ESAIM: COCV 27 (2021) S9
https://doi.org/10.1051/cocv/2020057

ESAIM: Control, Optimisation and Calculus of Variations
www.esaim-cocv.org

A FIXED POINT ALGORITHM FOR IMPROVING FIDELITY OF QUANTUM GATES[*]

Paulo Sergio Pereira da Silva[1,**], Pierre Rouchon[2]
and Hector Bessa Silveira[3]

Abstract. This work considers the problem of quantum gate generation for controllable quantum systems with drift. It is assumed that an approximate solution called seed is pre-computed by some known algorithm. This work presents a method, called Fixed-Point Algorithm (FPA) that is able to improve arbitrarily the fidelity of the given seed. When the infidelity of the seed is small enough and the approximate solution is attractive in the context of a tracking control problem (that is verified with probability one, in some sense), the Banach Fixed-Point Theorem allows to prove the exponential convergence of the FPA. Even when the FPA does not converge, several iterated applications of the FPA may produce the desired fidelity. The FPA produces only small corrections in the control pulses and preserves the original bandwidth of the seed. The computational effort of each step of the FPA corresponds to the one of the numerical integration of a stabilized closed loop system. A piecewise-constant and a smooth numerical implementations are developed. Several numerical experiments with a N-qubit system illustrates the effectiveness of the method in several different applications including the conversion of piecewise-constant control pulses into smooth ones and the reduction of their bandwidth.

Mathematics Subject Classification. 93D05, 93D30, 8108.

Received November 11, 2019. Accepted August 18, 2020.

1. Introduction

Quantum control theory is now an important subject whose introduction is presented in some textbooks [1, 3]. Many important problems of quantum control are essentially versions of open loop motion planning problems over the unitary Lie-group $\mathrm{U}(n)$, which can be considered by controllability theory, Lie Group decompositions, and optimal control theory [3]. In general, controllability theory provides only existence theorems. It is true that decomposition theory may furnish piecewise-constant control pulses as a solution of motion planning, but this solution is restricted to small dimensions n of systems state. Optimal control may furnish fast solutions in the context of state steering and quantum gate generation [13, 14]. However, when the dimension of the state

[*]The first author was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, Project 305546/2016-3, and by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brazil, Project 18/17463-7.

Keywords and phrases: Controllability, quantum control, right-invariant systems, Lyapunov stability, Banach Fixed-Point Theorem.

[1] Polytechnic School – PTC, University of São Paulo (USP), São Paulo, SP, Brazil.
[2] Centre Automatique et Systèmes, Mines ParisTech, Paris, France.
[3] Departamento de Automação e Sistemas (DAS), Universidade Federal de Santa Catarina (UFSC), Florianópolis, SC, Brazil.

[**] Corresponding author: paulo@lac.usp.br

Article published by EDP Sciences

n is large, optimal control may be impossible to be implemented due to complexity issues [23]. For quantum systems, for instance N-coupled qubits, the dimension of the state is 2^N, which gives a propagator that is a $2^N \times 2^N$ complex matrix. This fact justifies the need of new methods, like numerical optimal control, Lyapunov methods and so on. Lyapunov stabilization [6, 7, 11, 12, 15, 24–27] may be applicable for large n, but they generate slow solutions (in the sense that that the final time must be large) when compared to the ones that are generated by optimal control, when this last approach is applicable. Hence the study of numerical methods that can be applied for large n and also produces a fast solution is a relevant field of research, indeed.

In this paper, a piecewise-constant implementation of an algorithm means that the control pulses to be generated are piecewise-constant functions of time, whereas a smooth implementation means that the control pulses to be generated are smooth functions of time. The Krotov method [23], GRAPE [4, 8], and RIGA [19] in the piecewise-constant setting and GOAT [10], CRAB [20] or RIGA in the smooth setting, are efficient methods for solving the quantum gate generation problem. A comparison between the methods of Krotov, GRAPE, CRAB and GOAT for small dimensions (two and three qubits) is presented in [21]. A full comparison between the known methods (Krotov, GRAPE, CRAB, GOAT and RIGA) that includes high-dimensional systems is yet to be done, although some comparisons between the piecewise-constant version of RIGA and GRAPE were performed in [19] showing that RIGA is indeed a promising method. A smooth MATLAB® implementation[1] of RIGA is available in [16].

The basic problem of quantum gate generation considers controllable right-invariant systems with m inputs $u = (u_1, \dots, u_m) \in \mathbb{R}^m$ and state $X(t)$ evolving on the unitary Lie group $\mathrm{U}(n)$, that is, the set of n-square complex matrices X such that $X^\dagger X = I$, where $X^\dagger$ is the conjugate transpose of X and I is the identity matrix. Here $\mathfrak{u}(n)$ denotes its Lie-algebra formed by the n-square anti-Hermitian complex matrices, that is $\sigma \in \mathfrak{u}(n)$ if $\sigma^\dagger = -\sigma$. In this paper, $\|X\|$ stands for the Frobenius norm of a complex matrix. In the context of coherent control, this question relies on an open loop steering problem. The models of such quantum systems are of the form

$$\dot{X}(t) = -\imath \left(H_0 + \sum_{k=1}^{m} u_k(t) H_k \right) X(t) = S_0 X(t) + \sum_{k=1}^{m} u_k(t) S_k X(t), X(0) = I \tag{1.1}$$

where $X \in \mathrm{U}(n)$ is the state (propagator), $S_0 = -\imath H_0, S_k = -\imath H_k \in \mathfrak{u}(n)$, and $u_k(t) \in \mathbb{R}$ are the controls.

The Quantum Gate Generation Problem is defined by:

Problem 1.1. (Quantum Gate Generation Problem) fix $X_{goal} \in \mathrm{U}(n)$ associated to the desired goal quantum gate $X_{goal} \in \mathrm{U}(n)$ to be generated. Fix a desired final time $T_f > 0$. One says that this problem is solvable if it is possible to compute open-loop controls $u : [0, T_f] \to \mathbb{R}^m$ such that system (1.1) is steered from $X(0) = I$ to some $X(T_f) \in \mathrm{U}(n)$ for a desired final fidelity.

In quantum control, a fidelity function $\mathcal{F} : \mathrm{U}(n) \to [0, 1]$ is a measure of the precision of the solution of a given problem[2]. In this paper, the infidelity function $\mathcal{I} : \mathrm{U}(n) \to [0, 1]$ will be defined by

$$\mathcal{I}(\widetilde{X}) = 1 - \left(\frac{1}{n} \|\mathrm{trace}(\widetilde{X})\| \right)^2 \tag{1.2}$$

and the fidelity function is given by $\mathcal{I}(\widetilde{X}) = 1 - \mathcal{F}(\widetilde{X})$. The gate fidelity and infidelity are defined respectively by $\mathcal{F}(X_{goal}^\dagger X(T_f))$ and $\mathcal{I}(X_{goal}^\dagger X(T_f))$, as considered in [9].

In this paper, it is assumed that an approximate solution of the Problem 1.1 is given. This work presents a method, called Fixed-Point Algorithm (FPA) that is able to improve the fidelity of the quantum gate generation. Before describing the FPA, let us give a context of the previous contributions of the authors to this problem. Inspired by the Coron's return method [2], the authors have developed a method of quantum gate generation

[1] Mainly based on the same ideas of Section 3.2 for integrating numerically the quantum system.

[2] The fidelity function normally is related to the probability of the correctness of the result after a measure.

for driftless systems [24]. However, this method is not iterative, and hence the solutions are slow, that is, a large final time T_f that depends on the rate of convergence of the Lyapunov tracking control is needed. In [25], the method of [24] was generalized for systems with drift. However, it also generates slow solutions when compared for instance to GRAPE [8, 9]. In [19], we have introduced an iterative method, called RIGA, that produces results that are compatible with GRAPE, with the advantage that RIGA may also generate smooth controls. Note that GRAPE is conceived in a way that it can only generate piecewise-constant controls, and this may be a disadvantage for the physical implementation, since the small discontinuities of the pulses can be filtered by the communication channel that links the control pulse generator to the quantum device, generating distortions that will produce a fidelity degradation. In the present paper, it is shown that RIGA may be combined with FPA in order to shorten the runtime of the overall computations (see Tab. 1 of Sect. 4.2). Section 4.3 shows how to convert piecewise-constant control pulses (that could be generated by GRAPE) to smooth ones, improving the fidelity. Section 4.5 shows that the FPA can be used for shortening the bandwidth of a pre-computed control, also improving the fidelity. These potential applications enhances the interest of the FPA in quantum control.

The main ideas of the algorithm will be presented now. The original approximated solution will be called the seed of the FPA (see Def. 2.1). Under certain conditions, the FPA is a virtually exact method in the sense that the final Frobenius error $\epsilon_\ell = \|X(T_f) - X_{goal}\|$ that is obtained in the step ℓ is such that $\lim_{\ell\to\infty} \epsilon_\ell = 0$ (exponentially). One first obtains, by right translation of the approximate solution, a new trajectory $\overline{X}(t)$ such that its final condition $\overline{X}(T_f)$ is exactly equal to X_{goal}. The idea is then to consider the trajectory $\overline{X}(t)$ as a reference trajectory and to apply a stabilizing feedback of the corresponding tracking problem, as done in previous works of the authors [24, 25]. The novelty of this work is that we consider the construction of an adequate right-translation $R \in \mathrm{U}(n)$ that corrects the action of the feedback in order to obtain an exact generation of quantum gate. The existence of this exact correction R is based on the Banach Fixed-Point Theorem. The FPA is an algorithm that obtains an approximation $R_\ell \in \mathrm{U}(n)$ of this exact correction, where R_0 is the identity matrix and $R_\ell = \mathcal{G}(R_{\ell-1})$, where the map $\mathcal{G}$ is the flow of a closed loop stabilized system. Such map $\mathcal{G}$ will be a contraction and R_ℓ will converge exponentially when: (a) the error $\delta = \epsilon_0$ of the seed is small enough; and (b) the seed is attractive in the context of the tracking control. This second assumption (b) is shown to be satisfied with probability one with respect to the choice of the seed $\overline{u}(t)$ in a sense to be precised later.

This technique may be viewed as a "fine tuning" of other techniques, and it produces small corrections of the original input $u(t)$ of the approximate solution that are able to obtain the desired final precision. It will be clear that it suffices to show that such map $\mathcal{G}$ is a contraction and the proof of Banach Fixed-Point will assure the exponential convergence of R_ℓ to R. It will be shown that R_ℓ can be computed by successive numerical integrations of a closed-loop dynamics in the interval $[0, T_f]$.

Normally, the solution that is generated by the seed must have a good fidelity in order to assure the convergence of FPA described in Algorithm 1. In fact what is shown in this paper is a local result, just because the map $\mathcal{G}$ is only a "local contraction" and the radius of this domain cannot be estimated *a priori*. However, we shall present an iterative version of FPA (see Algorithm 2). For this algorithm, even if the map $\mathcal{G}$ is not a contraction, iterative applications of FPA can produce the desired fidelity. By comparing with RIGA [19], is not difficult to show that, if in each step of Algorithm 2 only one step of FPA is executed, then Algorithm 2 reduces[3] to RIGA. Otherwise, if various steps of FPA are executed, then Algorithm 2 behaves as an "accelerated RIGA", as shown in the numerical experiments of Section 4.2. However, as the Lyapunov-based feedback that is related to our choice of the Lyapunov function (see (2.1)) is not bounded, if the fidelity of the seed is too low, Algorithm 2 may generate a huge control effort (eventually giving rise to numerical problems). In this case the combination of RIGA and FPA is indicated because RIGA always works "far" from the singular points of the Lyapunov function.

Two different numerical implementations of FPA are considered. The first one considers piecewise-constant control pulses, and it is compatible with the solutions provided by GRAPE, Krotov and the piecewise-constant

[3] Without the protection against singular points of the Lyapunov function that RIGA provides.

version of RIGA. The second one considers smooth inputs, and it is compatible with GOAT, CRAB and the smooth version of RIGA.

Numerical experiments are presented here for an example that is also considered in [9, 19]. It consists of a coupled chain of N qubits and the goal is to implement a Hadamard gate for $N = 3, 4, \ldots, 10$. The seed is first generated with the algorithm RIGA[4] The FPA is then executed in order to improve the fidelity of the seed.

The examples of Section 4 illustrates that the FPA may be useful in the following four scenarios:

- First generate a piecewise-constant pulse that is an approximation of the solution of the problem and then use a piecewise-constant version of FPA in order to improve its fidelity.
- First generate a continuous pulse by some method and then use a smooth version of FPA in order to improve its fidelity;
- First generate a piecewise-constant pulse and then use a smooth version of FPA in order to convert it into a smooth pulse with the desired fidelity;
- When the seed has a "large" bandwidth, the FPA may be combined with a pre-filtration of the seed, and this process may be iterated, shortening the bandwidth of the control pulses, while preserving, or even improving, the desired fidelity.

Since the FPA produces small corrections of the control pulses, if the seed respects some restrictions of amplitude, bandwidth or other ones, these restrictions will be preserved up to small changes in the final solution. This is an important issue that may be included in the scenarios of applicability of the FPA. The idea is to generate the seed with all the necessary restrictions which may lead to a slow convergence and some difficulties to attain the desired fidelity. In this case, the application of FPA in combination to some method may shorten the total runtime of this combination with respect to the use of this method alone. This fact is illustrated in some numerical examples for the combination RIGA-FPA. A MATLAB® implementation of FPA is available in [16].

This work is organized as follows. Section 2 presents the FPA. Section 3 discusses the two different implementations of FPA. Section 4 presents the examples. Some numerical experiments cover the different scenarios of the applications of the FPA. Section 5 presents the convergence theorems concerning the FPA. Finally, Section 6 presents the concluding remarks. For convenience, many proofs were deferred to the Appendices.

2. Description of the fixed-point algorithm (FPA)

In order to give a complete description of the FPA, it will be necessary to recall some results of [25].

2.1. The closed loop error system

Consider the Lyapunov function $\mathcal{V}\colon \mathcal{W} \subset \mathrm{U}(n) \to [0, \infty)$ [25]:

$$\mathcal{V}(\widetilde{X}) = -\mathrm{Tr}\left(\frac{(\widetilde{X} - I)^2}{(\widetilde{X} + I)^2}\right) \geq 0, \tag{2.1}$$

where $\mathcal{W}$ is the open subset of $\mathrm{U}(n)$ defined as

$$\mathcal{W} = \{W \in \mathrm{U}(n) \mid \det(I + W) \neq 0\}. \tag{2.2}$$

It was shown in [25] that $\mathcal{V}(\widetilde{X})$ corresponds to a notion of distance between $\widetilde{X} \in \mathcal{W}$ and the identity matrix I. This notion of distance then naturally defines a (not bounded) notion of distance between two matrices X_1, X_2 of $\mathrm{U}(n)$ by considering the map $\mathrm{dist} : \mathrm{U}(n) \times \mathrm{U}(n) \to [0, \infty) \cup \{\infty\}$ defined by $\mathrm{dist}(X_1, X_2) = \mathcal{V}(X_1^\dagger X_2)$ when $X_1^\dagger X_2 \in \mathcal{W}$ and $\mathrm{dist}(X_1, X_2) = \infty$ when $X_1^\dagger X_2 \notin \mathcal{W}$.

[4] First results of the combination RIGA-FPA where reported in the conference paper [17]. The smooth implementations of RIGA and FPA are available in [16].

Consider the reference system (on $\mathrm{U}(n)$)

$$\dot{\overline{X}}(t) = S_0\overline{X}(t) + \sum_{k=1}^{m} \overline{u}_k(t) S_k \overline{X}(t), \quad \overline{X}(0) = \overline{X}_0, \tag{2.3}$$

As in [25] it will be considered that a reference input $\overline{u} : [0, T_f] \to \mathbb{R}^m$ is given and the corresponding solution $\overline{X} : [0, T_f] \to \mathrm{U}(n)$ of (2.3) is generated. Consider the (tracking) error matrix $\widetilde{X}(t) = \overline{X}^\dagger(t) X(t) \in \mathrm{U}(n)$. The dynamics of $\widetilde{X}(t)$ is given by

$$\dot{\widetilde{X}}(t) = \sum_{k=1}^{m} \widetilde{u}_k(t) \widetilde{S}_k(t) \widetilde{X}(t), \quad \widetilde{X}(0) = \widetilde{X}_0, \tag{2.4}$$

where $\widetilde{u}_k(t) = u_k(t) - \overline{u}_k(t)$ and $\widetilde{S}_k(t)$ is given by (2.5c). Assume that $\widetilde{X}(0) \in \mathcal{W}$. Consider the (smooth) feedback-law[5]

$$\widetilde{u}_k(t) = \widetilde{U}_k(\overline{X}(t), \widetilde{X}(t)) = K \mathrm{trace}\left[Z\left(\widetilde{X}(t)\right) \widetilde{S}_k(t) \right], \tag{2.5a}$$

$$u_k(t) = \widetilde{u}_k(t) + \overline{u}_k(t), \tag{2.5b}$$

$$\widetilde{S}_k(t) = \overline{X}^\dagger(t) S_k \overline{X}(t), \tag{2.5c}$$

where K is the chosen feedback gain[6], and Z is the map defined by

$$Z(\widetilde{X}) = \widetilde{X}(\widetilde{X} - I)(\widetilde{X} + I)^{-3}. \tag{2.5d}$$

This control law produces the closed-loop dynamics[7] (2.3)–(2.4)–(2.5). One may show that $\dot{\mathcal{V}} = -\sum_{k=1}^{m} \frac{4}{K} \widetilde{u}_k^2(t) \leq 0$. Several results of stability of this tracking error dynamics are studied in [24, 25] and those results are instrumental for the convergence results of the Section 5.

2.2. The fixed-point algorithm (FPA)

The input data of FPA is called seed, and it may be produced by any known method.

Definition 2.1. The pair $\{\overline{u}(\cdot), X_f\}$ is called seed whenever the input $\overline{u} : [0, T_f] \to \mathbb{R}^m$ with $\overline{u}(t) = (\overline{u}_1(t), \ldots, \overline{u}_m(t))$ is such that the solution of system (1.1) with $X(0) = I$ and $u(t) = \overline{u}(t)$ produces $X(T_f) = X_f$. Let $\delta > 0$. One says that the seed is a δ-approximated solution of Problem 1.1, if the final condition X_f is such that $\|X(T_f) - X_{goal}\| \leq \delta$.

The FPA is now presented:

Algorithm 1. Fixed Point Algorithm (FPA)
% BEGIN FPA
% BEGIN STEP 0 - Integrate the open loop system
Fix a desired infidelity **I**. Given a seed $\{\overline{u}(\cdot), X_f\}$ (see Def. 2.1), integrate numerically the open loop system (2.3) with $\overline{X}_0 = \overline{X}(0) = (X_f)^\dagger X_{goal}$ obtaining[8] the reference trajectory $\overline{X} : [0, T_f] \to \mathrm{U}(n)$. Initialize $R_{-1} = R_0 = I$. Choose $N_{fixed} > 2$ (maximum number of steps). Set $\ell = 0$ and Infidelity $= \mathcal{I}(X_{goal}^\dagger X_f)$.

[5] It is shown in [24] that $\mathcal{W}$ is a positively invariant set for the time-varying closed-loop system (2.4)–(2.5).

[6] For simplicity, the feedback gains are supposed to be equal for all $k = 1, \ldots, m$.

[7] The closed-loop system is also equivalently written as (1.1)–(2.3)–(2.5).

[8] The Proposition 5.5 assures that $\overline{X}(T_f) = X_{goal}$.

```
% END STEP 0 - Integrate the open loop system


WHILE Infidelity $> \mathbf{I}$ and $\ell < N_{fixed}$
    % BEGIN Step $\ell$ - integrate closed loop system
    Increment $\ell$
    $\widetilde{X}(0) = \overline{X}_0^\dagger R_{\ell-1}$
    Integrate numerically system (2.4)–(2.5) with initial condition $\widetilde{X}(0)$
    $R_\ell = \widetilde{X}(T_f)$
    IF $\|R_\ell - R_{\ell-1}\| \geq \|R_{\ell-1} - R_{\ell-2}\|$.
    THEN (Contraction test failed)
        EXIT WHILE
    ELSE (Contraction test passed)
        $u(\cdot) = \overline{u}(\cdot) + \widetilde{u}(\cdot)$
        Infidelity $= \mathcal{I}(R_{\ell-1}^\dagger R_\ell)$
    END
    % END Step $\ell$ - Integrate closed loop system
END % END While

% END FPA
```

Remark 2.2. It will be shown that the solution of system (1.1) with input $u(\cdot)$ that is generated in the step ℓ of Algorithm 1 is the solution of system (2.3)–(2.4) the in closed loop with $R_{\ell-1}$-corrected feedback (5.6). Then, by Proposition 5.11, the infidelity of the step ℓ of the FPA is $\mathcal{I}(R_{\ell-1}^\dagger R_\ell)$. Under mild assumptions, Theorem 5.9 imply that $\|R_\ell - R_{\ell-1}\|$ converges exponentially to zero, and so it will be easy to conclude from and Proposition 5.6 that the infidelity also converges to zero exponentially under the same conditions. If the FPA has improved the original infidelity of the reference trajectory, one may accept a feedback correction R_ℓ coming from a map that is not a contraction (*i.e.*, the "contraction test" has failed at some step ℓ). Such a situation has taken place in some of the examples of Section 4. If the "contraction test" has failed, them Algorithm 1 finishes, just because new iterations of FPA will not improve the fidelity. Then it is possible to repeat the FPA for $\bar{u}(\cdot) = u(\cdot)$, as done in Algorithm 2 below[9]. When X_f is not available, the open loop integration is performed with $\overline{X}(0) = I$ as in the implementation of the smooth version of FPA.

Algorithm 2. (Iterated FPA)
% BEGIN ALGORITHM 2.4
Fix a desired infidelity **I**. Assume that the seed $\{\overline{u}(\cdot), X_f\}$ (see Def. 2.1) is such that Infidelity $= \mathcal{I}(X_{goal}^\dagger X_f) > \mathbf{I}$.

```
WHILE Infidelity $> \mathbf{I}$
    Execute FPA with seed $\{\overline{u}(\cdot), X_f\}$ obtaining a control pulse $u(\cdot)$.
    Compute Infidelity $= \mathcal{I}(X_{goal}^\dagger X(T_f))$.
    Set $\overline{u}(\cdot) = u(\cdot)$
    Compute $X_f = X(T_f)$.
END % end While


% END ALGORITHM 2
```

[9] It is easy to show that, if the repetition of FPA fail in the contraction test in the first iteration every time, then Algorithm 2 reduces to RIGA.

3. The numerical implementations of FPA

This section will discuss the two ways of implementing the FPA, the piecewise-constant (discontinuous) and the smooth implementations. Since the heart of the FPA relies in the way that the open- and closed-loop systems are simulated, this section will be devoted to this aspect. In both implementations, the interval $[0, T_f]$ will be divided in N_{sim} equal parts, and the time t will be discretized at instants $t_s = s\delta, s = 0, 1 \ldots, N_{sim}$, where $\delta = T_f / N_{sim}$. By simplicity, the reference inputs $\overline{u}_k(t_s)$ and the input $u_k(t_s)$ will be denoted respectively by:

$$\overline{u}_k(t_s) = \overline{u}_{k_s} \tag{3.1a}$$

$$u_k(t_s) = u_{k_s}. \tag{3.1b}$$

In the piecewise-constant case, this means that

$$u_k(t_s + \tau) = u_{k_s}, \text{for } \tau \in [0, \delta), s = 0, 1, \ldots, N_{sim} - 1. \tag{3.2}$$

In the smooth case, for $\tau \in [0, \delta]$ the following linear interpolation is used as an approximation for the open-loop simulations:

$$u_k(t_s + \tau) = \left(\frac{\tau - \delta}{\delta}\right) u_{k_s} + \left(\frac{\tau}{\delta}\right) u_{k_{s+1}}, s = 0, 1, \ldots, N_{sim} - 1. \tag{3.3}$$

The discrete representation of the seed of the algorithm is the set of control inputs $\{\overline{u}_{k_s},\ \ k = 1, \ldots, m, s = 0, 1, \ldots, N_f\}$ that was produced by some algorithm, which produces a final condition $X(T_f) = X_f$. These control inputs are interpreted respectively as (3.2) with $N_f = N_{sim} - 1$ in the piecewise-constant case, and as (3.3) with $N_f = N_{sim}$ in the smooth case.

The step ℓ of the FPA produces the inputs $\{u_{k_s}, s = 0, 1, \ldots, N_{sim} - 1\}$, interpreted as piecewise-constant or smooth inputs (approximated by (3.3)), according to the case. By simplicity, denotes also $X_s = X(t_s)$, $\overline{X}_s = \overline{X}(t_s)$ for $s = 0, \ldots, N_{sim}$. In all implementations, a gain $K > 0$ is chosen.

3.1. The piecewise-constant implementation

The piecewise-constant implementation of FPA is based on the piecewise-constant implementation of RIGA [18]. Denote $T_s = S_0 + \sum_{k=1}^{m} u_{k_s} S_k$ and $\overline{T}_s = S_0 + \sum_{k=1}^{m} \overline{u}_{k_s} S_k$ where $S_k, k = 1, \ldots, m$ are the system matrices. The exponential of a matrix will be computed using a $4th$-order Padé approximation:

$$\exp(\delta S) \approx (I + \delta S/2 + \delta^2 S^2/12)(I - \delta S/2 + \delta^2 S^2/12)^{-1} \tag{3.4}$$

The open loop simulation reads:

```
% BEGIN Step 0 – open-loop simulation[10]
    $\overline{X}_0 = X_f^\dagger X_{goal}$
    FOR $s = 0, 1 \ldots N_{sim} - 1$
        $\overline{X}_{s+1} = \exp\left(\delta \overline{T}_s\right) \overline{X}_s$
    END
% END Step 0
```

The closed loop simulation corresponds to the simulation of system (2.4) in closed loop with the $R_{\ell-1}$-corrected feedback (5.6), and this reads:

```
% BEGIN Step $\ell$ – Closed-loop simulation
    $X_0 = I$
    $\widetilde{X}_0 = \overline{X}_0^\dagger$
    **FOR** $s = 0, 1 \ldots N_{sim} - 1$
        $\widetilde{X}_s = \overline{X}_s^\dagger X_s$
        **FOR** $k = 1, \ldots, m$
                $u_{k_s} = \overline{u}_{k_s} + \mathrm{Window}(s) K \mathrm{Tr}\left[Z\left(\widetilde{X}_s R_{\ell-1}\right)\overline{X}_s^\dagger S_k \overline{X}_s\right]$
        **END**
        $X_{s+1} = \exp\left(\delta T_s\right) X_s$
    **END**
    $R_\ell = \overline{X}_{N_{sim}+1}^\dagger X_{N_{sim}+1}$
% END Step $\ell$ - Closed Loop Simulation
```

Recall that map Z is given by (2.5d). The function Window(s) is normally equal to one for all s. In the case that one has used a window-function in RIGA (see the example in the next section), one may choose

$$\mathrm{Window}(s) = \begin{cases} 1, & \text{if } s = 1, 2, \ldots, N_f - 1 \\ 0, & \text{if } s = 0 \text{ and } s = N_f \end{cases} \tag{3.5}$$

where $N_f = N_{sim} - 1$. This will assure that the generated input will never be modified in the frontier points of the interval $[0, T_f]$.

3.2. The smooth implementation of FPA

The smooth implementation of FPA is based on the smooth implementation of RIGA. Both implementations are available in [16]. Using the same notation (3.1), recall that the control inputs are supposed to be smooth and in the open loop integration they will be approximated to piecewise-linear functions, which means that (3.3) holds. The idea of this implementation is to develop a 4th-order Runge–Kutta integration scheme for both closed-loop and open-loop cases. Since $\mathrm{U}(n)$ is not a Euclidean space, there is no sense in applying a Runge–Kutta method directly to the dynamics[11] (2.3)–(2.4). The idea is based on the method that is proposed in [5], and relies on a[12] smooth-map $\mathfrak{W}$ that is similar to the homographic function that was considered in [24] for defining the Lyapunov function $\mathcal{V}$ given by (2.1). Let $\mathfrak{W} : \mathcal{W} \subset \mathrm{U}(n) \to \Sigma \subset \mathfrak{u}(n)$ defined by $\widetilde{X} \mapsto (\widetilde{X} - I)(\widetilde{X} + I)^{-1}$, where $\Sigma \subset \mathfrak{u}(n)$ is the set of anti-Hermitian complex matrices σ such that $(\sigma - I)$ is invertible[13]. The inverse of $\mathfrak{W}$ is the smooth map $\mathfrak{X} : \Sigma \to \mathcal{W}$ such that $\mathfrak{X}(W) = -(W - I)^{-1}(W + I)$. From ([24], Eq. (9)), it is clear that $\mathcal{V}(\widetilde{X}) = \|\mathfrak{W}(\widetilde{X})\|^2$.

Some standard computations show easily that, if $\widetilde{X}(t) \in \mathrm{U}(n)$ is a solution of a differential equation $\frac{\mathrm{d}}{\mathrm{d}t}\widetilde{X}(t) = \widetilde{S}(t)\widetilde{X}(t)$, with $\widetilde{S}(t) \in \mathfrak{u}(n)$, then:

$$\dot{W}(t) = -\frac{1}{2}(W(t) - I)\widetilde{S}(t)(W(t) + I) \tag{3.6}$$

The equation (3.6) may be integrated numerically, instead of (1.1), with the advantage that $\mathfrak{u}(n)$ is a Euclidean space, and so the Runge-Kutta method may be applied in a natural way. Define the map $F : \Sigma \times \mathfrak{u}(n) \to \Sigma$ by:

$$F(W, \widetilde{S}) = -\frac{1}{2}(W - I)\widetilde{S}(W + I) \tag{3.7}$$

[11] The Runge–Kutta method in this case will generate non-unitary matrices.

[12] The Cayley transformation considered in [5] is given by $-\mathfrak{W}$.

[13] Recall that $\mathcal{W}$ is the set of complex matrices $\widetilde{X}$ such that $(\widetilde{X} + I)$ is invertible. It is easy to show that $(\mathfrak{W}(\widetilde{X}) - I) = -2(\widetilde{X} + I)^{-1}$ and so $(\mathfrak{W}(\widetilde{X}) - I)$ is always invertible, with inverse $-\frac{1}{2}(\widetilde{X} + I)$.

then equation (3.6) reads $\dot{W}(t) = F(W(t), \widetilde{S}(t))$. Each Runge–Kutta step of the open loop integration of (3.6) considers the interval $[t_s, t_{s+1}]$ with $W(t_s) = 0$ (corresponding to the identity in $\mathrm{U}(n)$), with posterior correction by right-invertibility. Note that $W(t)$ may not leave the region Σ, otherwise it will reach a singularity (and numerical problems will certainly occur). The simulation of the closed loop system does not have this problem, since $\mathcal{W}$ is invariant for the closed loop system just because the Lyapunov function is non-increasing [25]. So $W(t) = \mathfrak{W}(\widetilde{X}(t))$ will never leave $\Sigma \subset \mathfrak{u}(n)$. The details are presented in Appendix E.

4. Examples

This section will present some numerical experiments considering a coupled chain of N qubits for which the Hamiltonian is given by:

$$H(t) = J_0 \left[\sum_{s=1}^{N-1} \sigma_z^{(s)} \sigma_z^{(s+1)} \right] + J \left[\sum_{k=1}^{N} u_{x_k}(t) \sigma_x^{(k)} + u_{y_k}(t) \sigma_y^{(k)} \right] + J_g u_g(t) I, \tag{4.1}$$

where $\sigma_x, \sigma_y, \sigma_z$ are the Pauli matrices. The artificial input $u_g(t)$ is only a global phase input that may be disregarded in the application of the generated input, needed because our theory is developed in $\mathrm{U}(n)$ and not in $\mathrm{SU}(n)$. This example is a benchmark proposed in ([24], Eq. (9)), for testing a implementation of GRAPE. The goal is to generate a Hadamard gate. The Hadamard gate for $N = 1$ is defined by:

$$\mathcal{H}(1) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Then, one may obtain $X_{goal} = \mathcal{H}(N)$ inductively by the Kronecker product $\mathcal{H}(N) = \mathcal{H}(1) \otimes \mathcal{H}(N-1)$. The system (4.1) is considered with $J_0 = J = J_g = 2\pi 100\,\mathrm{MHz}$. The final time $T_f = 2N \times 10^{-9}(s)$ where N is the number of qubits. The dimension of the Hilbert space is $n = 2^N$ and $m = 2n + 1$. The experiments have considered the infidelity function given by (1.2). The value of $N_{sim} = 10N$ is considered in [9]. In this work, the value $N_{sim} = 20N$ is chosen in all numerical experiments. The feedback gains $K = \frac{1}{J}\mathcal{K}(N)$ where $\mathcal{K}(N)$ is the Nth entry of the vector $[10, 10, 10, 2, 2, 1, 0.5, 0.25, 0.125, 0.0625]$. The values of gains K were chosen with the aid of an option of the piecewise-constant MATLAB® $R2015a$ implementation of RIGA [18]. The same gains K are used both in the smooth and the piecewise-constant implementations of FPA. The smooth MATLAB® implementation of RIGA and FPA is available in [16]. All numerical experiments were done with MATLAB® $R2015a$ running on a PC having a `Intel(R) Core(TM) i7-8700 CPU @ 3.20\,GHz, 3.19\,GHz` processor, with 16G of RAM and in a Windows 10 environment.

4.1. Results with the piecewise-constant implementation of FPA

This subsection reports the results of the piecewise-constant implementation of FPA. The seed of FPA was generated by the piecewise-constant version of RIGA for $N = 8$ qubits obtaining a solution of an infidelity that is close to 0.005. The RIGA was executed with a Hamming-like window function (see Sect. E.2), which is an option for generating control pulses that are null at $t = 0$ and $t = T_f$. The window function used by FPA is given by (3.5), chosen in order to preserve the restrictions of the seed at the endpoints of the interval $[0, T_f]$. Algorithm 2 was executed for a desired infidelity given by 0.001. The RIGA was also used for generating a solution with infidelity 0.001 (without the FPA). The results are presented in Table 1, in which T_{RIGA} is the runtime of RIGA, T_{FPA} is the runtime of FPA. $\mathcal{I}_{RIGA}$ and $\mathcal{I}_{FPA}$ are respectively the infidelity of the solutions produced by RIGA and FPA. To prevent from possible numerical errors of the algorithms, these infidelities are computed by an open-loop simulation using the MATLAB® function expm instead of the Padé approximation (3.4), blue as a post-computation for precision evaluation (expm is not used in the method itself).

The Table 1 also presents the number N_{RIGA} of steps of RIGA, the number N_{calls} of times that the FPA was executed in Algorithm 2. Furthermore, N_{FPA} is the total sum of the number of steps of FPA that were

TABLE 1. Results of the simulations in the piecewise-constant case.

N (qubits)	$\mathcal{I}_{RIGA}$	N_{RIGA}	$\mathcal{I}_{FPA}$	N_{calls}	N_{FPA}	T_{FPA} (s)	$T_{RIGA}+$ T_{FPA} (s)
8.0	0.00491	78.0	9.94×10^{-4}	3.0	52.0	2311.0	5766.0
8.0	9.97×10^{-4}	191.0	0	0	0	0	9922.0

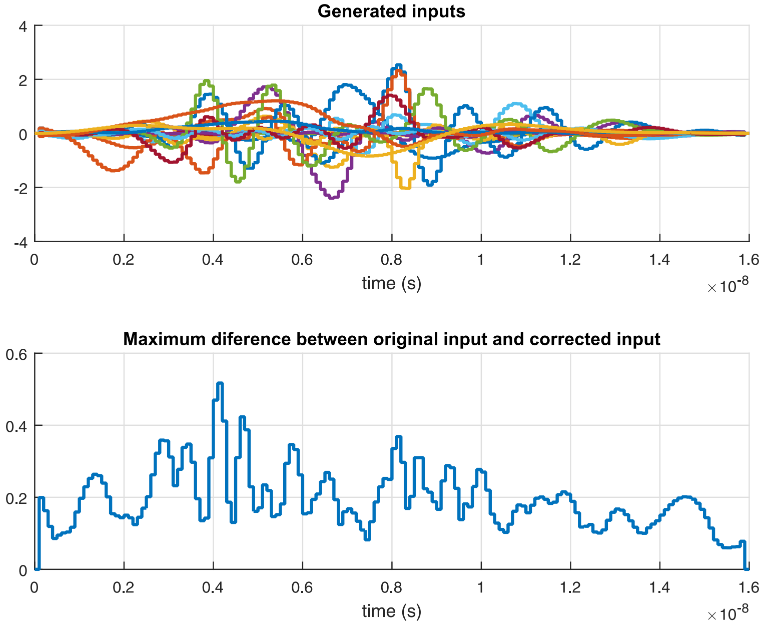

FIGURE 1. This figure illustrates the numerical experiment for $N = 8$ qubits in the piecewise-constant case. The seed of FPA was produced by RIGA with an infidelity that is close to 0.005. Top: plot of the control pulses produced by FPA with an infidelity that is less than 0.001. Bottom: plot of the function $\max\{|u_k(t) - \overline{u}_k(t)| : k = 1, 2, \ldots, m\}$, where $\overline{u}(t)$ is the seed input and $u(t)$ is the solution that is produced by the FPA.

executed in all calls of FPA of Algorithm 2. Note that $N_{calls} - 1$ is the number of times that the FPA was aborted because the contraction test has failed. It is then clear that the runtime for RIGA alone is greater than the runtime of the combination RIGA plus FPA. Similar behavior has been verified for $N = 2, 3, \ldots, 10$ qubits. Figure 1 shows the generated control pulses by the FPA for $N = 8$, illustrating that the FPA perform only small corrections in the control pulses.

4.2. Results with the smooth implementation of FPA

The numerical experiments of this section consider the smooth implementation of the FPA. The context of these experiments are essentially the same of the piecewise-constant case, but using the smooth versions of RIGA and FPA. The results of Table 2 are presented with the same notations of the Table 1. That table allows to conclude again that the combination of RIGA with FPA leads to lower runtime for the same desired infidelity with analogous qualities of the final control pulses. Figure 2 illustrates the fact that the FPA only performs small corrections, preserving the original desired restrictions of the seed. The only difference with respect to the piecewise-constant case is the fact that the numerical experiments were performed for $N = 3, 4, \ldots, 10$ qubits, illustrating the applicability of these algorithms for large dimensions. In fact, for $N = 10$ the dimension of

Table 2. Results of the simulations in the smooth case.

N	$\mathcal{I}_{RIGA}$	N_{RIGA}	$\mathcal{I}_{FPA}$	N_{calls}	N_{FPA}	T_{FPA}	$T_{RIGA}+$
(qubits)						(s)	T_{FPA} (s)
3.0	0.00489	24.0	9.67×10^{-4}	2.0	23.0	0.812	1.99
3.0	9.99×10^{-4}	55.0	0	0	0	0	2.59
4.0	0.00491	36.0	9.87×10^{-4}	1.0	28.0	2.09	6.34
4.0	9.87×10^{-4}	77.0	0	0	0	0	8.17
5.0	0.00499	266.0	9.84×10^{-4}	4.0	67.0	14.9	91.4
5.0	9.95×10^{-4}	477.0	0	0	0	0	135.0
6.0	0.00469	48.0	9.71×10^{-4}	2.0	31.0	27.0	81.8
6.0	9.87×10^{-4}	84.0	0	0	0	0	90.8
7.0	0.00484	93.0	9.26×10^{-4}	1.0	11.0	46.3	496.0
7.0	9.73×10^{-4}	130.0	0	0	0	0	632.0
7.0	0.00484	93.0	5×10^{-7}	16	9067	9.16×10^{4}	9.31×10^{4}
8.0	0.00485	75.0	0.0010	3.0	43.0	1566.0	4444.0
8.0	9.98×10^{-4}	148.0	0	0	0	0	5166.0
9.0	0.00494	189.0	0.0010	6.0	109.0	3.37×10^{4}	9.87×10^{4}
9.0	9.97×10^{-4}	362.0	0	0	0	0	1.24×10^{5}
10.0	0.00493	42.0	0.0010	5.0	82.0	1.86×10^{5}	2.9×10^{5}
10.0	9.96×10^{-4}	128.0	0	0	0	0	3.14×10^{5}

system matrices are $n = 2^{10} = 1024$ and $\mathfrak{u}(n)$ has dimension 2^{20}. To prevent from possible numerical errors of the algorithms, the infidelities $\mathcal{I}_{RIGA}$ and $\mathcal{I}_{FPA}$ that are presented in Table 2 are (re)computed by an open-loop simulation using 4th-order Runge-Kutta method with an step $\delta/2$ instead of $\delta = \frac{T_f}{T_{sim}}$.

Remark 4.1. When the seed $\overline{u}$ that is generated by RIGA does not respect restrictions, the runtime of the combination of RIGA-FPA is not that different from the runtime of RIGA alone. For instance, for $N = 8$ qubits, the RIGA was executed without the Hamming-like window option[14] in order to give an infidelity that is close to 0.005. Then the FPA was executed, produced a final infidelity of 0.001, in the same context of Table 1. Figure 3 shows the obtained control pulses. The runtime of RIGA-FPA was only 3% smaller than the runtime of RIGA alone. This behavior is also found in all other numerical experiments with $N = 3, 4, \ldots, 9$, when RIGA is performed without this window option, both in the smooth and in the piecewise-constant implementation. However, the use of a Hamming-like window is highly recommended in practical applications.

Another numerical experiment was done for the particular case where $N = 7$ qubits. We have generated the seed of Algorithm 2 with RIGA furnishing an infidelity approximately equal to 0.005 and we have executed Algorithm 2 with a desired infidelity of 5×10^{-7}, corresponding to line 11 of Table 2. After one day of runtime corresponding to 9067 steps of FPA, Algorithm 2 was able to produce the desired infidelity, indeed. It must be stressed that the desired infidelity must be compatible with the precision of the numerical integration[15]. If it is not the case, it would be necessary to increase N_{sim} (since the numerical integration step is $\delta = \frac{T_f}{N_sim}$). As the runtime of each step is proportional to N_{sim}, it is expected that runtime will increase accordingly.

4.3. Conversion of piecewise-constant control pulses into smooth ones of same fidelity

In this numerical experiment, done with $N = 6$ qubits, a piecewise-constant pulse $\{u_{k_s} : k = 1, \ldots, m, s = 0, 1, \ldots, N_{sim} - 1\}$ was generated by the piecewise-constant version of RIGA, assuring a final fidelity that is

[14] This window option of RIGA multiplies the gain K of the feedback by a Hamming-like function Window(t). This assures that the control pulses will be zero at the frontier points of $[0, T-_f]$. This options leads to a small bandwidth of the control pulses.

[15] The implementation of [18] gives the infidelity between X_f computed with step δ and X_f computed with step $\frac{\delta}{2}$. This infidelity must be compatible with the desired infidelity.

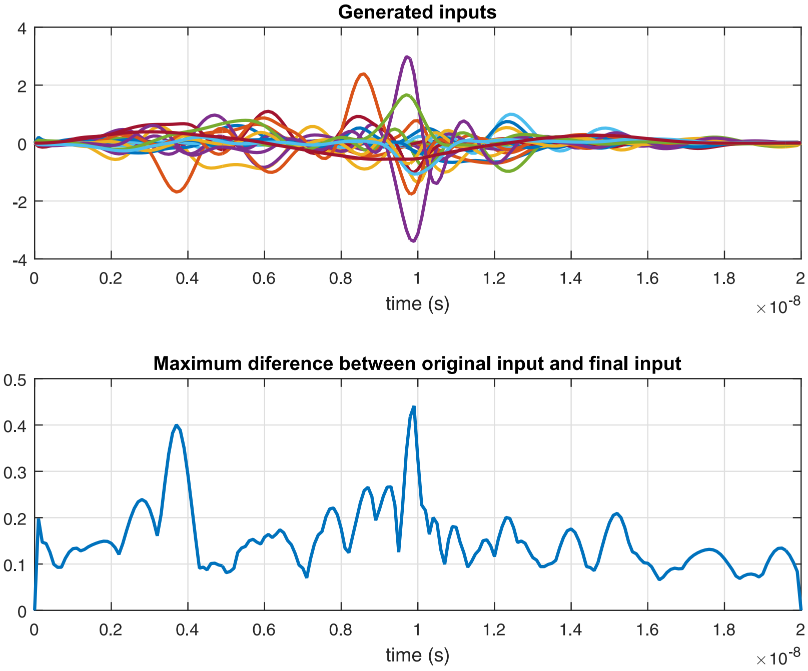

FIGURE 2. This figure illustrates the numerical experiment for $N = 10$ qubits in the smooth case. The seed of FPA was produced by RIGA with an infidelity that is close to 0.005. *Top*: plot of the control pulses produced by FPA with an infidelity that is less than 0.001. *Bottom*: plot of the function $\max\{|u_k(t) - \overline{u}_k(t)| : k = 1, 2, \ldots, m\}$, where $\overline{u}(t)$ is the seed input $u(t)$ is the solution that is produced by the FPA.

smaller than 0.001. This pulse was interpreted as continuous one by the smooth version of FPA simply by linear interpolation (3.3), considering $u_{k_{N_{sim}}} = u_{k_{N_{sim}-1}}$ for $k = 1, \ldots, m$. Under this assumption, the (smooth) open-loop simulation has produced an infidelity equal to 0.0105. Perhaps a reasonable conversion to a continuous signal could be obtained by filtering the control pulses, but this question will not be addressed here, since the smooth version of FPA is able to convert this signal to a continuous signal with any desired infidelity, chosen here to be less than 0.001 (to be precise, equal to 0.00094). The FPA has taken only 14 seconds to accomplish this task in 5 steps. Figure 4 shows the original piecewise-constant control pulses along with the superexposed smooth control pulses. Again, the FPA is able to perform small corrections to the original seed in order to obtain the desired fidelity.

4.4. The FPA preserves the bandwidth of its seed

The computation of the spectra of the seed and of the control pulses that are produced by FPA shows that it produces only small changes in the spectrum of the seed. Furthermore, these changes are concentrated in low frequencies. In particular the FPA preserves the original bandwidth of the seed $\overline{u}(t)$. For instance, in Figure 5, the absolute value FFT's of control signals of the smooth case with $N = 6$ qubits are depicted. This plot shows the average (in k) of the absolute value of the FFT's, that is, it is a plot of $\frac{1}{m}\sum_{k=1}^{m} \|\mathrm{FFT}(u_{k_s})\|$ and of $\frac{1}{m}\sum_{k=1} 1^m \|\mathrm{FFT}(\overline{u}_{k_s})\|$. The same qualitative behavior is found in all numerical experiments of this paper.

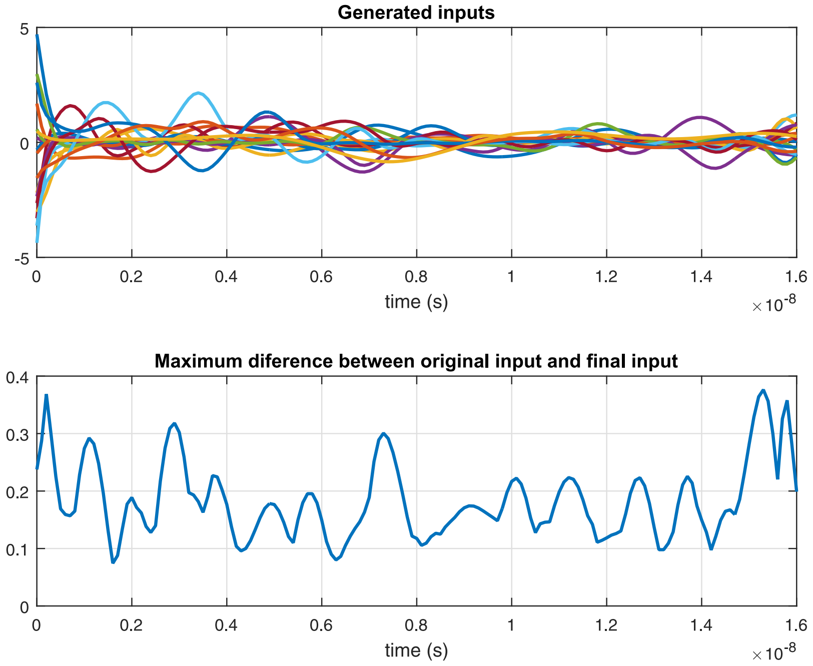

FIGURE 3. This figure illustrates the numerical experiment for $N = 8$ qubits in the smooth case. The seed of FPA was produced by RIGA with an infidelity that is close to 0.005 without the option of a Hamming-like window. Top: plot of the control pulses produced by FPA with an infidelity that is less than 0.001. Bottom: plot of the function $\max\{|u_k(t) - \overline{u}_k(t)| : k = 1, 2, \ldots, m\}$, where $\overline{u}(t)$ is the seed input $u(t)$ is the solution that is produced by the FPA.

4.5. The FPA for shortening the bandwidth

The result depicted in Figure 5 suggests the following idea. Given a seed (see Def. 2.1) formed by control pulses with a large bandwidth, but with the desired infidelity, then one may construct a filter and execute the following algorithm[16]:

Algorithm 3. (FILTERED FPA)
% BEGIN ALGORITHM 3
Fix a desired final infidelity $\mathbf{I}$ and an intermediary infidelity $\mathbf{I}_1 > \mathbf{I}$. Fix a number of steps N_{times}. Given a seed $\{\overline{u}(\cdot), X_f\}$ (see Def. 2.1) with small infidelity $\mathcal{I}(X_{goal}^\dagger X_f)$ but "large" bandwidth, execute the following

```
FOR $\ell = 1$ to $N_{times}$
    Filter $\overline{u}(\cdot)$, obtain $\overline{u}_{filtered}(\cdot)$
    Set $\overline{u}(\cdot) = \overline{u}_{filtered}(\cdot)$
    Execute Algorithm 2 for the filtered signal $\overline{u}$ for the desired infidelity $\mathbf{I}_1$.
    Algorithm 2 furnishes $u(\cdot)$
    Set $\overline{u}(\cdot) = u(\cdot)$
END
Execute Algorithm 2 for the filtered signal $\overline{u}$ for the desired infidelity $\mathbf{I}$.
Algorithm 2 furnishes $u(\cdot)$ corresponding to the desired infidelity $\mathbf{I}$
% END ALGORITHM 3
```

[16] It is not the aim of this paper to discuss Algorithm 4.2 very deeply, but only to establish a potential application of the FPA for reducing the bandwidth of the control pulses respecting the desired fidelity.

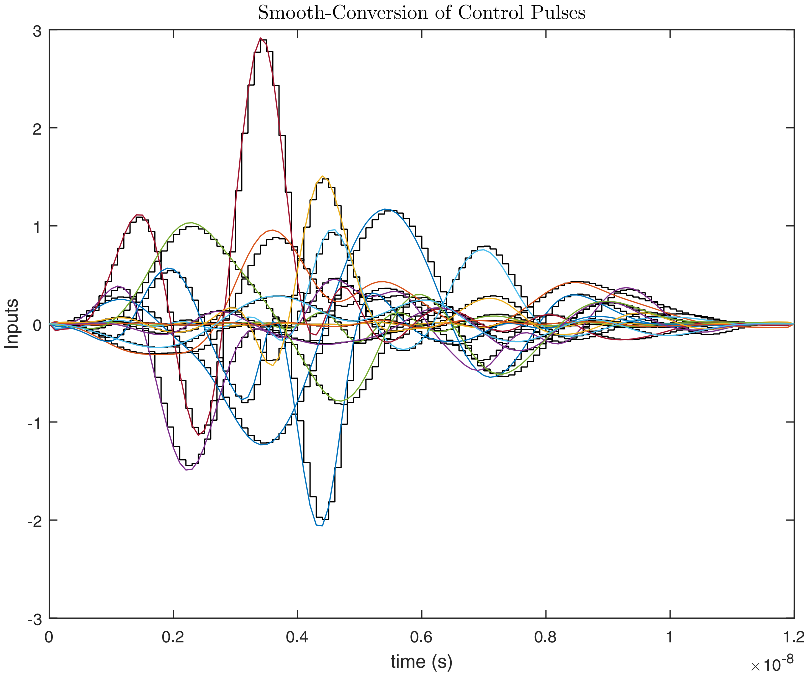

FIGURE 4. This figure superexposes the original piecewise-constant control pulses with the smooth control pulses that are obtained by the FPA, assuring a final infidelity of 0.00094.

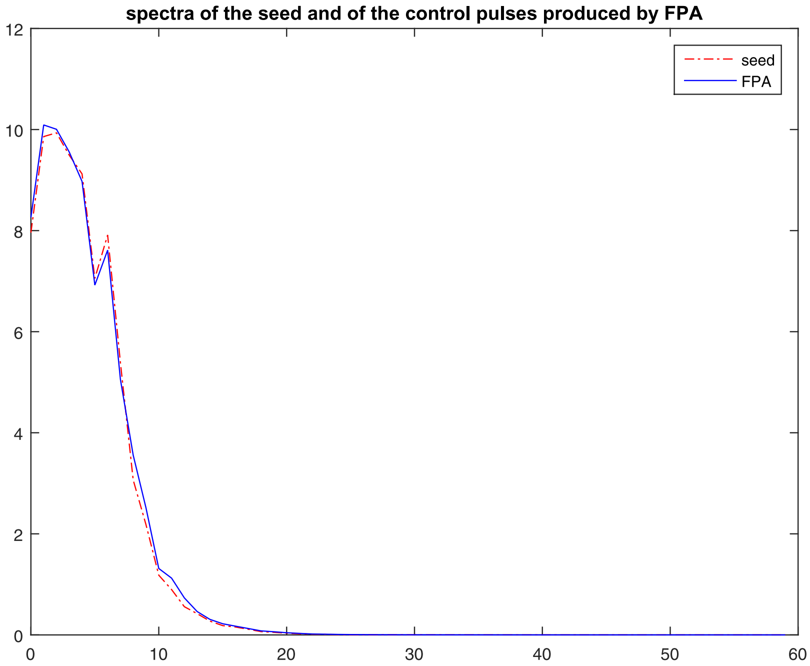

FIGURE 5. This figure shows the spectra (FFT) of the seed $\overline{u}(\cdot)$ and the solution $u(\cdot)$ that is produced by FPA in the smooth case with $N = 6$ qubits. Due to the known symmetry of FFT, only the first $N_{sim}/2$ points of the FFT are plotted.

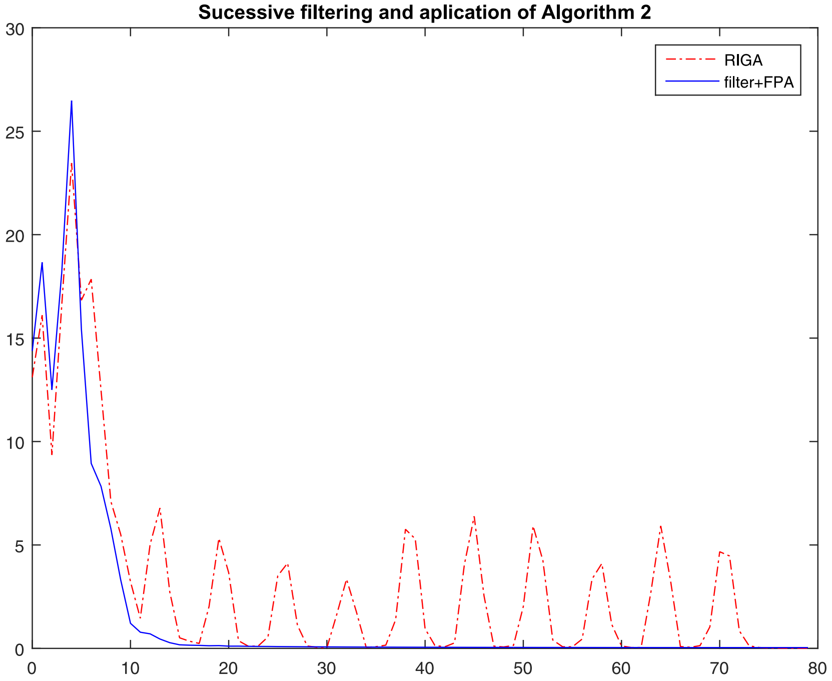

Figure 6. This figure shows the spectra (FFT) of the seed produced by RIGA with the legend “RIGA”. Note that the seed contains high frequencies components, that were completely eliminated by Algorithm 3, furnishing the solution with legend “filter+FPA”.

It must be pointed out is that the filtering in each step ℓ cannot increase the infidelity too much, otherwise the FPA works badly, and may present numerical instabilities since the closed loop simulation may attain points that are close to the frontier of $\mathcal{W}$. In fact, a limitation of FPA is the fact that its seed must have a sufficient small infidelity. To test this idea, the chosen parameters are $N = 4$ qubits, $\mathbf{I}_1 = 0.005$,$\mathbf{I} = 0.001$, and the low pass filter is a discrete one, that will filter the discrete signals $u_k(t_s)$ for $s = 0, 1, \ldots, N_{sim}$. The transfer function of the filter is $\frac{(z\alpha)^3}{(z-1+\alpha)^3}$ with $\alpha = 0.2\pi$. Recall that $\delta = \frac{T_f}{N_{sim}}$ is the sampling period. The effect of this filter corresponds approximately to have three poles in the frequency $\alpha/\delta = \frac{0.2\pi N_{sim}}{T_f}$. The RIGA normally furnishes a control pulse with “small ”bandwidth. To produce a seed with a “large” bandwidth, one has chosen the parameter T of RIGA (T is the fundamental frequency of the seed of RIGA [19]) twenty times lower than the chosen one in all other simulations. This will produce a seed for the RIGA which is a sum of $M = 11$ harmonics of the angular frequencies $\frac{2\pi}{T}$, and these frequencies appear in the seed $\overline{u}$ of Algorithm 4.2 (see the FFT of $\overline{u}$ in Figure 6, with the legend “RIGA”). The value of N_{sim} in this numerical experiment is $N_{sim} = 40N = 160$. To avoid an undesirable effects at the endpoints of the interval $[0, T_f]$ since the filtering process do not preserve the null control at $t = T_f$, the Hamming-like window function (E.2) was used for modulating the gain K of the FPA in the closed loop simulation. The final result is shown in Figure 6 with the legend “filter+FPA”. In fact, all the high-frequency components of the control pulses are almost completely eliminated after $N_{times} = 10$ iterations of Algorithm 3, respecting the desired infidelity of 0.001.

5. Convergence properties of the FPA

The idea behind the construction of the right translation R of our main convergence result is illustrated in a very simple example. For this consider, the scalar system:

$$\dot{x}(t) = u(t) \tag{5.1}$$

where $x(t), u(t) \in \mathbb{R}$. One will consider the tracking problem of a smooth trajectory $\overline{x}(t)$ that is obtained with $\overline{x}(0) = \overline{x}_0$ and a given smooth reference input $\overline{u}(t)$. The standard method is to consider the error $\widetilde{x}(t) = x(t) - \overline{x}(t)$. So, the corresponding error dynamics is given by

$$\dot{\widetilde{x}}(t) = \widetilde{u}(t)$$

where $\widetilde{u}(t) = u(t) - \overline{u}(t)$. Choose some $\alpha > 0$. Taking $\widetilde{u}(t) = \widetilde{U}(\widetilde{x}(t)) = -\alpha\widetilde{x}(t)$ one will get

$$\dot{\widetilde{x}}(t) = -\alpha\widetilde{x}(t) \tag{5.2}$$

and hence

$$\widetilde{x}(t) = \mathcal{F}(t, \widetilde{x}_0) = \exp(-\alpha t)\widetilde{x}_0$$

which assures exponential convergence for the tracking problem. Now, assume that one desires a finite time steering, namely $\widetilde{x}(t) = 0$ at $t = T_f$. For this, fix some $r \in \mathbb{R}$. Consider that r-translated state $\widetilde{x}_1(t) = \widetilde{x}(t) + r$ and apply the control law

$$\widetilde{u}(t) = \widetilde{U}(\widetilde{x}_1(t)) = -\alpha\widetilde{x}_1(t)$$

to system (5.2). Note that the closed loop dynamics may be rewritten in the following form

$$\dot{\widetilde{x}}_1(t) = -\alpha\widetilde{x}_1(t)$$

In particular (5.2) is invariant by r-translation, and so

$$\widetilde{x}_1(t) = \mathcal{F}(t, \widetilde{x}_0 + r)$$

Assume that r is such that

$$\mathcal{F}(\widetilde{x}_0 + r, T_f) = r$$

This means that $x_1(T_f) = r$, and so $\widetilde{x}(T_f) = 0$. In other words, applying $\widetilde{u}(t) = \widetilde{U}(\widetilde{x}_1(t))$ to (5.1), this control law will steer this system from $x(0) = \widetilde{x}_0$ to $x(T_f) = 0$. Clearly, for a fixed $\widetilde{x}_0$ and T_f, the map $\mathcal{G} : \mathbb{R} \to \mathbb{R}$ defined by

$$\mathcal{G}(r) = F(\widetilde{x}_0 + r, T_f) = (x_0 + r)\exp(-\alpha T_f)$$

is a contraction. This follows easily from the fact that $|\frac{\partial \mathcal{G}}{\partial r}| = |\exp(-\alpha T_f)| < 1$. In particular the Banach Fixed-Point Theorem always assures the existence of (a unique) fixed point r. Figure 7 illustrates this construction by considering $x_0 = 1$, $T_f = 1$, and $\alpha = 1$.

This simple idea will be directly translated to the more complex closed loop dynamics (2.4)–(2.5). For this, fix (a finite) $c > 0$ and consider the "closed ball"[17] in $\mathrm{U}(n)$ with radius c:

$$\overline{B}_c^{\mathcal{V}}(I) = \{X \in \mathrm{U}(n) \mid \mathrm{dist}(X, I) \leq c\}, \tag{5.3}$$

[17] It is shown in [25] that $\overline{B}_c^{\mathcal{V}}(I)$ is indeed a compact set in $\mathrm{U}(n)$.

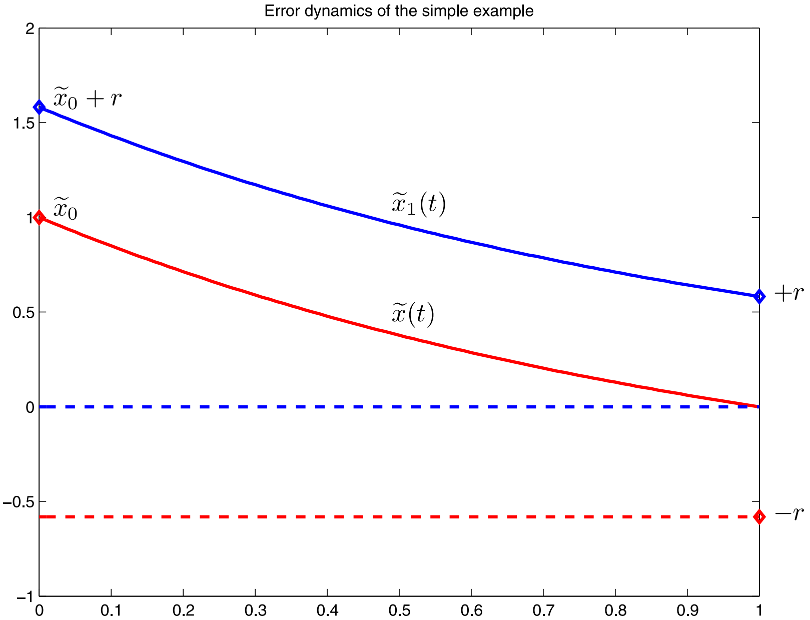

FIGURE 7. This figure illustrates the trick for constructing a finite time control law for steering the system from $\widetilde{x}(0) = \widetilde{x}_0$ to $\widetilde{x}(T_f) = 0$. Note that the trajectory $\widetilde{x}_1(t) = \mathcal{F}(T_f, \widetilde{x}_0 + r)$ is the r-translation of trajectory $\widetilde{x}(t)$. Furthermore, r is a fixed-point of the map $r \mapsto \mathcal{G}(r) = \mathcal{F}(T_f, \widetilde{x}_0 + r)$.

where "dist" is the notion of distance on $\mathrm{U}(n)$ induced by the Lyapunov function $\mathcal{V}$ in (2.1) (see the beginning of Sect. 2.1). It is clear that $\overline{B}^{\mathcal{V}}_c(I) \subset \mathcal{W}$ since $c > 0$ is finite. In this work one will also consider the usual closed ball of radius $\delta > 0$ induced by the Frobenius norm:

$$\overline{B}_\delta(I) = \{X \in \mathrm{U}(n) \mid \|X - I\| \leq \delta\}. \tag{5.4}$$

The following definition is instrumental in the context of the work [19]:

Definition 5.1. A reference trajectory $\overline{X} : [0, T_f] \to \mathrm{U}(n)$ generated by a reference input $\overline{u} : [0, T_f] \to \mathbb{R}^m$ is said to be λ-attractive on $\overline{B}^{\mathcal{V}}_c(I)$, where $0 < \lambda < 1$ and $c > 0$, when the closed-loop system (2.3)–(2.4)–(2.5) satisfies[18]

$$\mathrm{dist}(\widetilde{X}(T_f), I) \leq \lambda\, \mathrm{dist}(\widetilde{X}(0), I), \quad \text{for all } \widetilde{X}(0) \in \overline{B}^{\mathcal{V}}_c(I). \tag{5.5}$$

It is only the reference input that determines whether a reference trajectory is λ-attractive or not. The following properties stated in the next proposition shows the right-invariance of λ-attractive trajectories.

Proposition 5.2. *[19] If a reference trajectory $\overline{X}(t)$ that is generated by a solution of (2.3) with reference inputs $\overline{u} : [0, T_f] \to \mathbb{R}^m$ and initial condition $\overline{X}(0) = \overline{X}_0 \in U(n)$ is λ-attractive in $\overline{B}^{\mathcal{V}}_c(I)$, then for every $\overline{X}(0) \in U(n)$ and every $r > 0$, the corresponding solution is also λ-attractive in $\overline{B}^{\mathcal{V}}_r(I)$ (with the same $\lambda \in (0,1)$). Furthermore, this is equivalent to say that this property is invariant by right-translation of the reference trajectory $\overline{X}(t)$.*

[18] Since $\mathcal{V}(\widetilde{X}(t))$ is non-decreasing and $\overline{B}^{\mathcal{V}}_c(I)$ is compact, then $\widetilde{X}(t)$ is well-defined and remains in $\overline{B}^{\mathcal{V}}_c(I) \subset \mathcal{W}$ for $t \in [0, T_f]$.

The following proposition is proved in [19] and it shows that λ-attraction occurs with probability one with respect to the choice of the jet of the reference input. Before stating this result[19], we shall define some notations in the context of jet bundles [22].

Definition 5.3. Given a smooth map $\overline{u} : [0, T_f] \subset \mathbb{R} \to \mathbb{R}^m$, the M-degree polynomial approximation of $\overline{u}$ around x_0 given by its Taylor expansion at $\overline{t}$ is denoted by $\pi_{\overline{u}}$. Two maps $\overline{u} : [0, T_f] \to \mathbb{R}^m$ and $\overline{v} : [0, T_f] \to \mathbb{R}^m$ are M-equivalent if their M-degree polynomial Taylor approximations at $\overline{t}$ coincides, that is, $\pi_{\overline{u}} = \pi_{\overline{v}}$. This equivalence relation induces the set $J_{\overline{t}}(\mathbb{R}^m, [0, T_f])$ of all equivalent classes of smooth maps in $C^\infty([0, T_f], \mathbb{R}^m)$. The equivalent class corresponding to a representative $\overline{u}$ will be denoted by $[\overline{u}]$ and is called the M-jet of $\overline{u}$.

As the set $\mathbb{R}^M[t]$ of polynomials of degree M in the variable t may be identified with $\mathbb{R}^{M+1}$ (by choosing some ordering of their real coefficients), then the M-jet $[\overline{u}]$ may be canonically identified with a vector in $\left(\mathbb{R}^{M+1}\right)^m$.

Proposition 5.4. *[19] Fix $r > 0$ and choose $\overline{t} \in [0, T_f]$. One says that $[\overline{u}] \in (\mathbb{R}^{M+1})^m$ is an admissible M-jet if every smooth map $\overline{v} : [0, T_f] \to \mathbb{R}^m$ inside the equivalent class $[\overline{u}]$ generates a λ-attractive reference trajectory in $\overline{B}_r^{\mathcal{V}}(I)$ for some $\lambda \in (0, 1)$ (λ may depend on $\overline{v}$). Then, there exist $M \in \mathbb{N}$ large enough such that, a randomly chosen M-jet $[\overline{u}] \in (\mathbb{R}^{M+1})^m$ is admissible with probability one.*

The key ingredient of the proof of the convergence of the FPA is the Banach Fixed-Point Theorem applied to the flow of a closed-loop system. It is assumed that a seed is given. A reference trajectory $\overline{X}$ is then defined in the context of the next proposition:

Proposition 5.5. *Let $X : [0, T_f] \to U(n)$ be the solution of (1.1) that is obtained with $X(0) = I$ and the application of a δ-approximated seed $\overline{u}$ (see Def. 2.1). Then, by right-translation, one may construct a reference trajectory $\overline{X} : [0, T_f] \to U(n)$ with $\overline{X}(0) = \overline{X}_0 = X(T_f)^\dagger X_{goal}$ that is generated by the input $\overline{u}(t) = u(t)$. By construction, $\overline{X}(T_f) = X_{goal}$ and $\|\overline{X}_0 - I\| \leq \delta$.*

Proof. By right-invariance of (1.1), since $\overline{X}_0 = X_0 R_1$, with $R_1 = X(T_f)^\dagger X_{goal}$, then $\overline{X}(T_f) = X(T_f) R_1 = X_{goal}$. Since the Frobenius norm is right- and left- invariant by multiplications by unitary matrices, $\|\overline{X}_0 - I\| = \|X(T_f)^\dagger X_{goal} - I\| = \|X(T_f)[X(T_f)^\dagger X_{goal} - I]\| = \|X_{goal} - X(T_f)\| \leq \delta$. □

Fix $T_f > 0$ and consider a reference trajectory $\overline{X} : [0, T_f] \to \mathrm{U}(n)$ with $\overline{X}(T_f) = X_{goal}$, as in the context of the last proposition. Now, fix $R \in \mathrm{U}(n)$, and consider system (2.3)–(2.4) in closed-loop with the R-corrected feedback

$$\widetilde{u}_{1k}(t) = \widetilde{U}_k(\overline{X}(t), \widetilde{X}(t)R) = \mathrm{Tr}\left[Z\left(\widetilde{X}(t)R\right)\widetilde{S}_k(t)\right], \tag{5.6}$$

where $\widetilde{S}_k(t)$ is defined by (2.5c). Note that[20], by restricting to the initial instant $t_0 = 0$, the (time-varying) closed-loop system (2.4)–(5.6) admits a well-defined flow $\mathcal{F} : [0, T_f] \times \mathcal{W} \to \mathrm{U}(n)$ for which $\widetilde{X}(t) = \mathcal{F}(t, \widetilde{X}_0)$ [25]. Take the right-translation

$$\widetilde{X}_1(t) = \widetilde{X}(t)R. \tag{5.7}$$

By right-invariance of system (2.4), the dynamics of $\widetilde{X}_1(t)$ is time-varying and given by:

$$\dot{\widetilde{X}}_1(t) = \sum_{k=1}^{m} \widetilde{u}_{1k}(t)\widetilde{S}_k(t)\widetilde{X}_1(t), \quad \widetilde{X}_1(0) = \widetilde{X}_0 R \tag{5.8a}$$

$$\widetilde{u}_{1k}(t) = f_k \mathrm{Tr}\left[Z\left(\widetilde{X}_1(t)\right)\widetilde{S}_k(t)\right], \tag{5.8b}$$

[19] The original statement of [19] is different. It has been modified here for the sake of conciseness.

[20] When system (2.3) is disregarded from the closed-loop system (2.3)–(2.4)–(5.6), it turns out that the “reduced” closed-loop system (2.4)–(5.6) is time-varying.

that is, such system corresponds to same equations of the original closed-loop system (2.4)–(2.5), but with $\widetilde{X}$ replaced by $\widetilde{X}_1$ and with the right-translated initial condition $\widetilde{X}_1(0) = \widetilde{X}_0 R$. Since $\widetilde{X}(t) = \mathcal{F}(t, \widetilde{X}_0)$, it is immediate to show that $\widetilde{X}_1(t) = \mathcal{F}(t, \widetilde{X}_0 R)$. In particular, by assuming that $R \in \mathrm{U}(n)$ is a fixed point of the map[21]

$$\mathcal{G}_{T_f,\widetilde{X}_0}(R) = \mathcal{F}(T_f, \widetilde{X}_0 R), \tag{5.9}$$

that is, $\mathcal{G}_{T_f,\widetilde{X}_0}(R) = R$, then it follows that $\widetilde{X}_1(T_f) = R$. By (5.7) one has $\widetilde{X}(T_f) = I$, which in turn implies $X(T_f) = \overline{X}(T_f)\widetilde{X}(T_f) = X_{goal}$. Then it is easy to see that the step ℓ of Algorithm 1 corresponds to the numerical integration of system (5.8) with $\widetilde{X}(0) = \widetilde{X}_0 R^{\ell-1}$. In other words:

Proposition 5.6. *The construction of $R_\ell \in U(n)$ of the Algorithm 1 corresponds to the iteration $R_\ell = \mathcal{G}_{T_f,\widetilde{X}_0}(R_{\ell-1})$.*

Definition 5.7. Fix $T_f > 0$ and consider a reference trajectory $\overline{X} : [0, T_f] \to \mathrm{U}(n)$ which is the solution of (2.3) for arbitrarily fixed reference inputs. One says that the flow $\mathcal{F}$ (restricted to $t_0 = 0$) of the (time-varying) closed-loop system (2.4)–(5.6) is a local radial contraction (with respect to the Frobenius norm) if there exist $0 < \gamma < 1$ and $r > 0$ such that[22]

$$\|\mathcal{F}(T_f, \widetilde{X}_0) - I\| \leq \gamma \|\widetilde{X}_0 - I\|, \quad \text{for all } \widetilde{X}_0 \in \overline{B}^{\mathcal{V}}_c(I). \tag{5.10}$$

Condition (5.10) above is (locally) ensured by the next proposition if the reference trajectory is λ-attractive.

Theorem 5.8. *Let $\overline{X} : [t_0, T_f] \to U(n)$ be a given reference trajectory (solution of (2.3)). Assume that $\overline{X}$ is λ-attractive in $\overline{B}^{\mathcal{V}}_r(I)$ in the sense of Definition 5.1. Hence for every γ such that $\gamma^2 \in (\lambda, 1)$, there exists $c > 0$ small enough such that (5.10) holds for all $\widetilde{X}_0 \in \overline{B}^{\mathcal{V}}_c(I)$.*

Proof. See Appendix C. □

The existence of a fixed point R of $\mathcal{G}_{T_f,\widetilde{X}_0}$ in (5.9) is (locally) assured by the result below, whose proof is deferred to Appendix A:

Theorem 5.9. *Fix $T_f > 0$. Consider a reference trajectory $\overline{X} : [0, T_f] \to U(n)$ which is the solution of (2.3) for given reference inputs such that (5.10) holds for some $0 < \gamma < 1$ and $c > 0$. Fix any $\gamma < \lambda < 1$. Then, there exist $\epsilon_1, \epsilon_2 > 0$ such that, for every fixed $\widetilde{X}_0 \in B_{\epsilon_1}(I)$, one has that the map $\mathcal{G}_{T_f,\widetilde{X}_0} : \overline{B}_{\epsilon_2}(I) \to \overline{B}_{\epsilon_2}(I)$ defined by (5.9) is well-defined and is a λ-contraction. In particular, for every fixed $\widetilde{X}_0 \in B_{\epsilon_1}$, the Banach Fixed-Point Theorem implies that the map $\mathcal{G}_{T_f,\widetilde{X}_0}$ admits a unique fixed point R in $\overline{B}_{\epsilon_2}(I)$.*

Definition 5.10. When a fixed point $R \in \mathrm{U}(n)$ of $\mathcal{G}_{T_f,\widetilde{X}_0}$ exists, then it will be called an exact correction. If $R \in \mathrm{U}(n)$ is such that $\|R_1 - R\| \leq \epsilon$ (Frobenius norm) with $R_1 = \mathcal{G}_{T_f,\widetilde{X}_0}(R)$, then R will be called an ϵ-approximation of the exact correction.

Proposition 5.11. *Fix $T_f > 0$ and consider that the reference trajectory $\overline{X} : [0, T_f] \to U(n)$ is the solution of (2.3) for arbitrarily fixed reference inputs. Assume that $\overline{X}(0)$ is chosen so that $\overline{X}(T_f) = X_{goal}$. Let R be an ϵ-approximation of the correction. Then, the closed-loop system (2.4) with the R-corrected feedback (5.6) will produce a final state $X(T_f)$ such that $\|X(T_f) - X_{goal}\| \leq \epsilon$. Now let $R_1 = \mathcal{G}_{T_f,\widetilde{X}_0}(R)$. Then the infidelity $\mathcal{I}(X^\dagger_{goal} X(T_f)) = \mathcal{I}(R_1 R^\dagger) = \mathcal{I}(R^\dagger R_1)$.*

[21] At this point, we are not being completely precise, since the domain of the map $\mathcal{G}_{T_f,\widetilde{X}_0}$ is not defined.

[22] Recall that the “closed ball” $\overline{B}^{\mathcal{V}}_c(I)$ is defined by (5.3) based on the notion of distance induced by the Lyapunov function $\mathcal{V}$, whereas the usual closed ball induced by the Frobenius norm is given in (5.4).

Proof. Let $\widetilde{X}_1(t) = \widetilde{X}(t)R$, and recall that $R_1 \triangleq \mathcal{G}_{T_f,\widetilde{X}_0}(R) = \widetilde{X}_1(T_f)$ and $\widetilde{X}(0) = \overline{X}^\dagger(0)$ (see Prop. 5.5). In particular, since the Frobenius norm is invariant by left- and right-multiplication by a unitary matrix, one has $\|\widetilde{X}(T_f) - I\| = \|\widetilde{X}_1(T_f)R^\dagger - I\| = \|R_1R^\dagger - I\| = \|(R_1 - R)R^\dagger\| = \|R_1 - R\| \leq \epsilon$. Similarly, $\|X(T_f) - X_{goal}\| = \|\overline{X}^\dagger(T_f)[X(T_f) - X_{goal}]\|$. Since $\overline{X}(T_f) = X_{goal}$, one gets $\|X(T_f) - X_{goal}\| = \|\widetilde{X}(T_f) - I\| \leq \epsilon$. This means that the final error will be less than or equal to ϵ. Now, by the same reasoning, it follows that $X_{goal}^\dagger X(T_f) = \overline{X}^\dagger(T_f)X(T_f) = \widetilde{X}(T_f) = \widetilde{X}_1(T_f)R^\dagger = R_1R^\dagger$. □

For a fixed $\widetilde{X}_0$, Algorithm 1 of the previous section is a numerical method for determining an ϵ-approximation of the exact correction, whose existence is ensured by Theorem 5.9. The well-known proof of the Banach Fixed-Point Theorem finds the unique fixed-point R of a contraction $\mathcal{G}$ by taking the limit of the sequence R_ℓ as $\ell \to \infty$, with $R_{\ell+1} = \mathcal{G}(R_\ell)$. Note that Algorithm 1 is essentially the computation of R_ℓ in this way. Furthermore, if $\mathcal{G}$ is a ρ-contraction, then $\epsilon_{\ell+1} = \|R_{\ell+1} - R_\ell\| = \|\mathcal{G}(R_\ell) - \mathcal{G}(R_{\ell-1})\| \leq \rho\|R_\ell - R_{\ell-1}\|$. In particular, $\epsilon_\ell \leq \rho^\ell\epsilon_0$. In particular, from Proposition 5.11, the convergence of the FPA will assure an exponential convergence of the final error of the quantum gate.

6. Conclusions

This paper presented an iterative Lyapunov-based algorithm for improving the precision of quantum gate generation for systems with drift in a given fixed finite time T_f in the context of coherent (open-loop) control. When FPA converges, this convergence is exponential, and so the fidelity can be exponentially improved by FPA. The FPA will converge only when its seed (the original approximated solution) is already a good approximation of the quantum gate generation and it is also a λ-attractive trajectory in the sense of Definition 5.1. When the FPA does not converge, which is detected when the contraction test of Algorithm 1 fails, the Algorithm 2 is still able to reduce the infidelity to desired one. This situation was found in the examples of this work. The FPA may be combined with any technique for generating its seed, for instance, the Krotov method, [23], GRAPE [8], CRAB [20] or RIGA [19] in the piecewise-constant setting, and with GOAT [10] or RIGA in the smooth setting. An interesting feature is that FPA promotes only very small corrections in the inputs that were previously computed for its seed (see Figs. 1 and 2 respectively for the piecewise-constant and the smooth cases). It is important to stress that the RIGA [19] is already a method that produces exponential convergence of the solution[23]. Normally, RIGA has a convergence rate that is compatible with the one of FPA (see Rem. 4.1). However, when restrictions on the inputs are included (for instance the input format of Figs. 1 and 2, beginning and finishing with zero amplitude) this makes the convergence of RIGA (alone) slower than the combination RIGA-FPA (see Tabs. 1 and 2 respectively for the piecewise-constant and the smooth cases). Although the smooth version of the FPA is meant to be used with a continuous seed, one may start with a piecewise-constant set of control pulses and use the FPA to convert it to a smooth set of control pulses with the desired fidelity. An important feature of both RIGA and FPA is the presence of the simultaneous qualities: (a) they can consider high dimensional systems in implementations running fast in affordable personal computers; (b) they can generate smooth control pulses with small bandwidth; (c) they are rather flexible to be applied easily to any quantum system of the form (1.1). The FPA may be used also for shortening the bandwidth of control pulses, preserving or improving fidelity, as suggested by the numerical experiments regarding Algorithm 4.3.

Appendix A. Proof of Theorem 5.9

Along this proof, one will consider $U(n) \subset \mathbb{C}^{n\times n}$ as a real manifold. For this one will consider the standard projection $\pi : \mathbb{C} \to \mathbb{R}^2$ such that $a + \imath b \mapsto (a, b)$. Analogously, the inverse map $\varpi : \mathbb{R}^2 \to \mathbb{C}^2$ is defined by $(a, b) \mapsto a + \imath b$. Given a complex matrix $A + \imath B \in \mathbb{C}^{n\times n}$, such projection may be extended to a map $A + \imath B \mapsto (A, B) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$. In this context, one shall identify $\mathbb{R}^{2n^2} \cong \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \cong \mathbb{C}^{n\times n}$. Note that the Euclidian

[23] At least when T_f is large enough.

norm on $\mathbb{R}^{2n^2}$ induces the Frobenius norm on $\mathbb{C}^{n\times n}$. An open ball of $C^{n\times n}$ with center X and radius $\delta > 0$ with the Frobenius norm will be denoted by $B^F_\delta(X)$ (and will be identified with the corresponding ball of $\mathbb{R}^{2n^2}$ with the Euclidean norm). The ball $B_\delta(X) \subset \mathrm{U}(n)$ stands for $B^F_\delta(X) \cap \mathrm{U}(n)$. Closed balls will be denoted as usual, for instance $\overline{B}_\delta(X)$. One will consider the restriction $\mathcal{F}(t, \widetilde{X}_0)$ to the initial instant $t_0 = 0$ of the flow of the time-varying system (2.4)–(2.5) as a map $\mathcal{F} : [0, T_f] \times \mathcal{W} \to \mathbb{R}^{2n^2}$. One will denote the singular-value norm of a linear operator $L : \mathbb{R}^p \to \mathbb{R}^q$ by $\|L\|_\sigma$. The proof of Theorem 5.9 relies on the result below, which in turn is proved in Appendix B.

Lemma A.1. *Under the assumptions of Theorem 5.9, if one chooses any $\gamma < \lambda < 1$, then there exists $\epsilon > 0$ such that (Frobenius norm):*

$$\|F(x_1) - F(x_2)\| \leq \lambda \|x_1 - x_2\|, \quad \textit{for all } x_1, x_2 \in \overline{B}_\epsilon(I),$$

where $F : \mathcal{W} \subset U(n) \to \mathbb{R}^{2n^2}$ is defined by $F(\widetilde{X}) = \mathcal{F}(T_f, \widetilde{X})$.

Now, for showing Theorem 5.9, fix $\gamma < \lambda < 1$, and take $\epsilon = \epsilon(\lambda) > 0$ as in Lemma A.1 above. As $\mathcal{G}_{T_f, \widetilde{X}_0}(R) = F(\widetilde{X}_0 R)$, now one will construct $\epsilon_1 > 0$ and $\epsilon_2 > 0$ such that:

(a) $\widetilde{X}_0 R \in \overline{B}_\epsilon(I)$ for all $R \in \overline{B}_{\epsilon_2}(I)$ and $\widetilde{X}_0 \in \overline{B}_{\epsilon_1}(I)$;
(b) $F(\widetilde{X}_0 R) \subset \overline{B}_{\epsilon_2}(I)$ for all $R \in \overline{B}_{\epsilon_2}(I)$ and $\widetilde{X}_0 \in \overline{B}_{\epsilon_1}(I)$.

In order to construct ϵ_1 and ϵ_2 assuring the properties (a) and (b), let $\xi_R = R - I$ and $\xi_X = \widetilde{X}_0 - I$. Note that $\|\widetilde{X}_0 R - I\| = \|\xi_X + \xi_R + \xi_X \xi_R\| \leq \|\xi_X\| + \|\xi_R\| + \|\xi_X\|\|\xi_R\| \leq \epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2$. Let $\epsilon_1 = b\epsilon_2$, where $b \in (0, 1)$ will be chosen afterwards. Then, condition (a) is implied by following condition:

$$\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2 = \epsilon_2[1 + b + b\epsilon_2] < \epsilon,$$

which in turn is equivalent to $\epsilon_2 < \frac{\epsilon}{1+b+b\epsilon_2}$. Assuming that $\epsilon_2 < 1$ and $b < 1$, then in order to ensure that $\widetilde{X}_0 R \in B_\epsilon(I)$, it suffices to choose $\epsilon_2 \leq \min\{\frac{\epsilon}{3}, 1\}$ Now, to assure that $F(\widetilde{X}_0 R) \subset \overline{B}_{\epsilon_2}(I)$, note by Lemma A.1 (applied with $x_2 = I$ and $x_1 = \widetilde{X}_0 R$) that $\|F(\widetilde{X}_0 R) - I\| \leq \lambda \|\widetilde{X}_0 R - I\|$. Hence,

$$\|F(\widetilde{X}_0 R) - I\| \leq \lambda\epsilon_2[1 + b + b\epsilon_2].$$

It is thus clear that if one chooses ϵ_2 such that

$$\lambda\epsilon_2[1 + b + b\epsilon_2] \leq \epsilon_2, \tag{A.1}$$

then it will be true that $F(\widetilde{X}_0 R) \subset \overline{B}_{\epsilon_2}(I)$. Simple manipulations show that (A.1) is equivalent to have:

$$b \leq (1 - \lambda)/[\lambda(1 + \epsilon_2)] = \theta.$$

As $\lambda \in (\gamma, 1)$, one may choose $b = \min\{1, \theta\}$. This shows that one may construct ϵ_1 and ϵ_2 such that (a) and (b) hold. Now, recall that $\mathcal{G}_{T_f, \widetilde{X}_0}(R) = F(\widetilde{X}_0(R))$. Given $\widetilde{X}_0 \in \mathrm{U}(n)$ and $W \in C^{n\times n}$, note that $\|\widetilde{X}_0 W\| = \mathrm{trace}(W^\dagger \widetilde{X}_0^\dagger \widetilde{X}_0 W) = \mathrm{trace}(W^\dagger W) = \|W\|$. Hence, to complete the proof of Theorem 5.9, note that $\|F(\widetilde{X}_0 R_1) - F(\widetilde{X}_0 R_2)\| \leq \lambda \|\widetilde{X}_0 R_1 - \widetilde{X}_0 R_2\| = \lambda \|\widetilde{X}_0(R_1 - R_2)\| = \lambda \|R_1 - R_2\|$.

Appendix B. Proof of Lemma A.1

The present considers the same notations and the identification $\mathbb{R}^{2n^2} \cong \mathbb{C}^{n\times n}$ described in the beginning of Appendix A. The proof of Lemma A.1 is based on the following mathematical analysis results:

Lemma B.1. *Let $F : U \subset \mathbb{R}^p \to \mathbb{R}^q$ be a smooth map, where $U \subset \mathbb{R}^p$ is an open subset. Let $x_0 \in U$ and assume that $\left\| \frac{\partial F}{\partial x}\big|_{x_0} \right\|_\sigma = \gamma < \infty$. Fix any $\lambda \in (\gamma, \infty)$. Then, there exists $\delta > 0$ such that, for every $x_1, x_2 \in B_\delta(x_0) \subset U$, one has $\|F(x_1) - F(x_2)\| \leq \lambda \|x_1 - x_2\|$.*

Proof. The proof is left to the reader. It is in fact an easy consequence of the convexity of a ball, the continuity of the singular-value norm, and the mean value theorem. □

Lemma B.2. *Let $H : U \subset \mathbb{R}^p \to \mathbb{R}^q$ be a smooth map, where $U \subset \mathbb{R}^p$ is an open set, and let $u_0 \in U$. Assume that there exist $\epsilon, \lambda > 0$ such that $B_\epsilon(u_0) \subset U$ and $\|H(u) - H(u_0)\| \leq \lambda \|u - u_0\|$, for all $u \in B_\epsilon(u_0)$. Then $\left\| \frac{\partial H}{\partial u}|_{u_0} \right\|_\sigma \leq \lambda$.*

Proof. Note that the directional derivative $\frac{\partial H}{\partial u}|_{u_0} \cdot h$ is given by $\lim_{t \to 0} \frac{H(th+u_0)-H(u_0)}{t}$. For $|t|$ small enough, one gets $\| \frac{H(th+u_0)-H(u_0)}{t} \| \leq |\lambda| \|h\|$, and so the result easily follows since the singular-value norm of a matrix is its greater amplification factor. □

Assume that $\exp : \mathbb{C}^{n\times n} \to \mathbb{C}^{n \times n}$ stands for the exponential map and $\log : B_1^F(I) \subset \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ stands for the logarithm map, which is well-defined and smooth inside the ball centered at the identity matrix and with unit radius with respect to the Frobenius norm. One will abuse notation and consider the exponential as a map from $\mathbb{R}^{2n^2}$ to $\mathbb{R}^{2n^2}$, and the logarithm as a map from $B_1^F(I) \subset \mathbb{R}^{2n^2}$ to $\mathbb{R}^{2n^2}$. As $\exp(\sigma) = I + \sigma + \mathcal{O}(\sigma)$, it is easy to show that $\frac{\partial \exp}{\partial u}\Big|_0 = I$, the $2n^2 \times 2n^2$ identity matrix. By the inverse mapping theorem, it is clear that $\frac{\partial \log}{\partial x}\Big|_I = I$. As $\|I\|_\sigma = 1$, a double application of Lemma B.1 allows one to show the following proposition:

Proposition B.3. *Fix any $a \in (1, \infty)$. Then, there exists $\delta > 0$ such that, for the ball $B_\delta^F(I) \subset \mathbb{R}^{2n^2}$ and the open set $U = \log(B_\delta^F(I))$ (containing the null matrix), one has*

$$\|\exp(u_1) - \exp(u_2)\| \leq a\|u_1 - u_2\|, \forall u_1, u_2 \in U \subset \mathbb{R}^{2n^2}, \tag{B.1a}$$

$$\|\log(x_1) - \log(x_2)\| \leq a\|x_1 - x_2\|, \forall x_1, x_2 \in B_\delta^F(I) \subset \mathbb{R}^{2n^2}. \tag{B.1b}$$

Proof. First apply Lemma B.1 to the maps exp and log, obtaining inequalities (B.1a) and (B.1b) on the open balls $B_{\delta_1}^F(0)$ and $B_{\delta_2}^F(I)$, respectively. Then, take $\delta > 0$ small enough so that $B_\delta^F(I) \subset B_{\delta_2}^F(I) \cap \log^{-1}(B_{\delta_1}^F(0))$. By construction, (B.1b) holds on $B_\delta^F(I)$, and (B.1a) holds on $U = \log\left(B_{\delta_1}^F(I)\right)$. □

Proof. (of Lem. A.1) Fix $\lambda \in (\gamma, 1)$. Let $a_1 = \sqrt{\gamma/\lambda}$. Fix any $1 < a < a_1$ and apply Proposition B.3 for such value of $a > 1$. Now, note that[24] one can always take a smaller value of $\delta > 0$ in Proposition B.3 in a way that $B_\delta(I) \subset \overline{B}_c^{\mathcal{V}} \subset \mathcal{W}$, where $\mathcal{W}$ is defined in (2.2). This follows from ([25], Ineq. (C1) in Appendix 3). Let $F : \mathcal{W} \subset \mathrm{U}(n) \to \mathbb{R}^{2n^2}$ be defined by $F(\widetilde{X}) = \mathcal{F}(T_f, \widetilde{X})$. From the assumptions of Theorem 5.9, one obtains

$$\|F(\widetilde{X}) - I\| \leq \gamma \|\widetilde{X} - I\|, \forall \widetilde{X} \in B_\delta(I) \subset \mathrm{U}(n)$$

In particular, as $\gamma < 1$, one has $F(B_\delta(I)) \subset B_\delta(I)$. Let $U = \log(B_\delta(I))$ and $\mathcal{U} = \mathfrak{u}(n) \cap U$. Clearly, $\exp(\mathcal{U}) \subset B_\delta^F(I) \cap \mathrm{U}(n) = B_\delta(I)$. Then one may define the map $H : \mathcal{U} \to B_\delta(I)$ by $H(u) = F \circ \exp(u)$. By definition,

$$\|H(u) - I\| = \|F(\exp(u)) - I\| \leq \gamma \|\exp(u) - \exp(0)\| \leq \gamma a \|u - 0\|, \forall u \in U.$$

From Lemma B.2, one has $\| \frac{\partial H}{\partial u}\big|_0 \|_\sigma \leq \gamma a$. Let $\lambda_1 = a_1 \gamma > \gamma a$. By Lemma B.1, there exists an open neighborhood $\mathcal{V} \subset \mathcal{U}$ of the null matrix such that

$$\|H(u_1) - H(u_2)\| \leq \lambda_1 \|u_1 - u_2\|, \forall u_1, u_2 \in \mathcal{V}.$$

[24] Once again one is abusing notation through the identification $\mathbb{R}^{2n^2} \cong \mathbb{C}^{n\times n}$.

Since $\mathcal{V} \subset \mathfrak{u}(n)$ is an open set containing the null matrix, it follows that $\exp(\mathcal{V})$ is an open set of $\mathrm{U}(n)$ containing I. Choose $\epsilon > 0$ small enough such that $B_\epsilon(I) \subset \exp(\mathcal{V}) \subset \exp(\mathcal{U}) \subset B_\delta(I)$. Let $x_1, x_2 \in B_\epsilon(I)$, and $u_1 = \log(x_1)$ and $u_2 = \log(x_2)$. Then, by Proposition B.3, one gets:

$$\|F(x_1) - F(x_2)\| = \|H(u_1) - H(u_2)\| \leq \lambda_1 \|\log(x_1) - \log(x_2)\| \leq aa_1\gamma\|x_1 - x_2\|, \forall x_1, x_2 \in B_\epsilon(I).$$

Note that $aa_1\gamma \leq a_1^2\gamma = \lambda$. This shows Lemma A.1. □

Appendix C. Attractiveness and local radial contractions

The proof of Theorem 5.8 is based on the following lemma.

Lemma C.1. *Fix $\epsilon > 0$. Let $X \in U(n)$ and let $X = U^\dagger diag(\exp(\iota\theta_1), \ldots, \exp(\iota\theta_n))U$ be its eigenstructure, where $U \in U(n)$, and $\theta_i \in (-\pi, \pi], i = 1, \ldots, n$. Let $H : U(n) \to \mathbb{R}$ be defined by $H(X) = \sum_{i=1}^n \theta_i^2$ and $G : U(n) \to \mathbb{R}$ be defined by $G(X) = \|X - I\|^2$. Then, there exists $c_1 > 0$ such that, for every $X \in \overline{B}_{c_1}^{\mathcal{V}}(I)$, one has:*

$$\begin{aligned} &(a) \quad (1-\epsilon)\frac{H(X)}{4} \leq \mathcal{V}(X) \leq (1+\epsilon)\frac{H(X)}{4}; \\ &(b) \quad (1-\epsilon)H(X) \leq G(X) \leq (1+\epsilon)H(X). \end{aligned}$$

Proof. (of Lem. C.1) Recall that $\mathcal{V}(X) = \sum_{i=1}^n (\tan\frac{\theta_i}{2})^2$ [24]. Let $\mathcal{O}$ denote a function such that $\lim_{x\to 0} \frac{\mathcal{O}(x)}{|x|} = 0$. Since $\tan(x) = x + \mathcal{O}(x)$, it follows easily that $\tan^2(\theta/2) = \theta^2/4 + \mathcal{P}(\theta)$, where $\mathcal{P}(\theta)$ is such that $\lim_{\theta\to 0} \frac{\mathcal{P}(\theta)}{(\theta^2/4)} = 0$. In particular,

$$\mathcal{V}(X) = \sum_{i=1}^n \frac{\theta_i^2}{4}\left(1 + \frac{\mathcal{P}(\theta_i)}{\left(\frac{\theta_i^2}{4}\right)}\right).$$

Fix $\epsilon > 0$, and take r small enough such that $|\theta_i/2| \leq r$ implies that $\left|\frac{\mathcal{P}(\theta_i)}{\left(\frac{\theta_i^2}{4}\right)}\right| \leq \epsilon$. By the continuity of $\tan^2(\theta/2)$ and the fact that this function is an increasing map when restricted to the interval $(-\pi, \pi]$, then one take c_1 small enough such that $tan^2(\theta/2) < c_1$ implies that $|\theta/2| < r$. Now, if $\mathcal{V}(X) \leq c_1$ it is clear that $\left|\frac{\mathcal{P}(\theta_i)}{\left(\frac{\theta_i^2}{4}\right)}\right| \leq \epsilon$, and this shows (a). From ([25], Proof of Lem. C2) one has $G(X) = \sum_{i=1}^n (2\sin\frac{\theta_i}{2})^2$. Hence, in a similar fashion, one may show (b) (with a common c_1) by using the fact that $(2\sin\frac{\theta}{2})^2 = \theta^2 + \mathcal{Q}(\theta)$ where $\lim_{\theta\to 0}\frac{\mathcal{Q}(\theta)}{\theta^2} = 0$. □

Proof. (Of Thm. 5.8) One denotes here $\widetilde{X}_1 = \mathcal{F}(T_f, \widetilde{X}_0)$, where $\mathcal{F}$ is the flow of the closed-loop system (2.4) restricted to the initial instant $t_0 = 0$. Since $\mathcal{F}(T_f, I) = I$ one will assume that $\widetilde{X}_0 \neq I$, otherwise (5.10) is trivial. The assumption is equivalent to say that $\frac{\mathcal{V}(\widetilde{X}_1)}{\mathcal{V}(\widetilde{X}_0)} \leq \lambda$ for all $\widetilde{X}_0 \in \overline{B}_{c_1}^{\mathcal{V}}(I)$. Note that (5.10) is equivalent to say that $\frac{G(\widetilde{X}_1)}{G(\widetilde{X}_0)} \leq \gamma^2$ for all $\widetilde{X}_1 \in \overline{B}_{c_1}^{\mathcal{V}}(I)$. From the part (a) of the last lemma, it is easy to show that, for all $\epsilon > 0$ one may construct c_1 such that

$$\frac{(1-\epsilon)H(\widetilde{X}_1)}{(1+\epsilon)H(\widetilde{X}_0)} \leq \frac{\mathcal{V}(\widetilde{X}_1)}{\mathcal{V}(\widetilde{X}_0)} \leq \frac{(1+\epsilon)H(\widetilde{X}_1)}{(1-\epsilon)H(\widetilde{X}_0)}, \tag{C.1}$$

for all $\widetilde{X}_0 \in \overline{B}^{\mathcal{V}}_{c_1}(I)$. Using the part (b) of the last lemma, one may obtain a similar inequality that replaces $\frac{\mathcal{V}(\widetilde{X}_1)}{\mathcal{V}(\widetilde{X}_0)}$ by $\frac{G(\widetilde{X}_1)}{G(\widetilde{X}_0)}$. Note that, if x, y are real numbers in $[a, b]$, then $|x - y| \leq (b - a)$. Then, from these two inequalities, it is easy to verify that

$$\left| \frac{\mathcal{V}(\widetilde{X}_1)}{\mathcal{V}(\widetilde{X}_0)} - \frac{G(\widetilde{X}_1)}{G(\widetilde{X}_0)} \right| \leq \frac{(1+\epsilon)H(\widetilde{X}_1)}{(1-\epsilon)H(\widetilde{X}_0)} - \frac{(1-\epsilon)H(\widetilde{X}_1)}{(1+\epsilon)H(\widetilde{X}_0)} \leq \frac{4\epsilon H(\widetilde{X}_1)}{(1-\epsilon^2)H(\widetilde{X}_0)}.$$

In particular,

$$\frac{G(\widetilde{X}_1)}{G(\widetilde{X}_0)} \leq \frac{\mathcal{V}(\widetilde{X}_1)}{\mathcal{V}(\widetilde{X}_0)} + \frac{4\epsilon H(\widetilde{X}_1)}{(1-\epsilon^2)H(\widetilde{X}_0)}.$$

After using the left side of (C.1), one gets

$$\frac{G(\widetilde{X}_1)}{G(\widetilde{X}_0)} \leq \frac{\mathcal{V}(\widetilde{X}_1)}{\mathcal{V}(\widetilde{X}_0)} \left[1 + \frac{4\epsilon(1+\epsilon)}{(1-\epsilon^2)(1-\epsilon)} \right].$$

One concludes the proof by taking ϵ small enough such that $\left[1 + \frac{4\epsilon(1+\epsilon)}{(1-\epsilon^2)(1-\epsilon)}\right] < \gamma^2/\lambda$ (recall that $\gamma^2/\lambda > 1$ by assumption). So $\frac{G(\widetilde{X}_1)}{G(\widetilde{X}_0)} \leq \gamma^2$. □

Appendix D. Some properties of the Lyapunov function and the map Z

Proposition D.1. *Fix $c > 0$ and consider the ball $\overline{B}^{\mathcal{V}}_c(I)$ defined by (5.3). Then*
(a) There exists $k_1, k_2 > 0$ such that $k_1\|\widetilde{X} - I\|^2 \leq \mathcal{V}(\widetilde{X}) \leq k_2\|\widetilde{X} - I\|^2$, $\forall \widetilde{X} \in \overline{B}^{\mathcal{V}}_c(I)$.
(b) There exists $k_3, k_4 > 0$ such that $k_3\|\widetilde{X} - I\|^2 \leq \|Z(\widetilde{X})\|^2 \leq k_4\|\widetilde{X} - I\|^2$, $\forall \widetilde{X} \in \overline{B}^{\mathcal{V}}_c(I)$
(c) There exists $c_1 > 0$ such that $\|Z(\widetilde{X})\|^2 \leq c_1\mathcal{V}(\widetilde{X})$, $\forall \widetilde{X} \in \overline{B}^{\mathcal{V}}_c(I)$.

Proof. See ([19], Appendix 6). □

Appendix E. Smooth implementations of FPA

As said in Section 3.2, for the open loop simulation, each step of 4th-order Runge-Kutta in the interval $[t_s, t_{s+1}]$ relies on the right-invertibility of system (2.3). For fixed $s \in \{0, 1, \ldots, N_{sim}\}$, denote $W(t) = \mathfrak{W}(\overline{X}(t)X_s^{\dagger})$. The simulation of $W(t)$ in the interval $[t_s, t_{s+1}]$ using (3.6) will always consider the initial condition $W_0 = 0$ at $t = t_s$(corresponding to the identity in $\mathrm{U}(n)$). The final value of each step will be corrected by right-invertibility, accordingly. In other words, denoting $\overline{W}_{s+1} = W(t_{s+1})$, then $\overline{X}_{s+1} = \mathfrak{X}(\overline{W}_{s+1})\overline{X}_s$. Let $\tau \in [0, \delta]$, Define $S(t_s + \tau) = S_0 + \sum_{k=1}^{m} \overline{u}_k(t_s + \tau)S_k$, where S_k are the system matrices for $k = 0, 1, \ldots, m$ and $\overline{u}_k(t_s + \tau)$ is computed by linear interpolation as in (3.3). The open loop simulation reads:

```
% BEGIN Step 0 – Open-loop simulation
    $\overline{X}_0 = I$
    $W_0 = 0$
    FOR $s = 0, 1 \ldots N_{sim} - 1$
        % BEGIN RUNGE KUTTA 4th-order
        $k_1 = \delta F(W_0, S(t_s))$
        $k_2 = \delta F(W_0 + \frac{k1}{2}, S(t_s + \frac{\delta}{2}))$
        $k_3 = \delta F(W_0 + \frac{k2}{2}, S(t_s + \frac{\delta}{2}))$
        $k_4 = \delta F(W_0 + k3, S(t_s + \delta))$
        $\overline{W}_{s+1} = W_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
        % END RUNGE-KUTTA 4th-order
        $\overline{X}_{s+1} = \mathfrak{X}(\overline{W}_{s+1})\overline{X}_s$
    **END**
    $\overline{X}_f = \overline{X}_{N_{sim}}$
    **FOR** $s = 0, 1 \ldots N_{sim}$
        $\overline{X}_s = \overline{X}_s X_f^\dagger X_{goal}$
    **END**
**% END Step 0 – Open-loop simulation**
```

There are different approximations used in the open-loop and the closed loop simulations. For the open-loop, the smooth control $\overline{u}(t)$ is approximated by the linear interpolation (3.3), whereas the closed-loop integration considers that the feedback is smooth, but the values of $\overline{X}(t)$ in 2.5 are conveniently interpolated, as it will be stated. In order to avoid cumulative errors, the value of X_f will not be deduced from $\widetilde{X}(T_f)$, and that is the reason why X_f must be computed above[25] in each step 0 of the FPA.

The values of $\overline{W}_{s+1}$ that are previously obtained in the open loop simulation are used in the closed loop simulation, since the intermediary value of $\overline{X}(t_s + \tau)$ for $\tau = \delta/2$ is needed. The interpolation is done in the space $\mathfrak{u}(n)$ using these values of $\overline{W}_{s+1}$ in the following way:

$$\overline{X}(t_s + \frac{\delta}{2}) = \mathfrak{X}\left[\frac{1}{2}(W_0 + \overline{W}_{s+1})\right]\overline{X}_s = \mathfrak{X}\left[\frac{1}{2}\overline{W}_{s+1}\right]\overline{X}_s$$

Let $\widetilde{S}_k(t_s + \tau) = X(t_s + \tau)^\dagger S_k X(t_s + \tau)$. The computation of the feedback-law is given by

$$\widetilde{u}_k(t_s + \tau, W) = \text{Window}(t_s + \tau) K \text{trace}\left[Z_W(W)\widetilde{S}_k(t_s + \tau)\right] \tag{E.1}$$

which is done easily with $Z_W(W) = \frac{1}{4}W(W + I)(W - I)$ being the expression of (2.5d) as a function of W. Define $T(t_s + \tau, W) = \sum_{k=1}^m \widetilde{u}_k(t_s + \tau, W)\widetilde{S}_k(t_s + \tau)$. The function $\text{Window}(t_s + \tau)$ for $\tau \in [0, \delta]$ is obtained by linear interpolation from the values of $\text{Window}(t_s), s = 0, 1, \ldots, N_{sim}$ analogously to (3.3). One may take these values of $\text{Window}(t_s)$ all equal to one. When it is imperative that the values of control pulses will not be modified by FPA at $t = 0$ and $t = T_f$, one may take the same window function (3.5), but with $N_f = N_{sim}$ and for simplicity $\text{Window}(s)$ stands for $\text{Window}(t_s)$, for $s = 0, 1, \ldots, N_f$. Sometimes it is useful to use a Hamming-like window function given by (E.2) (that is also used in RIGA). It may let the convergence of FPA slower than the one with the unitary window but it produces a smaller bandwidth and it respects common restrictions of null control at the endpoints of the interval $[0, T_f]$.

$$\text{Window}(t) = \frac{1}{2}\left[1 - \cos\left(2\pi\frac{t}{T_f}\right)\right] \tag{E.2}$$

The closed-loop simulation reads:

```
**% BEGIN Step $\ell$ – Closed-loop simulation**
    $\widetilde{X}_0 = \overline{X}_0^\dagger$
    $W_0 = \mathfrak{W}(\widetilde{X}_0 R_{\ell-1})$
```

[25] This is not the case for the implementation of the piecewise-constant case.

```
    FOR $s = 0, 1 \dots N_{sim} - 1$
        % BEGIN RUNGE KUTTA 4th-order
        $\widetilde{u}_{k_s} = \widetilde{u}_k(t_s + \tau, W_s)$ % Save the value of the feedback control
        $k_1 = \delta F(W_s, T(t_s))$ % (save the values of $\widetilde{u}_k(t_s, W_s)$))
        $k_2 = \delta F(W_s + \frac{k1}{2}, T(t_s + \frac{\delta}{2}))$
        $k_3 = \delta F(W_s + \frac{k2}{2}, T(t_s + \frac{\delta}{2}))$
        $k_4 = \delta F(W_s + k3, T(t_s + \delta))$
        $W_{s+1} = W_s + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
        % END RUNGE-KUTTA 4th-order
    END
    $\widetilde{u}_{k_{N_{sim}}} = \widetilde{u}_k(t_{N_{sim}}, W_{N_{sim}}))$ % Save the Last value of the feedback control
    $R_\ell = \mathfrak{X}(W_{N_{sim}})$
% END Step $\ell$ - Closed loop simulation
```

References

[1] S. Cong, Control of Quantum Systems: Theory and Methods. John Wiley & Sons (2014).

[2] J.-M. Coron, Control and Nonlinearity. American Mathematical Society (2007).

[3] D. D'Alessandro, Introduction to Quantum Control and Dynamics. Chapman & Hall/CRC, Boca Raton (2008).

[4] P. de Fouquieres, S.G. Schirmer, S.J. Glaser and I. Kuprov, Second order gradient ascent pulse engineering. *J. Magn. Reason.* **212** (2011) 412–417.

[5] F. Diele, L. Lopez and R. Peluso, The Cayley transform in the numerical solution of unitary differential systems. *Adv. Comput. Math.* **8** (1998) 317–334.

[6] D. Dong and I.R. Petersen, Quantum control theory and applications: a survey. *IET Control Theory Appl.* **4** (2010) 2651–2671.

[7] Symeon Grivopoulos and Bassam Bamieh. Lyapunov-based control of quantum systems, in *42nd IEEE CDC*, Vol. 1 (2003) 434–438.

[8] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen and S.J. Glaser, Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J. Magn. Reason.* **172** (2005) 296–305.

[9] N. Leung, M. Abdelhafez, J. Koch and D. Schuster, Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Phys. Rev. A* **95** (2017) 042318.

[10] S. Machnes, E. Assémat, D. Tannor and F.K. Wilhelm, Tunable, flexible, and efficient optimization of control pulses for practical qubits. *Phys. Rev. Lett.* **120** (2018) 150401.

[11] M. Mirrahimi, Lyapunov control of a quantum particle in a decaying potential. *Ann. Inst. Henri Poincaré (C) Non-Linear Anal.* **26** (2009) 1743–1765.

[12] M. Mirrahimi, P. Rouchon and G. Turinici, Lyapunov control of bilinear Schrödinger equations. *Automatica* **41** (2005) 1987–1994.

[13] J.P. Palao and R. Kosloff, Quantum computing by an optimal control algorithm for unitary transformations. *Phys. Rev. Lett.* **89** (2002) 188301.

[14] J.P. Palao and R. Kosloff. Optimal control theory for unitary transformations. *Phys. Rev. A* **68** (2003) 062308.

[15] Y. Pan, V. Ugrinovskii and M.R. James, Lyapunov analysis for coherent control of quantum systems by dissipation, in *2015 American Control Conference (ACC)* (2015) 98–103.

[16] P.S. Pereira da Silva and P. Rouchon, RIGA and FPA, quantum control with smooth control pulses [source code] (2019).

[17] P.S. Pereira da Silva, P. Rouchon and H.B. Silveira, Geração rápida e virtualmente exata de portas quânticas via métodos iterativos do tipo Lyapunov, in *Proc. CBA'2018 - Congresso Brasileiro de Automática: CBA'2018, Brazilian Control Conference. João Pessoa, Brazil* (2018).

[18] P.S. Pereira da Silva, H.B Silveira and P. Rouchon, RIGA, a fast algorithm for quantum gate generation [source code] (2019).

[19] P.S. Pereira da Silva, H.B. Silveira and P. Rouchon, Fast and virtually exact quantum gate generation in U(n) via iterative Lyapunov methods. To appear in: *Int. J. Control* (2019). https://doi.org/10.1080/00207179.2019.1626023

[20] N. Rach, M.M. Müller, T. Calarco and S. Montangero, Dressing the chopped-random-basis optimization: a bandwidth-limited access to the trap-free landscape. *Phys. Rev. A* **92** (2015) 062343.

[21] B. Riaz, C. Shuang and S. Qamar, Optimal control methods for quantum gate preparation: a comparative study. *Quantum Inf Process* **18** (2019) 100.

[22] D.J. Saunders, The Geometry of Jet Bundles. Vol. 142 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, London (1989).

[23] S.G. Schirmer and P. de Fouquieres. Efficient algorithms for optimal control of quantum dynamics: the Krotov method unencumbered. *New J. Phys.* **13** (2011) 073029.

[24] H.B. Silveira, P.S. Pereira da Silva and P. Rouchon, Quantum gate generation by T-sampling stabilization. *Int. J. Control* **87** (2014) 1227–1242.

[25] H.B. Silveira, P.S. Pereira da Silva and P. Rouchon, Quantum gate generation for systems with drift in U(n) using Lyapunov-Lasalle techniques. *Int. J. Control* **89** (2016) 1–16.

[26] N. Yamamoto, K. Tsumura and S. Hara, Feedback control of quantum entanglement in a two-spin system. *Automatica* **43** (2007) 981–992.

[27] J. Zhang, Y.-x. Liu, R.-B. Wu, K. Jacobs and F. Nori, Quantum feedback: theory, experiments, and applications. *Phys. Rep.* **679** (2017) 1-60.