m = m_1 m_2$ as a product of two monomials. If $m \in NQ$, we only consider the decompositions with $m_1 \neq 1$ and $m_2 \neq 1$.

(Step 3) For each decomposition $m = m_1 m_2$ from the previous step, we consider a subproblem $\{z_1, \ldots, z_\ell\} \cup (\{m_1, m_2\} \setminus V)$. If its size is less than N and none of the pruning rules apply, we run recursively on this subproblem and update N if a more optimal inner-quadratic quadratization has been found by the recursive call.

4.2   Computing dissipativity-preserving quadratization

Based on the proof of Theorem 1, the main idea behind the search for dissipativity-preserving quadratization is to start with any inner-quadratic quadratization, and replace the right-hand side for the new variables, $\mathbf{q}_2(\mathbf{x}, \mathbf{y})$, by $\mathbf{q}_2(\mathbf{x}, \mathbf{y}) - \lambda \mathbf{h}(\mathbf{x}, \mathbf{y})$ (see (8)) for increasing values of λ until the desired quadratization is found. The detailed algorithm is given as Algorithm 2, the proof of its correctness and termination is provided by Proposition 3, and a step-by-step example is given in Example 5.

Proposition 3. *Algorithm 2 always terminates and produces a correct output.*

Proof. We will start with proving the correctness. Note that since $\mathbf{g}, \mathbf{q}_1, \mathbf{q}_2$ computed in **(Step 1)** yield a quadratization of the input system, and $\mathbf{h}$ vanishes if

---

Algorithm 2: Computing a quadratization dissipative at all provided equilibria

---

Input polynomial ODE system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$ and a list of its dissipative equilibria $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$;

Output a quadratization of the system which is dissipative at $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$.

(Step 1) Compute an inner-quadratic quadratization of $\mathbf{x}' = \mathbf{p}(\mathbf{x})$ using **Algorithm 1**. Let the introduced variables be $y_1 = g_1(\mathbf{x}), \ldots, y_m = g_m(\mathbf{x})$. Let $\mathbf{q}_1(\mathbf{x}, \mathbf{y})$ and $\mathbf{q}_2(\mathbf{x}, \mathbf{y})$ be the right-hand sides of the quadratic system as in Definition 1. If the corresponding quadratic system is dissipative at $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$, **return** it.

(Step 2) Construct the stabilizers $\mathbf{h}(\mathbf{x}, \mathbf{y})$ for the quadratization from **(Step 1)** as in Definition 6, and set $\lambda = 1$.

(Step 3) While **True**

(a) Construct a quadratic system Σ_λ

$$\mathbf{x}' = \mathbf{q}_1(\mathbf{x}, \mathbf{y}), \ \mathbf{y}' = \mathbf{q}_2(\mathbf{x}, \mathbf{y}) - \lambda \mathbf{h}(\mathbf{x}, \mathbf{y}).$$

(b) Check if Σ_λ is dissipative at $(\mathbf{x}_i^*, \mathbf{g}(\mathbf{x}_i^*))$ for every $1 \leqslant i \leqslant \ell$ (using the Routh-Hurwitz criterion [15, Chapter XV]). If yes, **return** quadratization defined by $\mathbf{g}(\mathbf{x}), \mathbf{q}_1(\mathbf{x}, \mathbf{y}), \mathbf{q}_2(\mathbf{x}, \mathbf{y}) - \lambda \mathbf{h}(\mathbf{x}, \mathbf{y})$. Otherwise, set $\lambda = 2\lambda$.

$\mathbf{y}$ is replaced with $\mathbf{g}(\mathbf{x})$, then $\mathbf{g}, \mathbf{q}_1, \mathbf{q}_2 - \lambda \mathbf{h}$ yield a quadratization of the original system as well. Furthermore, if the algorithm returned at **(Step 3)**b, then this quadratization is dissipative at $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$.

The termination of the algorithm follows from the proof of Proposition 2. We observe that the constructed $\mathbf{q}_2 - \lambda \mathbf{h}$ is exactly $\mathbf{q}_{2,\lambda}$ in the notation of the proof, and it is shown that there exists λ_0 such that, for every $\lambda > \lambda_0$, $\mathbf{g}, \mathbf{q}_1, \mathbf{q}_{2,\lambda}$ is dissipative at $\mathbf{x}_1^*, \ldots, \mathbf{x}_\ell^*$. Since λ in the algorithm is doubled on each iteration of the while loop, it will at some point exceed λ_0, and the algorithm will terminate.

Example 5. We will illustrate how Algorithm 2 works with the following differential equation:

$$x' = -x(x - a)(x - 2a) \tag{10}$$

where a is a positive scalar parameter. The system's equilibria are $0, a, 2a$, and, among them, $x = 0$ and $x = 2a$ are dissipative. Regardless of the value of a, Algorithm 1 called at **(Step 1)** will produce an inner-quadratic quadratization with one new variable $y = x^2$ and quadratic system:

$$\begin{cases} x' = -xy + 3ax^2 - 2a^2x, \\ y' = -2y^2 + 6axy - 4a^2x^2 \end{cases}$$

The stabilizer computed at **(Step 2)** will be $h(x, y) = y - x^2$.

Now let us fix $a = 1$ and continue with **(Step 3)**. At **(Step 3)**a, we form a new quadratic system Σ_λ:

$$\begin{cases} x' = -xy + 3x^2 - 2x, \\ y' = -2y^2 + 6xy - 4x^2 - \lambda(y - x^2) \end{cases}$$

For $\lambda = 1, 2, 4, 8, \ldots$ we check the eigenvalues of its Jacobian at points $(0,0)$ and $(2,4)$. The Jacobian of Σ_λ is

$$J = \begin{bmatrix} -y + 6x - 2 & -x \\ 6y + 2\lambda x - 8x & -4y - \lambda + 6x \end{bmatrix}$$

The eigenvalues we get on different iterations of the while-loop are summarized in Table 1 (the ones with nonnegative real parts are bold).

λ	at $(0,0)$	at $(2,4)$
1	$-2, -1$	$-2,$ **3**
2	$-2, -2$	$-2,$ **2**
4	$-2, -4$	$-2,$ **0**
8	$-2, -8$	$-2, -4$

Table 1: Eigenvalues of the Jacobian of Σ_λ at **(Step 3)**b

From the table we see, that the algorithm will stop and return at $\lambda = 8$. Note that our implementation offers three way of verifying the dissipativity: by computing the eigenvalues directly numerically (with NUMPY) or symbolically (with SYMPY) or by using the Routh-Hurwitz criterion [15, Chapter XV], [4, Chapter 3] (via TBCONTROL package [30]). The numerical evaluation of eigenvalues is the fastest (see Tables 2 and 3) but does not yield fully rigorous guarantees, the other two methods may be slower but provide such guarantees.

As the value of a increases, the original system is more unstable at equilibrium $x_{eq} = 2a$, which requires a larger value of λ in order to make the system dissipative at (x_{eq}, x_{eq}^2). We compute the dissipative quadratization of the system (10) with different values of a and the running time for each method, which is presented in Table 2.

5 Case studies

The code for reproducing the results of the case studies below is available in the "Examples" folder of [1].

a	λ (output)	time (*Numpy*)	time (*Routh-Hurwitz*)	time (*Sympy*)
1	8	33.63	36.89	40.98
5	128	34.04	38.36	39.84
10	512	33.07	41.91	43.66
50	16384	33.38	41.18	54.70
100	65536	34.90	43.06	54.31

Table 2: Output λ value and runtimes (in milliseconds) with different methods for a of the system 10, results were obtained on a laptop Apple M2 Pro CPU @ 3.2 GHz, MacOS Ventura 13.3.1, CPython 3.9.1. Runtime is averaged over 10 executions.

5.1   Application to reachability analysis

The reachability problem is: given an ODE system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$, a set $S \subseteq \mathbb{R}^n$ of possible initial conditions, and a time $t \in \mathbb{R}_{>0}$, compute a set containing the set

$$\{\mathbf{x}(t) \mid \mathbf{x}' = \mathbf{p}(\mathbf{x}) \ \& \ \mathbf{x}(0) \in S\} \subseteq \mathbb{R}^n$$

of all points reachable from S at time t. One recent approach to this problem in the vicinity of a dissipative equilibrium $\mathbf{x}^*$ proposed by Forets and Schilling in [14] is to use Carleman linearization to reduce the problem to the linear case which is well-studied. However, the approach described in [14] relied on explicit bounds available only for quadratic systems under the assumption of dissipativity and weak nonlinearity (see [14, definition 1 and 2]). Algorithm 2 allows this restriction to be relaxed by computing a quadratization which preserves the dissipativity of $\mathbf{x}^*$.

We will illustrate this idea using the *Duffing* equation

$$x'' = kx + ax^3 + bx'$$

which describes a damped oscillator with non-linear restoring force. The equation can be written as a first-order system by introducing $x_1 := x, x_2 := x'$ as follows

$$x_1' = x_2, \quad x_2' = kx_1 + ax_1^3 + bx_2.$$

We take $a = 1, b = -1, k = 1$. Then the system will have three equilibria $\mathbf{x}^* = (0,0), (-1,0), (1,0)$, among which it will be dissipative only at the origin. Algorithm 1 finds an inner-quadratic quadratization for the system using a new variable $y(\mathbf{x}) = x_1^2$ resulting in the following quadratic system:

$$x_1' = x_2, \quad x_2' = ax_1 y + bx_2 + kx_1, \quad y' = 2x_1 x_2.$$

Obviously, the quadratization is an inner-quadratic quadratization as well. By applying Algorithm 2, we get $\lambda = 1$ with the following dissipative quadratization:

$$\begin{cases} x_1' = x_2 \\ x_2' = x_1 y + x_1 - x_2 \\ y' = -y + x_1^2 + 2x_1 x_2 \end{cases} \tag{11}$$

For the initial conditions $x_1(0) = 0.1$, $x_2(0) = 0.1$, $y(0) = x_1(0)^2 = 0.01$, system (11) satisfies the requirement of the algorithm from [14]. We apply the algorithm with truncation order $N = 5$ and report the result of the reachability analysis in Figure 2. The grey curve is the computed trajectory and the blue area is an upper bound for the reachable set.
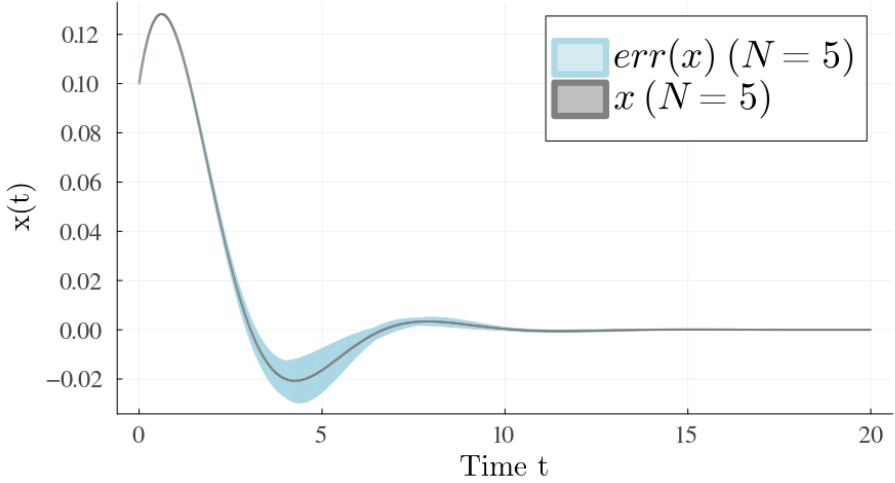

Fig. 2: Reachability analysis results with the computed trajectory (gray) and overapproximation of the reachable set (light blue). Initial condition $\mathcal{X}_0 = [0.1, 0.1, 0.01]$, truncation order $N = 5$, and the estimate reevaluation time $t = 4$ (see [14, Section 6.1]).

5.2 Preserving bistability

An ODE model is called *bistable* (or multistable) if it has at least two stable equilibria. This is a fundamental property for models in life sciences since such a model describes a system that can exhibit a switch-like behaviour, in other words, "make a choice" [12]. One of the smallest possible bistable models arising from a simple chemical reaction network [31, Table 1] is given by the following scalar ODE:

$$x' = k_1 x^2 - k_2 x^3 - k_3 x,$$

where k_1, k_2, k_3 are positive reaction rate constants. The equation has always one dissipative equilibrium at $x = 0$. It has two more equilibria as long as $k_1^2 > 4k_2 k_3$, and in this case, the largest of them will be dissipative as well. For any nonzero parameter values, the inner-quadratic quadratization computed by Algorithm 1 will consist of a single new variable $y(x) := x^2$ and the quadratic system:

$$x' = k_1 w - k_2 xw - k_3 x, \quad y' = 2k_1 xy - 2k_2 y^2 - 2k_3 y. \tag{12}$$

For the case-study, we pick $k_1 = 0.4, k_2 = 1, k_3 = 0.03$. For these parameter values, the dissipative equilibria are $x = 0$ and $x = 0.3$, and Algorithm 2 finds that (12) is dissipative at them already. The plot below shows that, indeed, the trajectories of (12) staring in the neighbouthoods of $(0, 0)$ and $(0.3, 0.09)$ converge to the respective equilibria (Fig. 3).
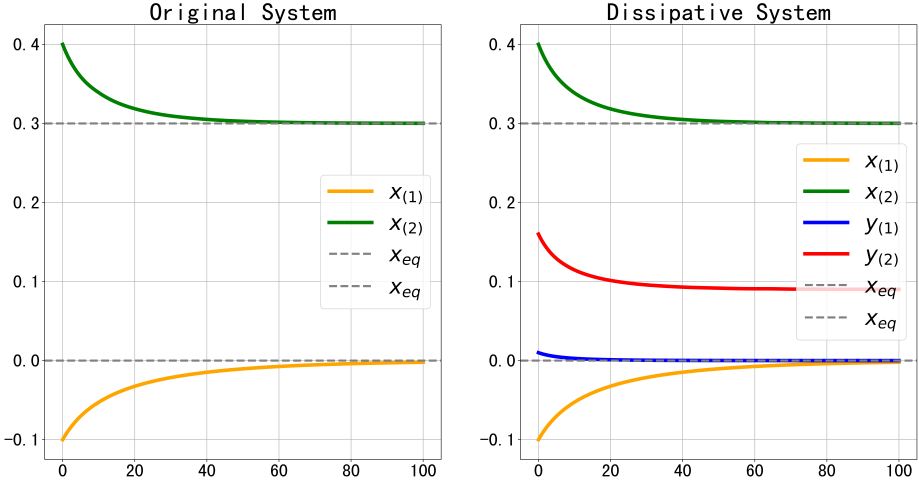

Fig. 3: Plot of the original equation and system (12) with initial state $\mathcal{X}_0 = [x_0, w_0] = [-0.1, 0.01]$ $(x_{(1)}, w_{(1)})$ and $\mathcal{X}_0 = [0.4, 0.16]$ $(x_{(2)}, w_{(2)})$.

5.3   Coupled Duffing oscillators

For a larger example, we will consider an ensemble consisting of Duffing oscillators from Section 5.1 which is an extended version of a pair of coupled oscillators from [3]. The model consisting of n oscillators is parametrized by a number $\delta \in \mathbb{R}$ and a matrix $A \in \mathbb{R}^{n \times n}$, and is defined by the following system:

$$\mathbf{x}'' = A\mathbf{x} - (A\mathbf{x})^3 - \delta \mathbf{x}',$$

where $\mathbf{x} = [x_1, \ldots, x_n]^\top$ are the positions of the oscillators, and $(A\mathbf{x})^3$ is the component-wise cube of vector $A\mathbf{x}$. Similarly to Section 5.1, we can rewrite this as a first-order system of dimension $2n$ by introducing new variables $\mathbf{z} = [z_1, \ldots, z_n]^\top$ for the derivatives of $\mathbf{x}$:

$$\dot{\mathbf{x}} = \mathbf{z}, \quad \mathbf{z}' = A\mathbf{x} - (A\mathbf{x})^3 - \delta \mathbf{z}.$$

Similarly, to [3, Table 1], if the eigenvalues of A are positive real numbers, then this system has 2^n dissipative equilibria. We run our code for $n = 1, \ldots, 8$ taking $\delta = 2$ and A being the tridiagonal matrix with ones on the diagonal and $\frac{1}{3}$ on the adjacent diagonals. Table 3 reports, for each n, the number of introduced variables and the times for computing inner-quadratic quadratization (Algorithm 1)

and making it dissipative at all 2^n equilibria (Algorithm 2 using NUMPY for the eigenvalue computation or the symbolics Routh-Hurwitz criterion). We can observe that numerical methods for checking the dissipativity scale well (given that the number of points grows exponentially) while symbolic methods become very costly as the dimension grows.

n	dimension	# equilibria	# new vars	time (inner-quadratic)	time (dissipative) NUMPY	ROUTH-HURWITZ
1	2	2	1	0.02	0.05	0.07
2	4	4	2	0.07	0.19	0.65
3	6	8	4	0.20	0.74	36.57
4	8	16	5	0.39	1.62	1179.33
5	10	32	7	0.72	4.30	> 2000
6	12	64	9	1.20	11.28	> 2000
7	14	128	10	1.75	28.23	> 2000
8	16	256	12	2.63	78.70	> 2000

Table 3: Runtimes (in seconds) for n coupled Duffing oscillators, results were obtained on a laptop with the following parameters: Apple M2 Pro CPU @ 3.2 GHz, MacOS Ventura 13.3.1, CPython 3.9.1.

6   Conclusions

While various quadratization techniques have been used recently in a number of application areas, and in most of the cases this was primarily involving numerical simulations, we are not aware of prior general results on the stability properties of the quadratized systems. In this paper, we studied quadratizations that preserve dissipativity at prescribed equilibria. First, we have shown that, for any set of dissipative equilibria such a quadratization exists. Then we have presented an algorithm capable of computing a quadratization with this property with dimension low enough to be of interest for applications. We showcase the algorithm on several case studies, including examples from reachability analysis and chemical reaction network theory.

The key ingredient of our algorithm is the computation of a quadratization (we call it inner-quadratic) which gives us substantial control over the stability properties of the quadratized system. We expect that this construction will be useful for further research in this direction.

In future research, we plan to extend the results of the paper in different directions. One natural problem is to extend the results and algorithms from the present paper beyond polynomial systems, for example, by designing an algorithm for dissipativity-preserving polynomialization. Additionally, exploring the preservation of other stability properties, such as limit cycles, attractors, and Lyapunov functions, is another promising avenue for research.

References

1. Dissipative-Quadratization Package (2023), https://github.com/yubocai-poly/DQbee
2. Appelroth, G.G.: Основная форма системы алгебраическихъ дифференціальныхъ уравненій. Sbornik: Mathematics **23**(1), 12–23 (1902), http://mi.mathnet.ru/sm6683
3. Balamurali, R., Kengne, L.K., Rajagopal, K., Kengne, J.: Coupled non-oscillatory Duffing oscillators: Multistability, multiscroll chaos generation and circuit realization. Physica A: Statistical Mechanics and its Applications **607**, 128174 (2022), https://doi.org/10.1016/j.physa.2022.128174
4. Bavafa-Toosi, Y.: Introduction to Linear Control Systems. Elsevier (2019). https://doi.org/10.1016/c2016-0-03896-2, http://dx.doi.org/10.1016/C2016-0-03896-2
5. Benner, P., Breiten, T.: Two-sided projection methods for nonlinear model order reduction. SIAM Journal on Scientific Computing **37**(2), B239–B260 (2015)
6. Bychkov, A., Issan, O., Pogudin, G., Kramer, B.: Exact and optimal quadratization of nonlinear finite-dimensional non-autonomous dynamical systems (2023), https://arxiv.org/abs/2303.10285
7. Bychkov, A., Pogudin, G.: Optimal monomial quadratization for ODE systems. In: Flocchini, P., Moura, L. (eds.) Combinatorial Algorithms. pp. 122–136. Springer International Publishing, Cham (2021)
8. Carothers, D.C., Parker, G.E., Sochacki, J.S., Warne, P.G.: Some properties of solutions to polynomial systems of differential equations. Electron. J. Diff. Eqns. **2005**(40), 1–17 (2005), https://eudml.org/doc/125330
9. Carothers, D.C., Parker, G., Sochacki, J.S., Warne, P.G.: Some properties of solutions to polynomial systems of differential equations. Electronic Journal of Differential Equations (EJDE) p. Paper No. 40 (2005), http://eudml.org/doc/125330
10. Carravetta, F.: Global exact quadratization of continuous-time nonlinear control systems. SIAM Journal on Control and Optimization **53**(1), 235–261 (2015), https://doi.org/10.1137/130915418
11. Carravetta, F.: On the solution calculation of nonlinear ordinary differential equations via exact quadratization. Journal of Differential Equations **269**(12), 11328–11365 (2020), https://doi.org/10.1016/j.jde.2020.08.028
12. Craciun, G., Tang, Y., Feinberg, M.: Understanding bistability in complex enzyme-driven reaction networks. Proceedings of the National Academy of Sciences **103**(23), 8697–8702 (2006), https://doi.org/10.1073/pnas.0602767103
13. Fages, F., Le Guludec, G., Bournez, O., Pouly, A.: Strong Turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs. In: Computational Methods in Systems Biology: 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27–29, 2017, Proceedings 15. pp. 108–127. Springer (2017), https://doi.org/10.1007/978-3-319-67471-1_7
14. Forets, M., Schilling, C.: Reachability of weakly nonlinear systems using Carleman linearization. In: Lecture Notes in Computer Science, pp. 85–99. Springer International Publishing (2021), https://doi.org/10.1007%2F978-3-030-89716-1_6
15. Gantmacher, F.R.: The theory of matrices. Chelsea Publishing Company (1984)
16. Gözükirmizi, C., Demiralp, M.: Solving ODEs by obtaining purely second degree multinomials via branch and bound with admissible heuristic. Mathematics **7**(4) (2019), https://www.mdpi.com/2227-7390/7/4/367

17. Gu, C.: QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **30**(9), 1307–1320 (2011), https://doi.org/10.1109/TCAD.2011.2142184

18. Guillot, L., Cochelin, B., Vergez, C.: A generic and efficient Taylor series–based continuation method using a quadratic recast of smooth nonlinear systems. International Journal for numerical methods in Engineering **119**(4), 261–280 (2019), https://doi.org/10.1002/nme.6049

19. Guillot, L., Cochelin, B., Vergez, C.: A Taylor series-based continuation method for solutions of dynamical systems. Nonlinear Dynamics **98**(4), 2827–2845 (2019), https://doi.org/10.1007/s11071-019-04989-5

20. Hemery, M., Fages, F.: Algebraic biochemistry: A framework for analog online computation in cells. In: Computational Methods in Systems Biology, pp. 3–20 (2022), https://doi.org/10.1007/978-3-031-15034-0_1

21. Hemery, M., Fages, F., Soliman, S.: On the complexity of quadratization for polynomial differential equations. In: Abate, A., Petrov, T., Wolf, V. (eds.) Computational Methods in Systems Biology. pp. 120–140. Springer International Publishing, Cham (2020), https://doi.org/10.1007/978-3-030-60327-4_7

22. Hindmarsh, A.C.: ODEPACK, a systemized collection of ODE solvers. Scientific computing (1983)

23. Hubbard, J.H., West, B.H.: Differential Equations: A Dynamical Systems Approach. High-dimensional Systems. Springer (1995)

24. Kerner, E.H.: Universal formats for nonlinear ordinary differential systems. Journal of Mathematical Physics **22**(7), 1366–1371 (1981)

25. Kramer, B., Willcox, K.: Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. AIAA Journal **57**(6), 2297–2307 (2019), https://doi.org/10.2514/1.J057791

26. Kramer, B., Willcox, K.: Balanced truncation model reduction for lifted nonlinear systems. In: Beattie, C., Benner, P., Embree, M., Gugercin, S., Lefteriu, S. (eds.) Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas, pp. 157–174. Springer International Publishing, Cham (2022), https://doi.org/10.1007/978-3-030-95157-3_9

27. Lagutinskii, M.N.: Къ вопросу о простѣйшей формѣ системы обыкновенныхъ дифференціальныхъ уравненій. Sbornik: Mathematics **27**(4), 420–423 (1911), http://mi.mathnet.ru/sm6583

28. Morrison, D.R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C.: Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization **19**, 79–102 (2016), https://doi.org/10.1016/j.disopt.2016.01.005

29. Petzold, L.: Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. SIAM journal on scientific and statistical computing **4**(1), 136–148 (1983)

30. Sandrock, C.: tbcontrol package, https://github.com/alchemyst/Dynamics-and-Control

31. Wilhelm, T.: The smallest chemical reaction system with bistability. BMC Systems Biology **3**(1) (2009), https://doi.org/10.1186/1752-0509-3-90