Fast and virtually exact quantum gate generation in U(n) via Iterative Lyapunov Methods

Published in International J. Control,
DOI: 10.1080/00207179.2019.1626023.

Paulo Sergio PEREIRA DA SILVA[1]
Hector Bessa SILVEIRA[2]
Pierre ROUCHON[3]

[1] University of São Paulo – USP – Brazil
[2] Federal University of Santa Catarina – UFSC – Brazil
[3] Systems and Control Centre (CAS)– MinesParistech – France

The first author was partially supported by CNPq, Brazil, Project 305546/2016-3, and by FAPESP, Brazil, Project 18/17463-7.

May, 2019

The MATLAB CODE regarding this paper can be found in https://codeocean.com/capsule/c0c3844b-68f7-4c46-b4fc-5ebdca8cb50d/

Abstract

- One studies the **quantum gate generation** problem in the context of coherent **(open loop)** control.
- One considers **right-invariant quantum systems with drift** with propagators evolving in U(n).
- This is a motion planning problem (open loop control) that could be considered by optimal control techniques.
- However, for large dimensions n, optimal control techniques are not applicable due to complexity issues.
- An approach is the numerical optimal control (GRAPE) (gradient-descent methods) that may produce excellent results.
- Another approach (ours) is the Lyapunov stabilization.
- The control is generated by a feedback law that may be recorded and applied in open loop.

- However, asymptotic methods produce slow controls, and cannot be exact for fixed T_f (final time).
- Normally, there is a compromise between precision and speed.
- To improve the precision one may increase T_f.
- This work develops a method that is **virtually exact** (exact up to a desired precision) for a fixed T_f.
- The numerical implementation of this method produces a method that is comparable to GRAPE.
- Numerical experiments with a benchmark has produced faster convergence than GRAPE.

Abstract (previous results of Silveira et al. [2016])

- The paper Silveira et al. [2016] have considered this problem via Lyapunov methods.
- The tracking of a trajectory $\overline{X}(t)$ is the main idea.
- The tracking problem was solved by a convenient Lyapunov-based feedback law...
- ... the recorded stabilizing control can be applied as an open loop control.
- The trajectory $\overline{X}(t)$ to be tracked has final condition $\overline{X}(T_f) = \overline{X}_{goal}$
- X_{goal} corresponds to the unitary operation to be performed by the quantum gate.
- T_f is big enough, (weak version of the problem).

Abstract (New Results)

- Main Contribution: **Reference Input Generation Algorithm (RIGA)** :
- It is essentially the repetition of the Lyapunov-based procedure N_1 times for a fixed T_f.
- This means, the successive simulations of the closed loop system.
- Given a reference trajectory $\overline{X}^{\ell} : [0, T_f] \rightarrow \mathrm{U}(n)$ with $\overline{X}^{\ell}(0) = I$.
- One obtains a new reference trajectory $\overline{X}^{\ell+1} : [0, T_f] \rightarrow \mathrm{U}(n)$ with $X^{\ell+1}(0) = I$.
- With $\mathrm{dist}(X^{\ell}(T_f), X_{goal}) < \mathrm{dist}(X^{\ell+1}(T_f), X_{goal})$ in each step ℓ.

Summary

Summary

Introductory remarks and references

- The basic problem of quantum gate generation relies on the motion planning of right-invariant and controllable systems with input $u(t) \in \mathbb{R}^m$ and state $X(t)$ evolving U(n) (or SU(n)).
- Note that $X(t)$ is the propagator (this is not state preparation, which is another problem!)
- Textbooks on quantum control D'Alessandro [2008],Cong [2014].
- Optimal control may produce fast controls in the context of state preparation and quantum gate generation Palao and Kosloff [2002] Palao and Kosloff [2003].
- However, for large n, that approach is not applicable due to complexity restrictions Schirmer and de Fouquieres [2011].

- An alternate approach: Lyapunov stabilization Yamamoto et al. [2007] Mirrahimi [2009] Mirrahimi et al. [2005] Grivopoulos and Bamieh [2003] Zhang et al. [2014] Pan et al. [2015] Dong and Petersen [2010].
- The present paper is mainly based on Lyapunov stabilization techniques that are developed in Silveira et al. [2016].
- Lyapunov methods normally lead to slow generation of the gate when compared to optimal control.
- Numerical schemes like GRAPE Khaneja et al. [2005] can consider the problem with large n.
- Our approach is meant to be an alternate method to GRAPE.

Feedback as a tool to generate open-loop controls

- Coherent quantum control are open-loop controls and can be generated “off-line”.
- They can be generated by “recording” a feedback-law in closed-loop (**old trick**).
- As the control may be generated “off-line”, one may use iterative methods.
- In each step a feedback is used in a (virtual) interval $[0, T_f]$ for generating better controls.
- This works as if one can repeat ℓ-times the interval $[0, T_f]$ (**new trick**).
- The off-line computational effort grows with the number of iterations, but with T_f fixed.
- The precision is improved exponentially with the increase of the number of iterations.

Quantum System Model (with Drift)

- System Model (Propagator equation)

$$\begin{aligned}\dot{X}(t) &= -\iota\left(H_0 + \sum_{k=1}^{m} u_k(t) H_k\right) X(t) \\ &= S_0 X(t) + \sum_{k=1}^{m} u_k(t) S_k X(t), \quad X(0) = X_0,\end{aligned}$$

- $X \in \mathrm{U}(n)$ is the state (propagator),
- $\iota \in \mathbb{C}$ is the imaginary unit,
- $S_0 = -\iota H_0, S_k = -\iota H_k \in \mathfrak{u}(n)$ (Anti-Hermitian Matrices),

Control Problem (strong version)

- **QUANTUM GATE GENERATION PROBLEM** :
- **Fix T_f (strong version).**
- **Find** controls $u_k : [0, T_f] \rightarrow \mathbb{R}$
  To steer $X(0) = I \dashrightarrow X(T_f) = X_{goal}$ **up to a desired precision ϵ.**
- $X_{goal} \in \mathrm{U}(n)$ represents the desired unitary operations of the gate.
- In this presentation one is interested in the **strong version of the problem**, where T_f **is fixed a priori.**
- One wants $\|X(T_f) - X_{goal}\| \leq \epsilon$ (Frobenius norm), where ϵ is also fixed a priori.

Previous results of Silveira et al. [2016]

- We recall now the **weak version** of the Quantum Gate Generation Problem. (final time T_f big enough).
- One summarizes the solution of this **weak version** Silveira et al. [2016].
- The idea is to track asymptotically a trajectory $\overline{X}(t)$ for which $\overline{X}(T_f) = X_{goal}$.

Previous results of the Silveira et al. [2016] (continued)

- **QUANTUM GATE GENERATION PROBLEM** :
- Find T_f big enough **(weak version)**
- Find controls $u_k : [0, T_f] \rightarrow \mathbb{R}$
  The goal : To steer from $X(0) = I \dashrightarrow X(T_f) = X_{goal}$
- One wants $\|X(T_f) - X_{goal}\| \leq \epsilon$ (Frobenius norm).
- $X_{goal} \in \mathrm{U}(n)$ represents the desired unitary operator representing the gate.

Lyapunov function

- The Lyapunov Function is defined by

$$\mathcal{V}(\widetilde{X}) = -\mathrm{Tr}\left(\frac{(\widetilde{X} - I)^2}{(\widetilde{X} + I)^2}\right) \geq 0, \tag{1}$$

- REMARK: The traditional Lyapunov function $\Re\left[\mathrm{Tr}\left(\widetilde{X}\right)\right]$ has a lot of critical points (this number increases with n).
- It may be shown that if the eigenvalues of $\widetilde{X}$ are $\{exp(\jmath\theta_1), \ldots, exp(\jmath\theta_n)\}$ then $\mathcal{V}(\widetilde{X}) = \sum_{j=1}^{n} \tan^2(\theta_j/2)$.
- $\mathcal{V} : \mathcal{W} \to \mathbb{R}$ is well defined in the open set $\mathcal{W} = \{W \in \mathrm{U}(n) \mid \det(I + W) \neq 0\}$.
- This Lyapunov function $\mathcal{V}(W)$ has a unique critical point in $\mathcal{W}$ (the identity).
- However it tends to infinity when W tends to $\mathcal{W}$

$\mathcal{V}$-distance

- "†" denotes the Hermitian operator (transpose and conjugate)
- $\text{dist}(\overline{X}, X) = \mathcal{V}(\overline{X}^{\dagger} X)$ is a left- and right-invariant notion of **distance** between unitary matrices $\overline{X}$ and X.
- $\text{dist}(\overline{X}R, XR) = \text{dist}(\overline{X}, X)$
- $\text{dist}(XR, X) = \text{dist}(R, I)$.
- The **closed $\mathcal{V}$-ball with radius** c, centered at X is defined by

$$\overline{B}_c^{\mathcal{V}}(X) = \{W \in \mathrm{U}(n) \mid \text{dist}(W, X) \leq c\} \tag{2}$$

- $\overline{B}_c(X) = \{W \in \mathrm{U}(n) \mid \|W - X\| \leq c\}$ is the **closed Frobenius-ball with radius** c, centered at X.

The reference system and the error system

- Consider that one has chosen reference inputs $\overline{u}_k(t)$. The reference trajectory will be a solution of the reference system:

$$\dot{\overline{X}}(t) = S_0\overline{X}(t) + \sum_{k=1}^{m} \overline{u}_k(t) S_k \overline{X}(t), \quad \overline{X}(0) = \overline{X}_0, \tag{3}$$

- Construct the error matrix

$$\widetilde{X}(t) = \overline{X}^{\dagger}(t) X(t)$$

- Note that $\mathcal{V}(\widetilde{X}(t)) = \mathrm{dist}(\overline{X}(t), X(t))$

- Simple computations produces the **(driftless)** *Error System*

$$
\begin{aligned}
\dot{\widetilde{X}}(t) &= \sum_{k=1}^{m} \widetilde{u}_k(t)\widetilde{S}_k(t)\widetilde{X}(t), \\
\widetilde{X}(0) &= \widetilde{X}_0 = \overline{X}_0^{\dagger} X_0, \\
\widetilde{u}_k(t) &= u_k(t) - \overline{u}_k(t), \\
\widetilde{S}_k(t) &= \overline{X}^{\dagger}(t) S_k \overline{X}(t)
\end{aligned}
$$

The Lyapunov-based feedback law

- The Error System is **driftless** (but time-varying),
- computing $\frac{d\mathcal{V}(\widetilde{X}(t))}{dt}$ one gets

$$\dot{\mathcal{V}} = \sum_{k=1}^{m} \mathcal{F}_k(\overline{X}, \widetilde{X})\widetilde{u}_k$$

- Roughly speaking, the functions $\mathcal{F}_k$ are "$\frac{\partial \mathcal{V}}{\partial \widetilde{X}} \cdot \widetilde{S}_k$"
- One may choose $\widetilde{u}_k = -f_k \mathcal{F}_k$
- $f_k > 0, k = 1, \ldots, m$ are chosen gains and
- $\dot{\mathcal{V}} = \sum_{k=1}^{m} -f_k \mathcal{F}_k^2 \leq 0$

Feedback law (continued)

- Some computations gives

$$\widetilde{u}_k(t) = \widetilde{U}_k(\overline{X}(t), \widetilde{X}(t)) = f_k \mathrm{Tr}\left[Z\left(\widetilde{X}(t)\right)\widetilde{S}_k(t)\right], \quad (4)$$

$$u_k(t) = \widetilde{u}_k(t) + \overline{u}_k(t), \quad (5)$$

$$\widetilde{S}_k(t) = \overline{X}^{\dagger}(t) S_k \overline{X}(t), \quad (6)$$

- where Z is the map defined by

$$Z(\widetilde{X}) = \widetilde{X}(\widetilde{X} - I)(\widetilde{X} + I)^{-3}. \quad (7)$$

Closed Loop Error System

- The *Closed Loop Error System* is given by

$$\dot{\widetilde{X}}(t) = \sum_{k=1}^{m} \widetilde{u}_k(t)\widetilde{S}_k(t)\widetilde{X}(t), \quad \widetilde{X}(0) = \widetilde{X}_0 = \overline{X}_0^{\dagger} X_0, \tag{8}$$

$$\widetilde{u}_k(t) = \widetilde{U}_k(\overline{X}(t), \widetilde{X}(t)) \tag{9}$$

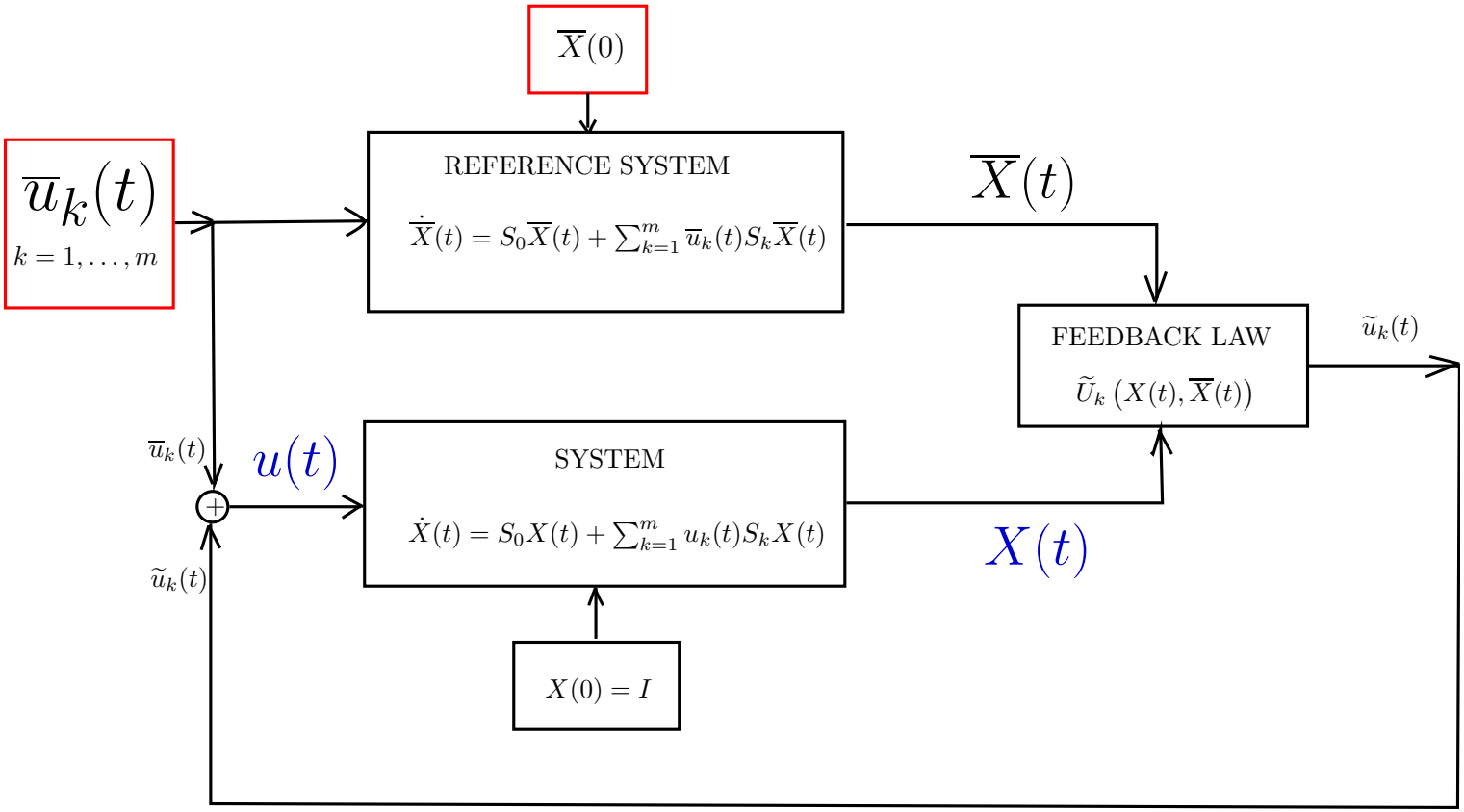


Figure: Simulation scheme for the closed loop system.

The periodic reference input

- The reference input must be T-periodic (to apply LaSalle's theorem to the closed-loop system).

$$\overline{u}_k(t) = \sum_{\ell=1}^{M} \left[a_{k\ell} \sin(2\ell\pi t/T) + b_{k\ell} \cos(2\ell\pi t/T) \right], \tag{10}$$

- The coefficients of these harmonics will be denoted by

$$\mathbf{a} = (a_{11}, a_{12}, \ldots, a_{1M}, \ldots, a_{m1}, a_{m2}, \ldots, a_{mM}) \tag{11}$$

$$\mathbf{b} = (b_{11}, b_{12}, \ldots, b_{1M}, \ldots, b_{m1}, b_{m2}, \ldots, b_{mM}) \tag{12}$$

- By a convenient right-translation, choosing T_f, one may always obtains $\overline{X}(T_f) = X_{goal}$.

Main results of Silveira et al. [2016].

Assume that $\widetilde{X}(0)$ is in the compact $\mathcal{V}$-ball $\overline{B}_c^{\mathcal{V}}(I)$.

- With **probability one with respect to the choice of a** and **b**, the closed loop system does not admit any nontrivial LaSalle's invariant.
- Proof based on Coron results (Coron's return method)
- By LaSalle's theorem one shows that $\widetilde{X}(t)$ converges to I.
- Equivalently, $X(t)$ converges to $\overline{X}(t)$.
- $\text{dist}(X(t), \overline{X}(t))$ converges (monotonically) **exponentially** to zero.
- Unfortunately the convergence speed is not "tunable" (presence of the drift).

Geometric Interpretation of the Lyapunov method I.

Given a reference $\overline{X}_{a,b}(t)$ with $\overline{X}_{a,b}(0) = I$

Compute a right-translation R such that

$$\overline{X}_{a,b}(T_f)R = X_{goal}.$$

Define $\overline{X}(t) = \overline{X}_{a,b}(t)R$. (parallel transport)

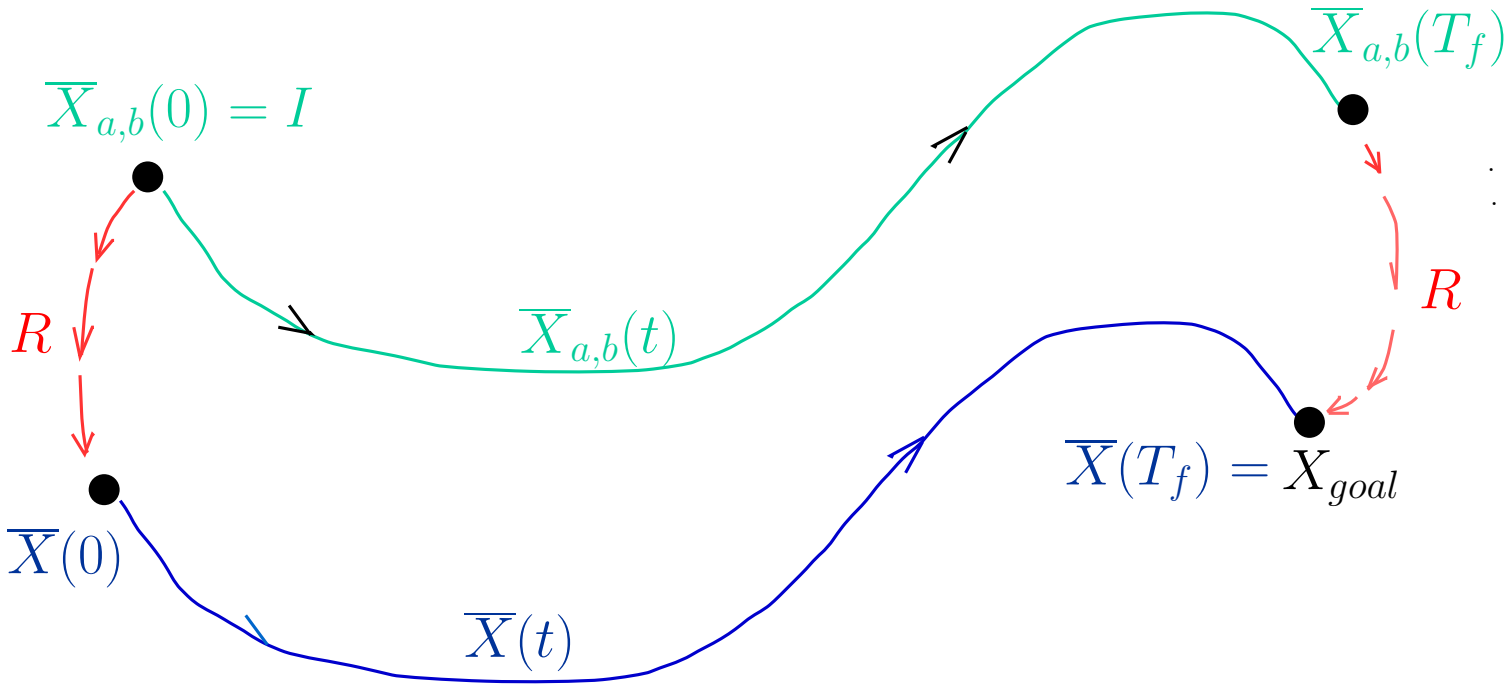


Figure: Geometric Interpretation of the (noniterative) Lyapunov method I.

Geometric Interpretation of the Lyapunov method II

Simulate the closed loop system.
Obtain $X(t)$ with $dist(X(T_f), X_{goal}) < dist(X(0), I)$.

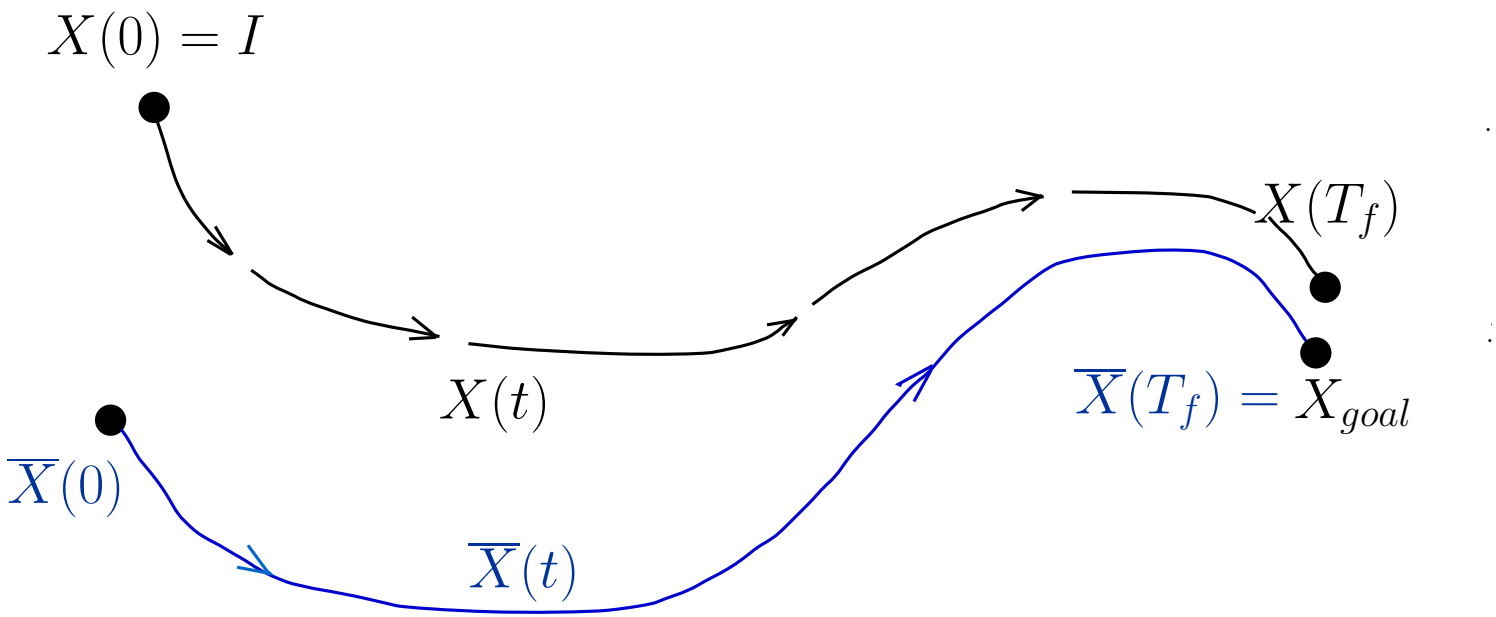


Figure: Geometric Interpretation of the (noniterative) Lyapunov method II.

(Non-iteractive) Lyapunov method

Choose T and coefficients $\mathbf{a}, \mathbf{b}$ of the periodic reference inputs $\overline{u}_k$.

- Choose T_f.
- Simulate reference system in $[0, T_f]$ (open loop) with $\overline{X}_{a,b}(0) = I$ and inputs $\overline{u}_k(t)$. Obtain $\overline{X}_{a,b}(T_f)$.
- Compute the right-translation $R = \overline{X}_{a,b}(T_f)^\dagger X_{goal}$.
- By construction $\overline{X}_{a,b}(T_f)R = X_{goal}$.
- Simulate $X(t)$ for the closed loop system in $[0, T_f]$ with:
  Reference inputs $\overline{u}_k$.
  $\overline{X}(0) = IR = R$.
  $X(0) = I$.
- Record $u_k(t) = \overline{u}_k(t) + \widetilde{u}_k(t)$
- Test if $\|X(T_f) - X_{goal}\| < \epsilon$.
- If this is not true, increase T_f and try again

Summary

Step ℓ of RIGA.

Roughly speaking each step of RIGA may be represented by the following block diagram:

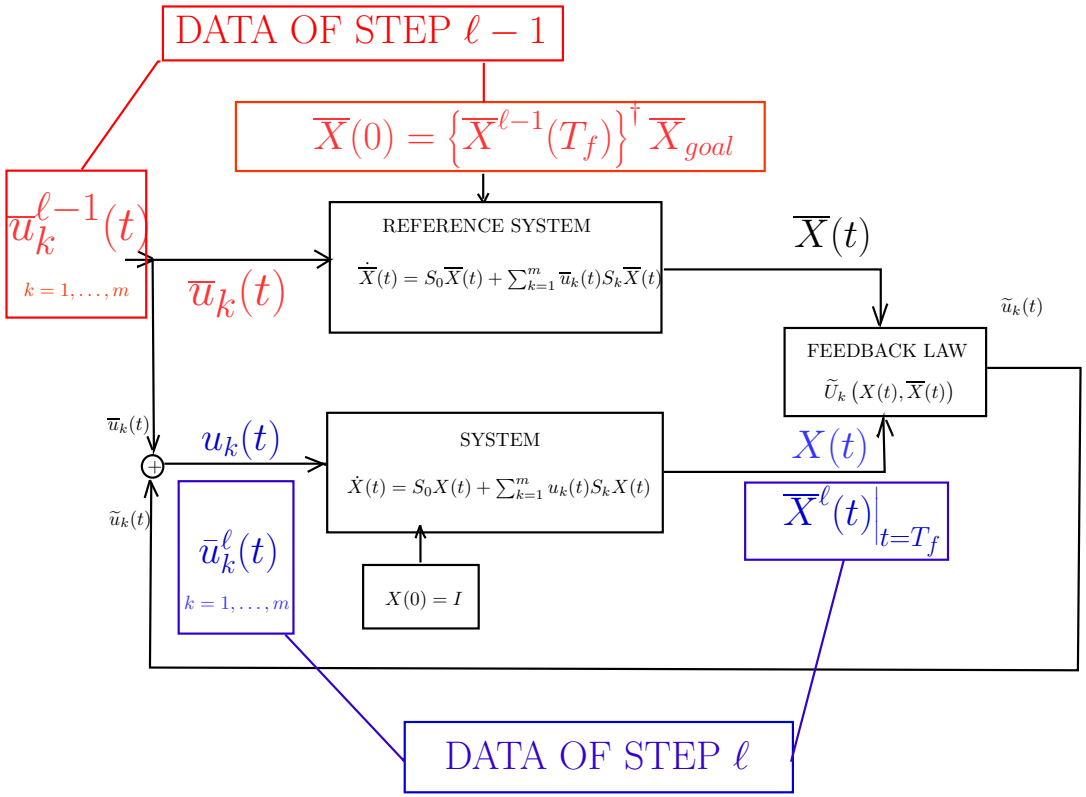


Figure: The "input data" is $\{\overline{X}^{\ell-1}(T_f),\overline{u}_k^{\ell-1}(\cdot)\}$ (marked in red). The output data is $\{\overline{X}^{\ell}(T_f),\overline{u}_k^{\ell}(\cdot)\}$ (marked in blue).

- The heart of the RIGA is ALGORITHM A, which is essentially represented by the last block diagram.
- We shall see that, in general, cannot choose $\overline{X}_{goal} = X_{goal}$, which is the natural choice.
- This choice could generate big correcting controls, or even singularities in the algorithm.
- Hence we include ALGORITHM B (Goal matrices generation) and ALGORITHM C (Goal matrix selection) whose descriptions will be given later.

Algorithm 1 - RIGA

The structure of RIGA is the following:
Algorithm 1 - RIGA

- **Step 0**
  SEED GENERATION (Choose $\overline{u}_k^0(\cdot)$ and simulate reference system)
  GOAL MATRICES GENERATION (Algorithm B - Define a set of goal matrices)
- **Step** ℓ
  GOAL MATRIX SELECTION (Algorithm C - Choose the goal matrix $\overline{X}_{goal}$).
  SINGLE ITERATION (Algorithm A - Simulate the closed loop system)

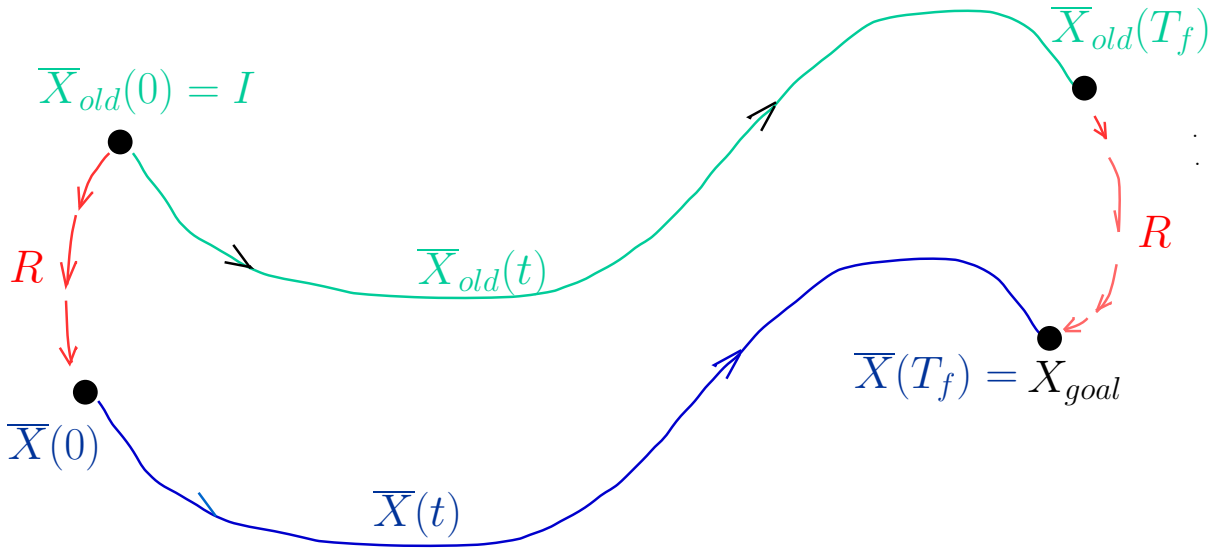
Seed Generation

Seed Generation. Choose $M > 0$ and reference controls $\overline{u}_k(t) : [0, T_f] \to \mathbb{R}$ of the form (10) for some choice of the pair $(\mathbf{a}, \mathbf{b})$. Determine $\overline{X}^0(t) = \overline{X}(t)$, $t \in [0, T_f]$, by numerical integration of system (3) with $\overline{X}^0(0) = \overline{X}(0) = I$. Define $\overline{u}_k^0(t) = \overline{u}_k(t)$, $k = 1, \ldots, m$, $t \in [0, T_f]$, and save the value of $X_f = \overline{X}^0(T_f)$.

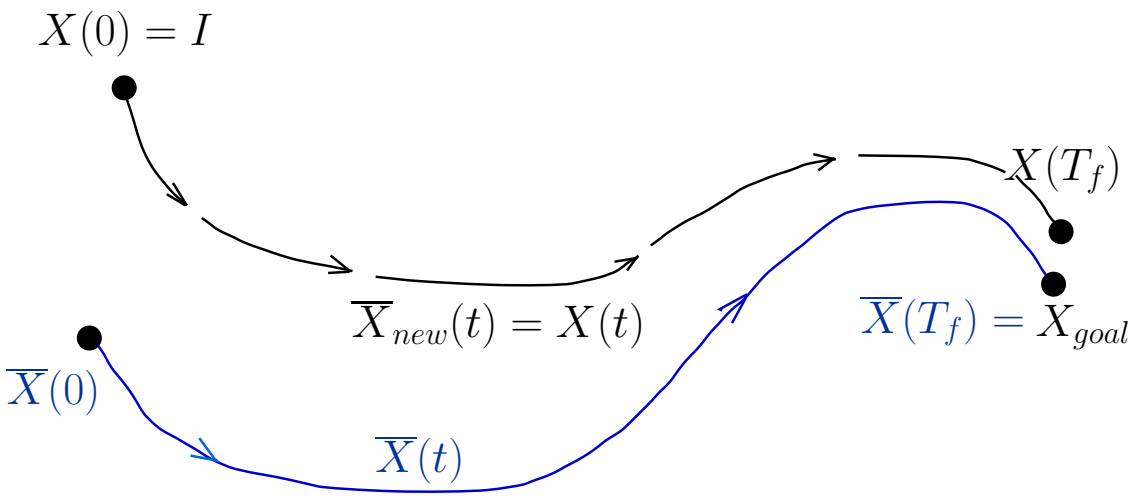
Algorithm A - SINGLE ITERATION

- The heart of RIGA, is Algorithm A, that is the simulation of the closed loop system for $t \in [0, T_f]$.
- The input of step ℓ is $\overline{X}^{\ell-1}(\cdot) = \overline{X}_{old}(\cdot)$.
- The output of step ℓ is $\overline{X}^{\ell}(\cdot) = \overline{X}_{new}(\cdot)$.

Geometric Description of Algorithm A (Single Iteration)



FIRST PHASE: PARALLEL TRANSPORT – MOVE $\overline{X}_{old}(T_f)$ TO X_{goal}



SECOND PHASE: CLOSED LOOP SYSTEM SIMULATION – OBTAIN $\overline{X}_{new}(t)$

Algorithm A (Single Iteration). Fix $\overline{X}_{goal} \in \mathrm{U}(n)$. Fix $T_f > 0$. Assume that a reference input $\overline{u}^{old} : [0, T_f] \to \mathbb{R}^m$ is given and let $\overline{X}_{old}(t)$ be the corresponding pre-computed reference trajectory with $\overline{X}^{old}(0) = I$. Then

Phase 1. Compute the right-translation $R \in \mathrm{U}(n)$ such that $\overline{X}_{old}(T_f)R = \overline{X}_{goal}$, that is $R = \overline{X}_{old}(T_f)^{\dagger}\overline{X}_{goal}$.

Phase 2. Numerically integrate the closed loop system for $t \in [0, T_f]$ with $\overline{u}(t) = \overline{u}^{old}$, $X(0) = I$ and $\overline{X}(0) = R$. By right-invariance, $\overline{X}(T_f) = \overline{X}_{goal}$.

Obtain the solution $\overline{X}^{new}(t) = X(t)$ and the input $\overline{u}^{new}(t) = u(t)$, where $X(t)$ and $u(t) = \widetilde{u}(t) + \overline{u}(t)$ are respectively the solution and the control-law that was computed for the closed-loop system.

- A first idea for the RIGA (that is called by Algorithm 2 - RIGA - local version in the paper) is to eliminate Algorithm B and to consider that Algorithm C is the simple procedure that chooses $\overline{X}_{goal} = X_{goal}$ in every step ℓ.
- The local version of RIGA cannot work well if dist$(\overline{X}^{\ell}(T_f), X_{goal})$ is too big.
- The feedback inputs will become too big in this case. This could be undesirable.
- Even if one accepts big inputs, this may cause numerical instabilities and imprecisions.

A first solution for this problem to consider that Algorithm B is void and to consider that Algorithm C is the “saturation of angles”:

- Let $X_f = \overline{X}^{\ell-1}(T_f)$ be final condition of step $\ell - 1$. Let $R = X_f^{\dagger} X_{goal} = W^{\dagger} \mathrm{diag}[\exp(\imath\theta_1), \ldots, \exp(\imath\theta_n)] W$ (its eigenstructure with $\theta_i \in [-\pi, \pi]$, that may be computed by the Schur decomposition). Then:
- $$R_{sat} = W^{\dagger} \mathrm{diag}\left\{\exp[\imath \mathrm{sat}(\theta_1)], \ldots, \exp[\imath \mathrm{sat}(\theta_n)]\right\} W, \quad (13a)$$
$$\overline{X}_{goal} = X_f R_{sat} \quad (13b)$$
- where $\mathrm{sat}(\theta) = \theta$ for $\theta \in [-\pi/4, \pi/4]$, $\mathrm{sat}(\theta) = \pi/4$ for $\theta > \pi/4$ and $\mathrm{sat}(\theta) = -\pi/4$ when $\theta < -\pi/4$.

Switching the Goal Matrices

- A second solution is the policy of “switching the Goal Matrices.”
- We can consider that ALGORITHM B create a set of goal matrices $X_{goal}^{q} : q = 1, 2, \ldots, p$ and then ALGORITHM C will “switch” along the steps of the algorithm, choosing one of them.

Switching the Goal Matrix

- The distance $\text{dist}(X^{q}_{goal}, X^{q-1}_{goal}) \leq \alpha$ for a chosen α for $q = \{1, 2, \ldots, p\}$.
- $X^{0}_{goal} = \overline{X}^{0}(T_f)$.
- The final goal matrix $X^{p}_{goal} = X_{goal}$.
- The index $q \in \{1, 2, \ldots, p\}$ that defines X^{q}_{goal} is increased along the steps $\ell = 1, 2, 3, \ldots$ of the algorithm.
- The test for switching the value of q in a step ℓ is to choose the greater q such that $\text{dist}(\overline{X}^{\ell}(T_f), X^{q}_{goal}) < \beta$,
- The switching value β is such that $0 < \alpha < \beta$.

Algorithm B: (Goal Matrices Generation)

Algorithm B: (Goal Matrices Generation). Fix $\alpha > 0$. For a minimal $p \in \mathbb{N}$, compute $X^q_{goal}, q = 0, \ldots, p$ with $X^0_{goal} = \overline{X}^{\ell}(T_f)$, $X^p_{goal} = X_{goal}$ and $\text{dist}(X^{q-1}_{goal}, X^q_{goal}) < \alpha$ for $q = 1, \ldots, p$. Set $q_{last} = 1$.

Algorithm C: (Goal Matrices Selection)

Algorithm C: (Goal Matrices Generation).
(Trying to switch $\overline{X}_{goal}$). If $q_{last} < p$, let q^* be the greater $q \in \{q_{last}, \ldots, p\}$ such that $\text{dist}(\overline{X}^{\ell-1}(T_f), X^q_{goal}) < \beta$. Set $\overline{X}_{goal} = X^{q^*}_{goal}$. Set $q_{last} = q^*$.
(No switching). If $q_{last} = p$, then $\overline{X}_{goal} = X_{goal}$.

Switching the goal matrices.

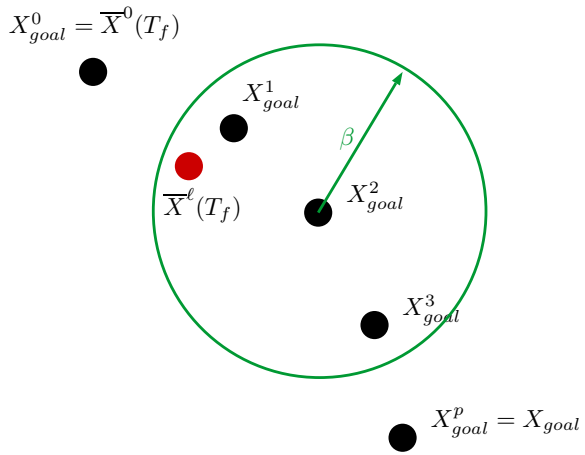


Figure: The figure shows an example of the switching policy. One has $p = 4$ goal matrices. In the present step ℓ one has $q = 1$, that is, the present goal matrix that is chosen is X^1_{goal}. Observe that $\overline{X}^\ell(T_f)$ is close enough to X^1_{goal} in a way that $\text{dist}(\overline{X}^\ell(T_f), X^2_{goal}) < \beta$. Hence in this step one will switch to $q = 2$.

Input saturations

- One may include saturation of inputs in Algorithm A of RIGA.
- It suffices to apply to the quantum system a saturated input:

$$u_k(t) = \text{sat}\left(\overline{u}_k(t) + \widetilde{u}_k(t)\right), k = 1, \ldots, m$$

- It can be shown that the $\frac{d\mathcal{V}(\widetilde{X}(t))}{dt}$ **remains non-positive for the saturated closed system** .
- See the paper for a smooth version for the saturation function.

Summary

Attractive reference trajectories

- The definition of a λ-attractive reference trajectory is introduced in the paper.
- This property holds with probability one with respect to the jet of the reference output in a sense that is precised in the paper.
- If a reference trajectory $\overline{X}^{old}$ is λ-attractive, then in Algorithm A one has $\text{dist}(X_{goal}, \overline{X}^{old}(T_f)) \leq \lambda \text{dist}(X_{goal}, \overline{X}^{new}(T_f))$.

Convergence of the local RIGA - Algorithm 2

Theorem

Fix $X_{goal} \in U(n)$. Choose $T > 0$, $M > 0$, $c > 0$ and coefficients $(\mathbf{a}, \mathbf{b})$ for the reference input (10), and consider the corresponding reference trajectory $\overline{X}^0 : [0, T_f] \to U(n)$ with $\overline{X}^0(0) = I$, which defines a seed of Algorithm 2. Assume that this seed is a λ-attractive trajectory on $\overline{B}_c^{\mathcal{V}}$ (this is true with probability 1 for M big enough). If $dist(\overline{X}^0(T_f), X_{goal}) \leq c$, then there exists sufficiently large T_f, such that:

- *The reference trajectory generated in step ℓ of Algorithm 2 is θ-attractive on $\overline{B}_c^{\mathcal{V}}$ for some $0 < \theta < 1$ and for all $\ell \in \mathbb{N}$;*
- *Algorithm 2 converges exponentially, that is, $d_\ell \leq \theta^\ell d_0$, where $d_\ell = dist(X_{goal}, \overline{X}^\ell(T_f))$;*
- *The reference inputs that are generated by Algorithm 2 are uniformly bounded. In other words, there exists $H > 0$ such that $\|\overline{u}_k^\ell(t)\| < H$, for all $t \in [0, T_f]$, all $k = 1, \ldots, m$ and all $\ell \in \mathbb{N}$.*

Convergence Result for the global RIGA-Algorithm 1

Theorem

Fix $X_{goal} \in U(n)$. Choose $\alpha > 0, \beta > \alpha$ and $c \leq \beta$. Choose $T > 0$, $M > 0$ and coefficients $(\mathbf{a}, \mathbf{b})$ for the reference input (10), and consider the corresponding reference trajectory $\overline{X}^0 : [0, T_f] \to U(n)$ with $\overline{X}^0(0) = I$, which defines a λ-attractive seed of Algorithm 1. Then, for all sufficiently large T_f:

- *The reference trajectory generated in step ℓ of Algorithm 1 is θ-attractive for some $0 < \theta < 1$ and for all $\ell \in \mathbb{N}$;*
- *Step ℓ of Algorithm 1 switches until the final goal matrix X_{goal} is chosen, when it reduces to step ℓ of Algorithm 2. When Algorithm 1 switches in step ℓ, then d_ℓ may be greater than $d_{\ell-1}$, but with $d_\ell \leq \beta$, where $d_\ell = dist(X_{goal}, \overline{X}^\ell(T_f))$. Between switchings, the inequality $d_\ell \leq \theta d_{\ell-1}$ always holds. In particular, after the last switch, d_ℓ converges exponentially to zero, monotonically.*
- *The reference inputs generated by Algorithm 1 are uniformly bounded, as in the statement of Theorem 1.*

Summary

Numerical Implementation

- 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 $\delta = T_f / N_{sim}$.
- Consider that the inputs (and the reference inputs) are piecewise contant in the intervals $[t_s, t_{s+1})$.
- Denote $T_s = S_0 + \sum_{k=1}^{m} u_k(t_s) S_k$ and $\overline{T}_s = S_0 + \sum_{k=1}^{m} \overline{u}_k(t_s) S_k$ where $S_k, k = 0, \ldots, m$ are the system matrices.

Numerical Implementation

- The simulation of the closed loop system is computed by

$$\begin{aligned} \overline{X}(t_{s+1}) &= \exp\left(\delta \overline{T}_s\right) \overline{X}(t_s) \\ u_k(t_s) &= \overline{u}_k(t_s) - f_k \zeta_k(t_s), k = 1, \ldots, n \\ X(t_{s+1}) &= \exp\left(\delta T_s\right) X(t_s) \end{aligned}$$

- where $\zeta_k(t_s) = -\mathrm{Tr}\left[Z\left(\widetilde{X}(t_s)\right)\widetilde{S}_k(t_s)\right]$, and the map Z is given by (7).
- The exponential of a matrix δS will be computed using a $4th$-order Padé approximation:

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

Summary

Example 1: Three coupled qubits

- The present example was considered in Problem 1 of [N. Khaneja and S. J. Glaser and R. Brockett] (2002) concerning three coupled qubits..
- Minimum time $T_f = T^*$ for the gate generation is known.
- The Hilbert space is $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2$, each $\mathbb{C}^2$ is the space of one qubit. The propagator evolves on U(8)
- The spin Pauli matrices (and the identity) are denoted by:

$$I_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, I_y = \frac{1}{2}\begin{pmatrix} 0 & \jmath \\ \jmath & 0 \end{pmatrix}, I_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- The following matrices are useful for defining the system dynamics

$$I_{1w} = I_w \otimes I_2 \otimes I_2, I_{2w} = I_2 \otimes I_w \otimes I_2, I_{3w} = I_2 \otimes I_2 \otimes I_w$$

where w may represent x, y, or z.

- For instance, I_{1x} stands for $I_x \otimes I_2 \otimes I_2$.

- Consider the controllable quantum system model

$$\dot{X}(t) = -\imath H_0 X(t) - \imath \sum_{k=1}^{7} u_k(t) H_k X(t), \quad X(0) = X_0 = I,$$

- where the Hamiltonian matrices of the systems are given by $H_0 = 2\pi J I_{1z} I_{2z} + 2\pi J I_{2z} I_{3z}$ (drift term), $H_1 = 2\pi I_{1x}$, $H_2 = 2\pi I_{1y}$, $H_3 = 2\pi I_{2x}$, $H_4 = 2\pi I_{2y}$, $H_5 = 2\pi I_{3x}$, $H_6 = 2\pi I_{3y}$, $H_7 = I_2 \otimes I_2 \otimes I_2$ (global phase control) and $J = 1$ is a constant (normalized value).
- The aim is to generate the gate corresponding to $X_{goal} = \exp(-\jmath\theta I_{1z} I_{2z} I_{3z})$ whose optimal minimum generation time is known and given by

$$T^* = \frac{\sqrt{2\pi\theta - (\theta/2)^2}}{2\pi J}$$

- For the presented simulations one has chosen $\theta = \pi$.

- The feeback gains are all equal to $f_k = \pi, k = 1, 2, \ldots, 7$.
- To initialize the reference $X^0(t)$ of Phase 1, one one has used T-periodic reference controls with $T = 10T^*$.
- One has used $M = 5$ (the number of sinusoidal-cosinusoidal harmonics of the seed).
- One has chosen for the seed of all simulations random (uniform independently distributed) values of the entries of $(\mathbf{a}, \mathbf{b})$ in the interval $[-Am, Am]$.
- One has chosen $A_m = 1/10000$. This very small amplitude A_m of the reference input generates a reference trajectory $\overline{X}^0(t)$ that is close to the null input trajectory.
- However, as the number of the step ℓ increases, the reference input is shaped in a certain way that produce very good results.

- One has used input saturation with $u_{max} = 12$ (equal saturation for all inputs for all the steps $\ell \leq 500$.
- When the step number $\ell > 500$, one considers $u_{max} = 15$.
- This last situation will occur only with $N_1 > 500$.

Simulation results for Example 1. The number of steps is N_1. Final time was normalized with respect to T^*. The error ϵ is the Frobenius norm $\|\overline{X}(T_f) - X_{goal}\|_{\ell=N_1}$. The run time of our MATLAB© program regards a standard PC with Windows 10®. The worst case fidelity regards the probability of an error in the operation of the gate in the worst case.

T_f/T^*	N_1	Error ϵ	run time (s)	$1-\mathcal{F}$
2.0	200	1.79×10^{-4}	111.0	3.74×10^{-9}
1.35	200	1.34×10^{-5}	110.0	1.81×10^{-11}
1.3	400	0.00351	218.0	1.18×10^{-6}
1.1	2000	0.0098	1122.0	8.14×10^{-6}
1.1	5000	4.95×10^{-4}	2755.0	2.08×10^{-8}
1.0	5000	0.117	2766.0	0.00113

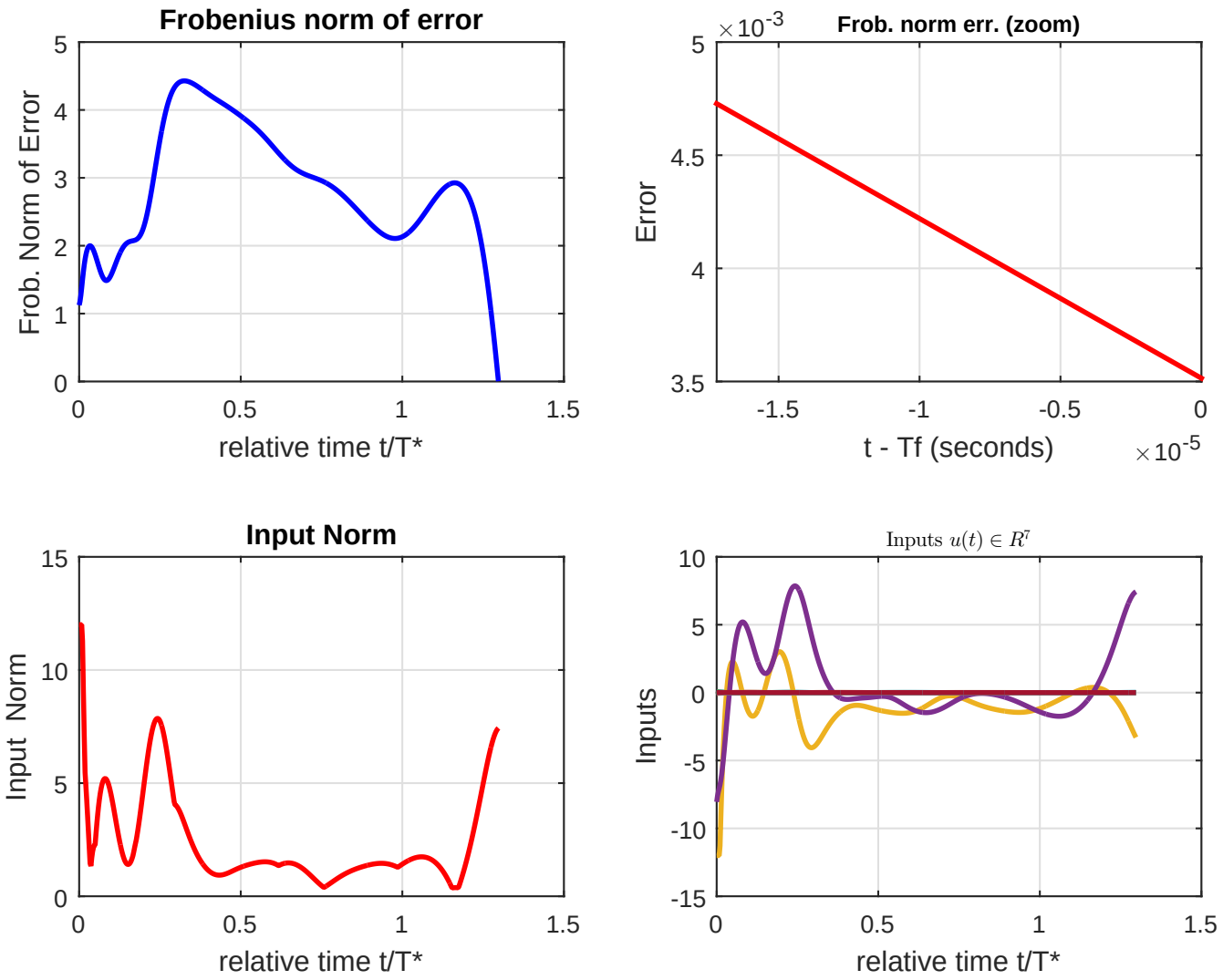


Figure: Simulation results for System 1 (Three Qubits), with $T_f / T^* = 1.3$, $N_1 = 400$.

Switching the goal matrices in the Example 1

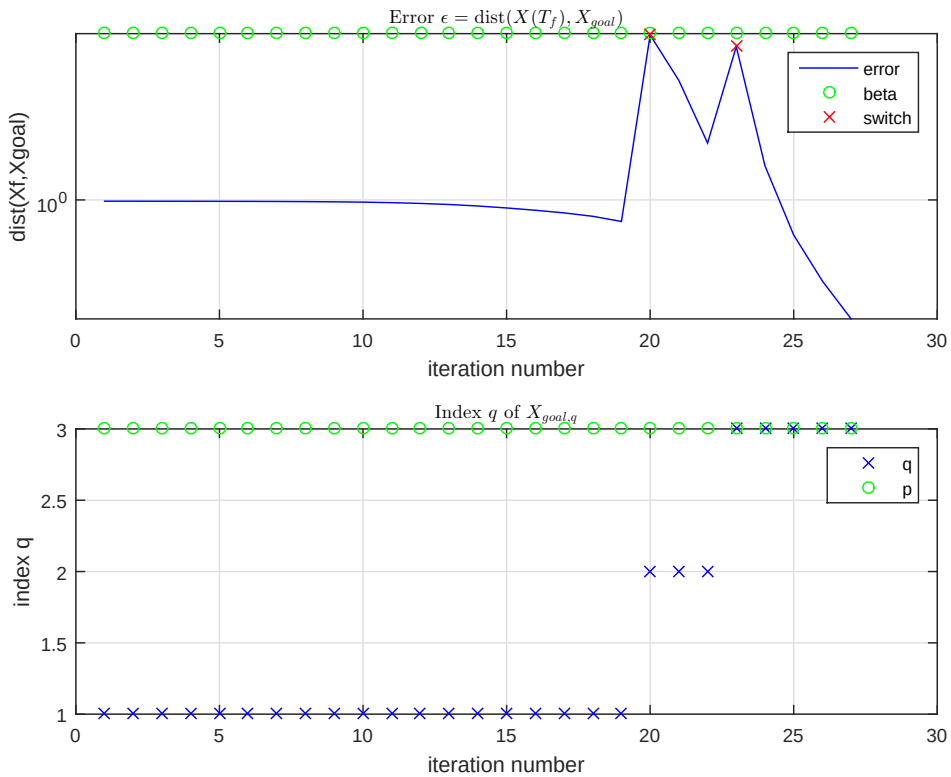


Figure: This picture shows the evolution of $\mathrm{dist}(\overline{X}^{\ell}(T_f), X_{goal}) = \mathrm{dist}(\overline{X}^{\ell+1}(0), X_{goal})$ obtained in Example 1 for $N_1 = 400$ and $T_f/T^* = 1.3$. Note that after a switching, the value of q is increased and this distance can also increase but is always bounded by $\beta = 5$.

Exponential convergence of the error

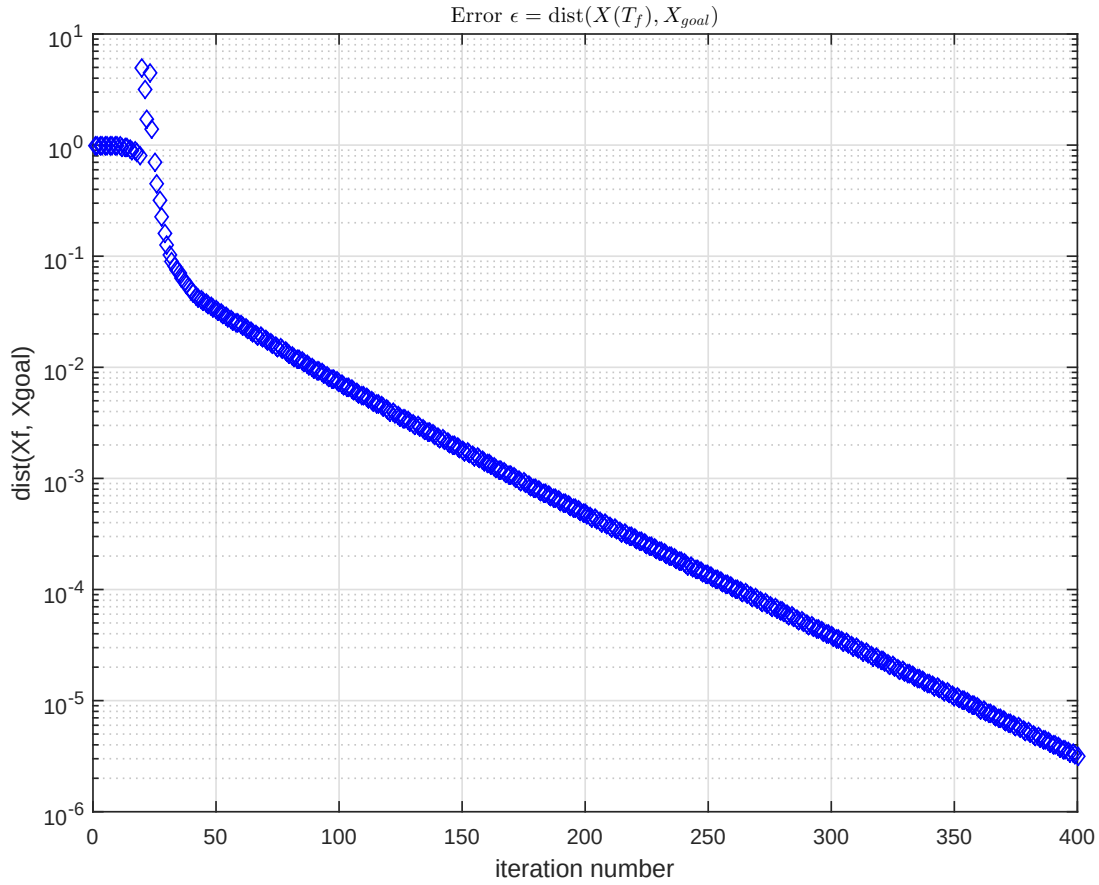


Figure: The figure illustrates the exponential convergence of Algorithm 1. One notes that $\lim_{\ell \to \infty} \mathrm{dist}(\overline{X}^{\ell}(T_f), X_{goal}) = 0$ exponentially. It presents simulation results for System 1 (Three Qubits), with $T_f / T^* = 1.3$, and $N_1 = 400$.

The choice of $(\mathbf{a}, \mathbf{b})$ (seed) is not that important

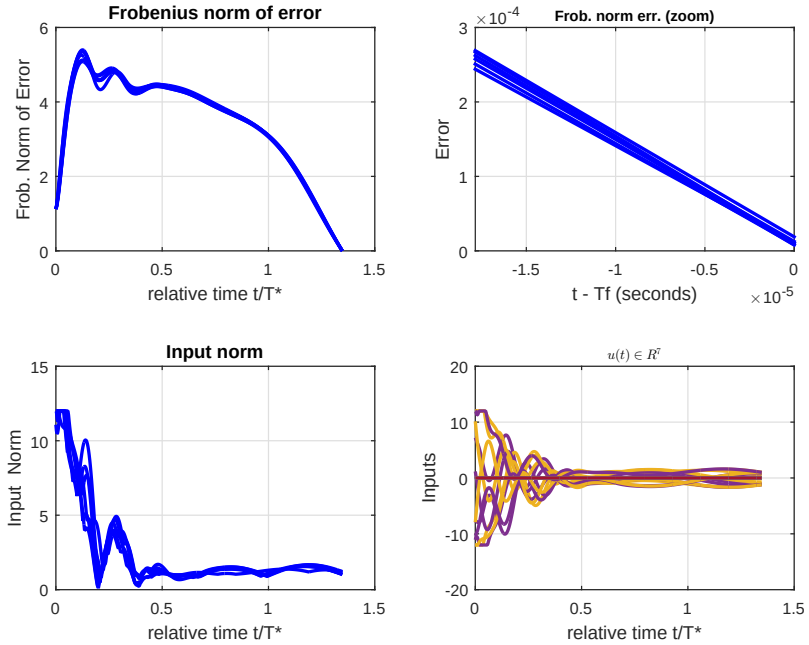


Figure: This figure presents simulation results for System 1 (Three Qubits), with $T_f/T^* = 1.35$, and $N_1 = 200$. One shows six superposed simulations with different choices $(\mathbf{a}, \mathbf{b})$ (seed). In each simulation, the $2Mm$ entries of $(\mathbf{a}, \mathbf{b})$ are (uniformly independently) randomly chosen in the interval $(-1e-5, 1e-5)$. The final error is almost independent on this choice. The inputs that are generated in this way may vary, but the their norm is rather independent of such choice.

Example 2: Coupled cavity-qubit

- One considers the SNAP gate described in Heeres et al. [2015].
- It is coupled cavity-qubit of the form

$$\dot{\mathcal{X}}(t) = -\iota \left\{ H_0 + H_1 v_x(t) + H_2 v_y(t) \right\} \mathcal{X}(t), \tag{14}$$

- $H_0 = H_{cc} + H_{qq} + H_{int}$ is the Hamiltonian of the system.
- $H_{cc} = H_c \otimes I_2$ is the cavity Hamiltonian, with $H_c = \omega_c(a^\dagger a) + \frac{K_2}{2}(a^\dagger)^2 a^2 + \frac{K_3}{6}(a^\dagger)^3 a^3$,
- The qubit Hamiltonian is given by $H_{qq} = \omega_q(I \otimes E)$, with $E = |e\rangle\langle e|$,
- The interaction Hamiltonian is given by $H_{int} = H_i \otimes E$, with $H_i = \chi(a^\dagger a) + \frac{\chi'}{2}(a^\dagger)^2 a^2 + \frac{\chi''}{6}(a^\dagger)^3 a^3$.
- The control Hamiltonians (only on the qubit) are $H_1 = I_n \otimes \sigma_x$ and $H_2 = I_n \otimes \sigma_y$.

- Note that there is no actuation over the cavity. Only the qubit is driven by the two local real controls $v_x(t)$ and $v_y(t)$.
- The operators a and $a^\dagger$ are the usual destruction and creation operators
- Note the state of this system evolves on $\mathbb{C}^n \otimes \mathbb{C}^2$, and hence the propagator $\mathcal{X}(t)$ is a $2n \times 2n$ matrix.
- Data: cavity frequency $\omega_c/(2\pi) = 8226.787$ MHz; qubit pulsation frequency $\omega_q/(2\pi) = 7627.05$ MHz; dispersive shift $\chi/(2\pi) = 8281.3$ kHz and its two corrections $\chi'/(2\pi) = 48.8$ kHz, $\chi''/(2\pi) = 0.5kHz$; Kerr effect $K_2/(2\pi) = -107.9$ kHz and its correction $K_3/(2\pi) = 3.4$ kHz.
- In the numerical experiments, these parameters are normalized with respect to χ. So, one chooses $\chi = 2\pi$. For $T_f = 4\pi/\chi$ one gets $T_f = 2$ n.u., where 1 n.u. is equal to $2\pi/\chi$ seconds.

THE SNAP GATE

- Given a set $\{\theta_0, \theta_1, \ldots, \theta_{n-1}\} \subset \mathbb{R}$, the SNAP is defined by the steering the state from $|k\rangle \otimes |g\rangle$ to $\exp(\iota\theta_k)\left(|k\rangle \otimes |g\rangle\right)$ for $k = 0, 1, \ldots, n-1$.
- The SNAP gate to be considered is defined by $\theta_0 = 0$, $\theta_1 = \pi/2$, $\theta_3 = 0$, $\theta_4 = \pi/2$, and so on. A first set of simulations has considered for n equal to ten levels (from level zero to level 9). A second set of simulations has considered n equal to fifteen levels.

- Fix some ω_r and let

$$v_x(t) + \iota v_y(t) = \left[u_x(t) + \iota u_y(t)\right] \exp(\iota \omega_r t). \tag{15}$$

- This is similar to the context of the Rotating Wave Approximation (RWA), but here one will obtain an exact expression.
- It is shown in the Appendix that this problem reduces to the control of a system defined on $\mathrm{SU}(2)^n = \underbrace{\mathrm{SU}(2) \times \ldots \times \mathrm{SU}(2)}_{\text{n-times}}$ given by:

$$\begin{aligned} \dot{W}_k(t) &= -\iota \left\{ \tfrac{(\alpha(k)-\omega_r)}{2} \sigma_z + \sigma_x u_x(t) + \sigma_y u_y(t) \right\} W_k(t), \\ & k = 0, 1, \ldots, n-1, \end{aligned} \tag{16}$$

- $\alpha(k) = \omega_q + \langle k | H_i | k \rangle$, $k = 0, 1, \ldots, n-1$. The matrices $W_k(t)$ evolves on SU(2).

- One may show that the proposed control problem regarding the SNAP gate is equivalent to steering the ensemble (16) from $W_k(0) = I_2$ to $W_k(T_f) = W_{goal,k}$ for $k = 0, \ldots n-1$, where:

$$W_{goal,k} = \exp\left(\iota \frac{\omega_r}{2} T_f \sigma_z\right) \begin{bmatrix} \exp[\iota(\theta_k + \gamma(k))] & 0 \\ 0 & \exp[-\iota(\theta_k + \gamma(k))] \end{bmatrix}, \tag{17}$$

- $\gamma(k) = \exp\left\{\iota \left[\frac{\alpha(k)}{2} + \beta(k)\right] T_f\right\}$ and $\beta(k) = \langle k|H_c|k\rangle\,,\ k = 0, 1, \ldots, n-1.$

- Note that $SU(2)^n$ is (canonically) a Lie group.
- The product of $W^1 = (W_1^1, \ldots W_n^1)$, $W^2 = (W_1^2, \ldots W_n^2) \in SU(2)^n$ is defined by $W^1 W^2 = (W_1^1 W_1^2, \ldots W_n^1 W_n^2)$.
- Lie brackets may be defined componentwise and the controllability of system (16) was checked by a computer program for the given set of parameters ω_c, ω_q, χ, χ', χ'', K_2, K_3.
- The algorithm proposed in this paper may be applied with some trivial adaptations and a convenient Lyapunov function.

- Given a reference trajectory $\overline{W}(t) = (\overline{W}_1(t), \ldots, \overline{W}_n(t))$, clearly the **error matrix** is $\widetilde{W}(t) = \overline{W}^{\dagger}(t)W(t) = (\widetilde{W}_1(t), \ldots, \widetilde{W}_n(t)) = (\overline{W}_1^{\dagger}(t)W_1(t), \ldots \overline{W}_n^{\dagger}(t)W_n(t))$.
- Simple manipulations produce the error matrix dynamics:

$$\dot{\widetilde{W}}_k(t) = -\iota \left\{ \widetilde{\sigma}_x^k(t)u_x + \widetilde{\sigma}_y^k(t)u_y \right\} \widetilde{W}_k(t), \ k = 0, 1, \ldots, n-1, \quad (18)$$

- $\widetilde{\sigma}_x^k(t) = \overline{W}_k^{\dagger}(t)\sigma_x \overline{W}_k(t)$ and $\widetilde{\sigma}_y^k(t) = \overline{W}_k^{\dagger}(t)\sigma_y \overline{W}_k(t)$.
- Let $\mathcal{W} = \{W \in \mathrm{SU}(2) \mid \det(W + I_2) \neq 0\}$. Define the Lyapunov function $\mathcal{L} : \mathcal{W}^n \to \mathbb{R}$ by $\mathcal{L}(\widetilde{W}_1, \ldots, \widetilde{W}_n) = \sum_{k=1}^n \mathcal{V}(\widetilde{W}_k)$, where $\mathcal{V}$ is the Lyapunov function of this paper.
- $\mathcal{L}$ vanishes only at $\mathcal{I} = (I_2, I_2, \ldots, I_2)$.

- Computations similar to the ones presented in Silveira et al. [2014] Silveira et al. [2016] give: $\frac{d}{dt}\mathcal{L}(\widetilde{W}(t)) = \widetilde{u}_x F_x(t) + \widetilde{u}_y F_y(t)$,
- $F_x(t) = -\sum_{i=1}^{n} 4\text{trace}\left[(\widetilde{W}_k(t) - I_2)(\widetilde{W}_k(t) + I_2)^{-3}\widetilde{\sigma}_x^k\right]$
- $F_y(t)$ has a similar expression, replacing $\widetilde{\sigma}_x^k$ by $\widetilde{\sigma}_y^k$.
- The control law $(\widetilde{u}_x(t), \widetilde{u}_y(t)) = (-K_x F_x(t), -K_y F_y(t))$, where $K_x > 0$ and $K_y > 0$ are (real) feedback gains, assures that $\dot{\mathcal{L}} \leq 0$,
- One may adapt all the results of Silveira et al. [2016] to this system, showing exponential convergence of $\widetilde{W}(t)$ to $\mathcal{I}$.
- One has translated the SNAP gate problem of Heeres et al. [2015] to a gate generation for system (16).

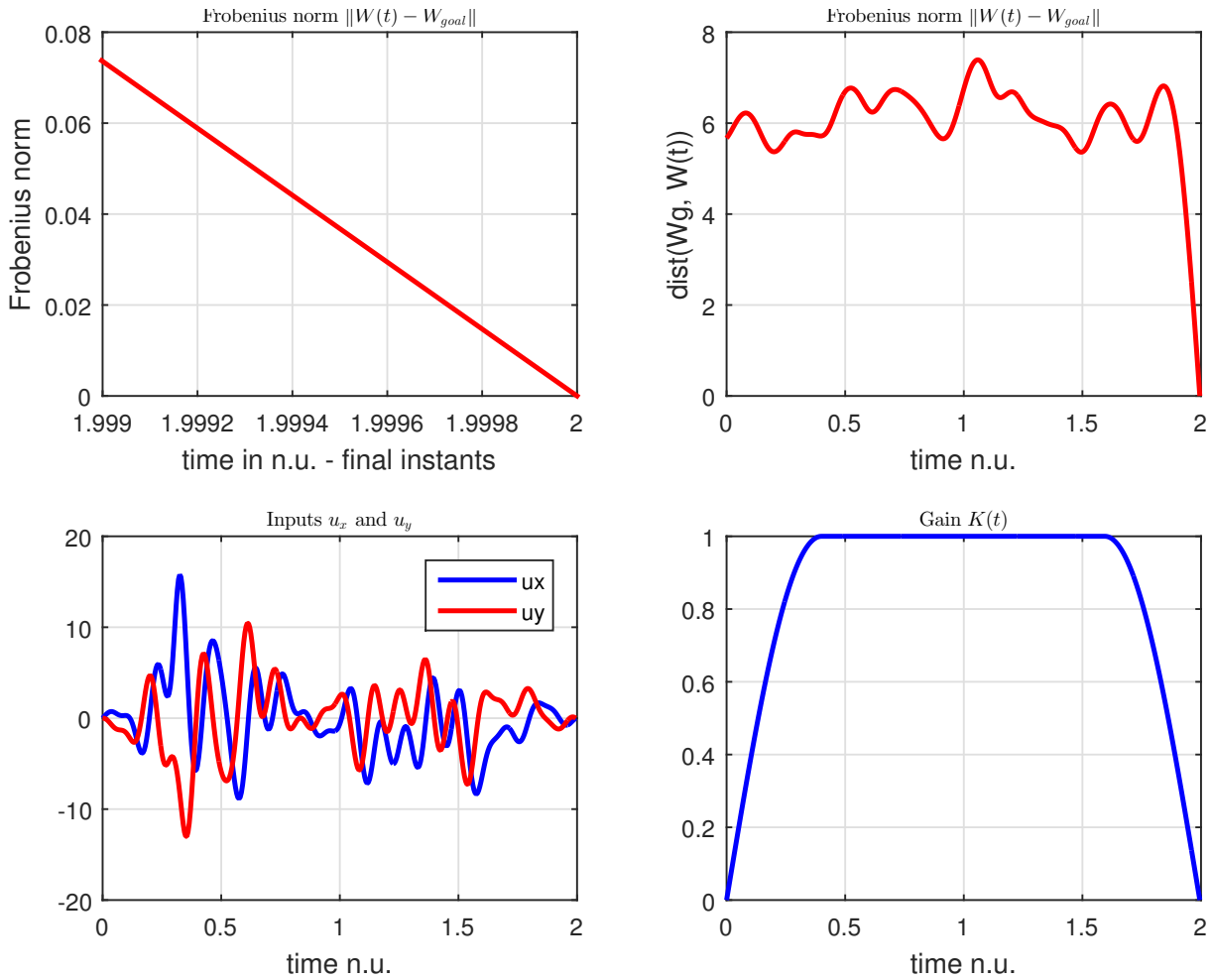
- This problem may be solved, with trivial adaptations, by the techniques of the present paper for the Lie group $SU(2)^n$ and the Lyapunov function $\mathcal{L}$.
- In a first set of simulations, ten levels were regarded ($n = 0, 1, \ldots, 9$), with the SNAP gate defined by the angles $\theta_0 = 0$, $\theta_1 = \pi/2$, $\theta_2 = 0$, $\theta_3 = \pi/2$, $\ldots$, $\theta_9 = \pi/2$. A second set of simulations has considered $n = 15$.
- No input saturation was taken into account. However, one has considered time-varying gains $K_x(t) = K_y(t) = |\chi/(2.5)|K(t)$, where $K(t)$ is the function depicted in the bottom right plot of Figure 11.

- As in Example 1 one has chosen for the seed the two simulations (for once and for all) random (uniform distributed) values of $(\mathbf{a}, \mathbf{b})$ in the interval $[-A_m, A_m]$ with $Am = 1/100000$.
- The other parameters are $\alpha = 0.070$ (corresponding to $p = 14$ different goal matrices for the switching policy) and $\beta = 2\alpha$, $\omega_r = \omega_q$.
- One defines the Frobenius norm $\|W(t) - W_{goal}\| = \sqrt{\sum_{k=0}^{n} \|W_k(t) - W_{goal,k}\|^2}$, which is a notion of distance of the actual propagator from the goal propagator.

The Table 2 summarizes the obtained results for ten levels of the cavity. Note that T_f is the final time in n.u., where 1 n.u. is equal to $2\pi/\chi$ seconds, N_1 is the number of steps of the RIGA, $\epsilon = \|W(T_f) - W_{goal}\|$, and the run time of each numerical experiment is given in seconds for a standard PC with Windows 10®. The worst case fidelity $\mathcal{F}$ is computed regarding the original propagator which is a $2n \times 2n$ matrix, with $n = 10$.

T_f	N_1	Error ϵ	run time (s)	$1 - \mathcal{F}$
2.0	300	0.00113	413.0	1.07×10^{-7}
2.0	600	3.37×10^{-6}	826.0	5.58×10^{-13}

The Figure 11 depicts the simulation results for $T_f = 2n.u.$, where 1 n.u. is equal to $2\pi/\chi$ seconds, and $N_1 = 600$. Note that the feedback gains are modulated by the function $K(t)$ of the subfigure at the bottom left position.



The Table 3 summarizes the obtained results for fifteen levels of the cavity. Note that T_f is the final time in n.u., where 1 n.u. is equal to $2\pi/\chi$ seconds, N_1 is the number of steps of the RIGA, $\epsilon = \|W(T_f) - W_{goal}\|$, and the run time of each numerical experiment is given in seconds for a standard PC with Windows 10®. The worst case fidelity $\mathcal{F}$ is computed regarding the original propagator which is a $2n \times 2n$ matrix, with $n = 15$.

T_f	N_1	Error ϵ	run time (s)	$1 - \mathcal{F}$
2.0	1200	0.022	2733.0	3.38×10^{-5}
2.0	2400	0.0026	7399.0	4.79×10^{-7}

Benchmark for a comparison between GRAPE and RIGA

- This third set of numerical experiments is a benchmark proposed in [Leung et al., 2017, section D] for testing a implementation of GRAPE. The goal is to generate a Hadamard gate.
- The system is a coupled N-qubits system

$$H(t) = J_0 \left[\sum_{s=1}^{N-1} \sigma_z^{(s)} \sigma_z^{(s+1)} \right] + J \left[\sum_{k=1}^{N} u_{x_k}(t) \sigma_x^{(k)} + u_{y_k}(t) \sigma_y^{(k)} \right] + J_g u_g(t) I, \tag{19}$$

where $\sigma_x, \sigma_y, \sigma_z$ are the Pauli matrices.

Benchmark for a comparison between GRAPE and RIGA

- The system (19) is considered with $J_0 = J = J_g = 2\pi 100$ MHz.
- The inputs are considered to be bounded with $\|u_k(t)\| \leq 5$ for $k = 1, \ldots, m$. 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 $\mathcal{I} = 1 - \|\frac{1}{n}\text{trace}X_{goal}^{\dagger}X(T_f)\|^2$. The RIGA is performed until $\mathcal{I}$ is less than 0.001.

- The value of $N_{sim} = 10N$ is considered in Leung et al. [2017] for GRAPE.
- The value of N_{sim} and the feedback gains K ($f_k = K$ for all k) used by RIGA are given in Table 2.
- The RIGA does not work well when N_{sim} is too low, which means that the quality of the numerical integration is poor.
- For instance, for $N = 9$ and $N = 10$, the convergence was too slow for $N_{sim} = 10N$. So we have used $N_{sim} = 15N$, which has presented good results.

- Random matrices $\widetilde{a}, \widetilde{b}$ in $\mathbb{R}^{11\times 20}$ with entries in $[-1, 1]$ were generated for once and for all, to be used in all numerical experiments. Then a_1, b_1 are respectively the submatrices $\widetilde{a}(1:m, 1:M)$ and $\widetilde{b}(1:m, 1:M)$ (in MATLAB notation).
- The values of M are given in table 2. The chosen amplitudes for the seed (24) are $A_m = 2/M$, and $a = A_m a_1$, $b = a_m b_1$ and $T = \pi T_f$.
- The policy of choosing $\overline{X}_{goal}$ in each step ℓ that was considered is the one of (13), which has presented better results than the one described in Algorithm C. The input saturation function that was used is the smooth version that is described in the paper.

N	iterations	run time (s)	N_{sim}	M	K	infidelity $\mathcal{I}$
2	13	1.68	10 N	10	10/J	0.0007904
3	20	1.85	10 N	10	10/J	0.0009176
4	58	5.10	10 N	11	2/J	0.0009897
5	59	8.43	10 N	10	2/J	0.0009932
6	103	69.18	10 N	14	1/J	0.000988
7	89	272.6	10 N	14	0.5/J	0.0009879
8	251	4640	10 N	14	0.25/J	0.0009982
9	232	61992	15 N	14	0.125/J	0.0009992
10	221	485336	15 N	14	0.0625/J	0.0009974

Table 2: Results of the Benchmark for comparison with GRAPE

- Our numerical experiments are done in a CPU Intel(R) Core(TM) I5 7200U 2.5 GHz with 8G bytes of RAM. The RIGA was implemented as a (non-compiled) “m-file” and executed in MATLAB® R2015a in a Windows 10 environment.
- The CPU model used by Leung et al. [2017] was an Intel Core® i7-6700K CPU 4.00 GHz and a GPU NVIDIA® Tesla® K40c and GRAPE was implemented in the *TensorFlow* library developed by Google’s machine intelligence research group Abadi et al. [2016].
- Their software cannot run in our Windows platform, and so this comparison must take into account the difference of CPU’s. The CPU benchmarks are 4612 for our CPU and 11109 for their CPU (2.4 times faster than ours).

- The software implementation of RIGA may be improved a lot, by optimizing the software and compiling the $m - file$.
- Even so, the results of the benchmark indicates that performance of the RIGA in our CPU is comparable, and even better in many cases, with ones of Leung et al. [2017] in their CPU.
- If the CPU speed is taken into account, the runtime of our results are at least 2 times faster than GRAPE. The case $N = 10$ is four times faster than GRAPE, which indicates that our performance is very good in high dimensions.

- This means that RIGA is a promising algorithm, but more tests must be done also comparing the qualities of the results in other examples.
- A disadvantage of the RIGA is the fact that its performance is dependent on the choice of the gain K.
- The authors are studying an automatic tuning policy for K.
- The performance of RIGA also depends on the parameters that determine its seed (T, M, A_m, $a = A_m a_1$ and $b = A_m b_1$.
- This last fact is natural, and the performance of GRAPE is also affected by the choice of its seed.

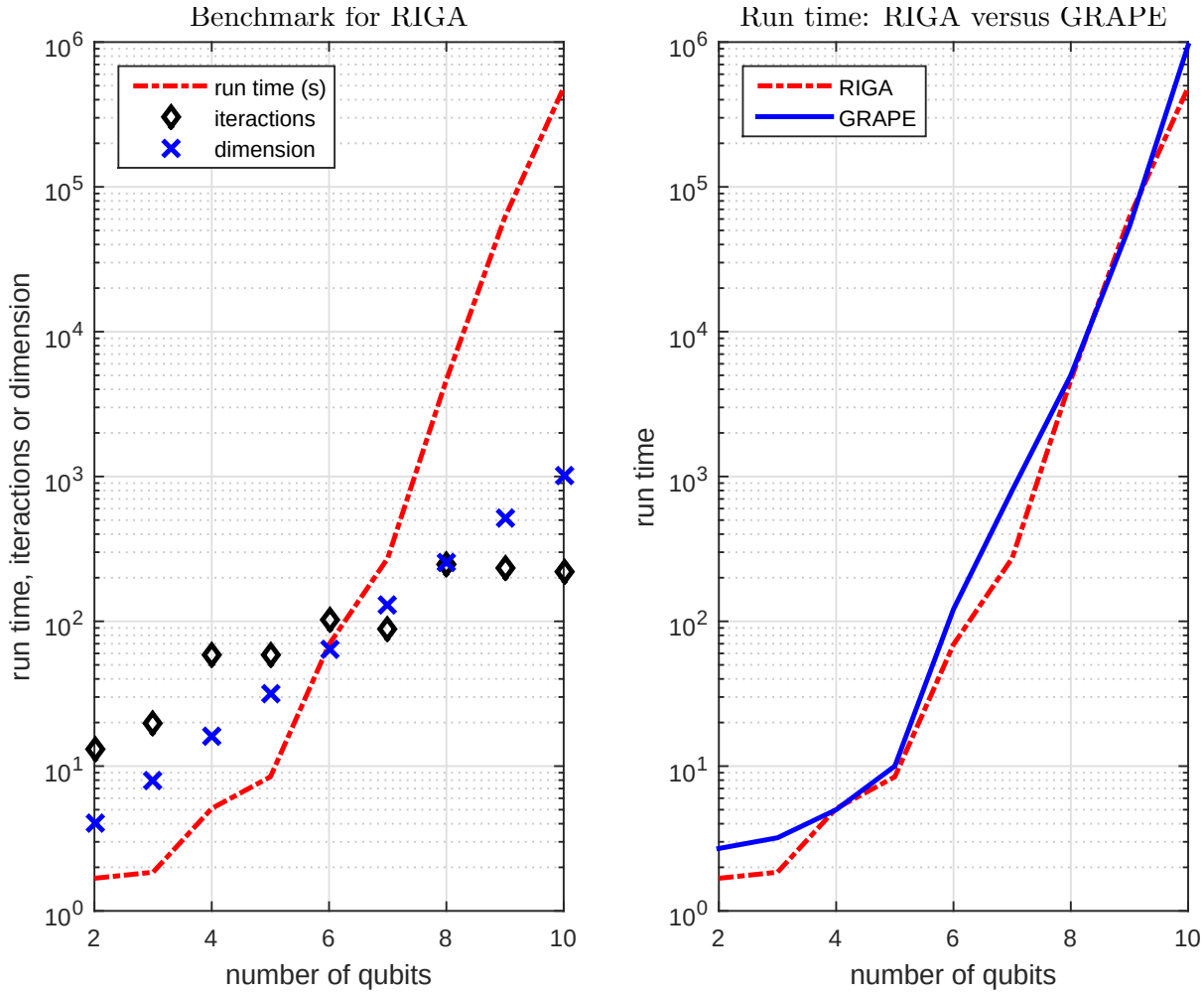


Figure: Left: Benchmark results for RIGA. Top Right: Comparison of run time between RIGA and GRAPE. A correction could be done taking into account the CPU speeds. In this case the results of RIGA could be considered to be 2.4 faster than the ones shown in this figure.

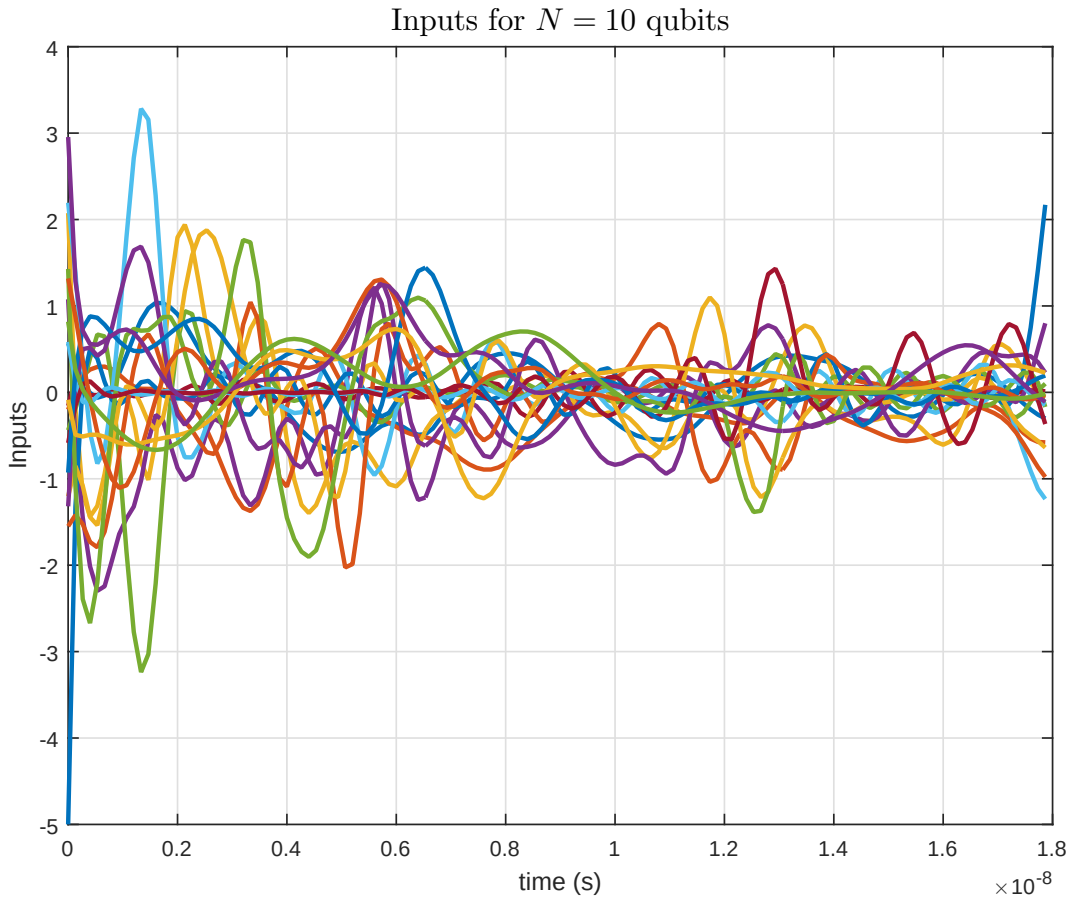


Figure: Generated inputs for $N = 10$ qubits. Note that, even with a low number of simulation points $N_{sim} = 15N$, the generated input is quite smooth.

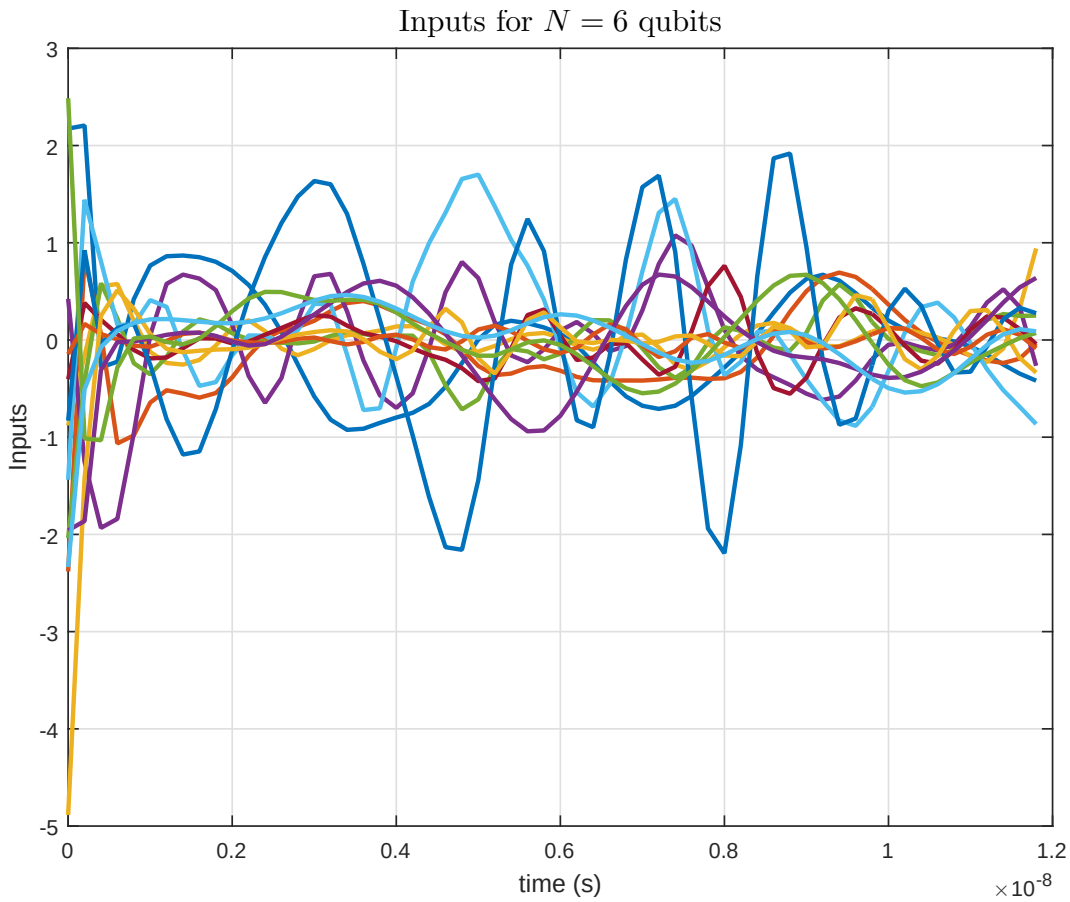


Figure: Generated inputs for $N = 6$ qubits. Note that, with this low number of simulation points $N_{sim} = 10N$, it is possible to see the effect of the discretization on the generated input.

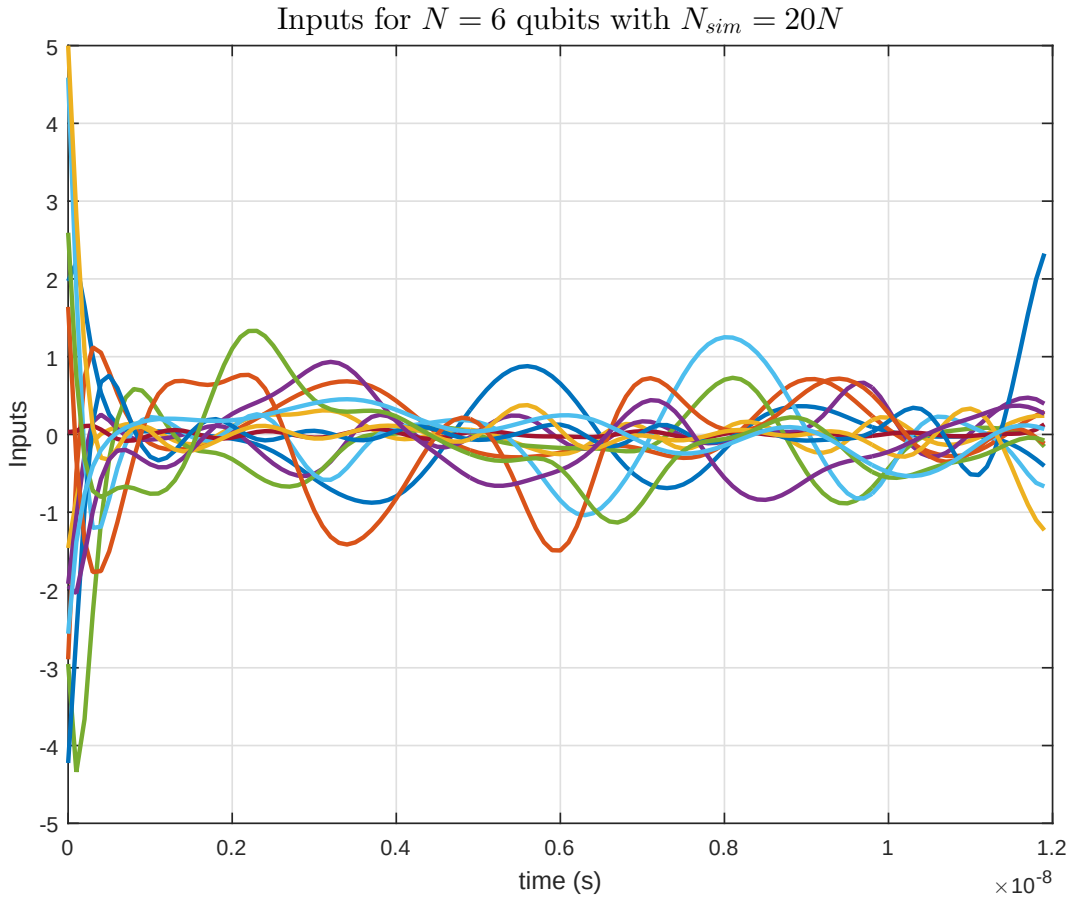


Figure: Generated inputs for $N = 6$ qubits. Now we have increased the value of N_{sim} to $N_{sim} = 20N$. The generated inputs are smoother than the ones generated with $N_{sim} = 10N$. An increase of N_{sim} also implies in a increase of runtime of each iteration.

Effect of the choice of N_{sim} in RIGA

- An increase of N_{sim} implies in "smoother" generated inputs.
- An increase of N_{sim} also implies in a increase of runtime of each iteration.
- However the RIGA may present a faster convergence with a greater value of N_{sim}, in the sense of a smaller total number of iterations, perhaps with a greater runtime.
- For instance, for $N = 6$ and $N_{sim} = 10N$, the RIGA has attained the desired fidelity of the benchmark in step $\ell = 103$.
- For the same $N = 6$ and $N_{sim} = 20N$, the desired fidelity was obtained in step $\ell = 77$.
- In the numerical implementation of RIGA, the inputs are assumed to be piecewise constant in each interval $[s\delta, (s+1)\delta)$ where $\delta = T_f/N_{sim}$ and $s = 1, 2, \ldots, N_{sim}$.
- Note that N_{sim} must be compatible with the technology for generating the inputs in real time.

Summary

- A MATLAB software was developed for implementing the RIGA. This software can be optimized for obtaining a CAD for Quantum Gate Generation that could be faster than GRAPE.
- The complexity of each step of the RIGA corresponds essentially to the simulation of the closed loop system in the interval $[0, T_f]$.
- One may include input saturations in this Algorithm, which may be useful in practical applications. A smooth version of a saturation function was tested in the benchmark, showing excellent results.
- One may include a modulation of the feedback-gains of each step, which is able to generate null inputs for $t = 0$ and for $t = T_f$ (see Example 2). This is a smoothing process of the inputs, and tends to generate a small bandwidth, which is important in practical applications.
- In the chosen examples, the RIGA was able to solve the Strong Quantum Gate Generation problem in a final time T_f (fixed a priori) and with arbitrary precision, at least when T_f is grater than the minimal time.

- In Example 1, where the minimum time T^* is known, we have verified that the RIGA has worked very well even for $T_f = T^*$.
- The ideas of the RIGA can be easily adapted to other Lyapunov functions (see Example 2).
- We have produced excellent results for the fast generation of the SNAP gate with 10 levels of the cavity of Example 2.

- As each step of our algorithm is the simulation of a closed-loop stabilized system, one obtains in the end a well behaved scheme from the numerical point of view.
- Our method has **feasible** complexity and excellent performance, even when compared with numerical schemes like GRAPE Khaneja et al. [2005].
- When compared with GRAPE, our method can cope with restrictions on the inputs without penalties in the final fidelity.
- The benchmark considered in this paper seems to demonstrate that RIGA can be superior to GRAPE, at least in this example.
- The quality of the generated inputs of RIGA and GRAPE must be compared as well. The the runtime is only a measure of software performance. A more complete comparison must be made in the future, with other examples.

- The RIGA was able to generate the Hadamard gate for the benchmark for $N = 10$ qubits with a run time that was the half of the runtime of GRAPE. If the CPU speed is taken into account, the RIGA would be 4 times faster than GRAPE.
- Another feature of our method is the fact that its convergence is guaranteed by a mathematical result.
- It is possible that our method may be combined with GRAPE in some situations and this seems to be interesting to investigate.
- The RIGA may be combined with a Fixed-point method, and this may improve its performance in some cases. Preliminary results in this direction are reported in Pereira da Silva et al. [2018].

Summary

M. Abadi, P. Barham, J. Chen, Z. Chen, A.Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, 2016. USENIX Association. ISBN 978-1-931971-33-1. URL `https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi`.

Shuang Cong. *Control of quantum systems: theory and methods*. John Wiley & Sons, 2014.

D. D'Alessandro. *Introduction to Quantum Control and Dynamics*. Chapman & Hall/CRC, Boca Raton, 2008.

D. Dong and I. R. Petersen. Quantum control theory and applications: a survey. *IET Control Theory Applications*, 4(12):2651–2671, December 2010. ISSN 1751-8644. doi: 10.1049/iet-cta.2009.0508.

Symeon Grivopoulos and Bassam Bamieh. Lyapunov-based control of quantum systems. In *42nd IEEE CDC*, volume 1, pages 434–438, 2003.

Reinier W. Heeres, Brian Vlastakis, Eric Holland, Stefan Krastanov, Victor V. Albert, Luigi Frunzio, Liang Jiang, and Robert J. Schoelkopf. Cavity state manipulation using photon-number selective phase gates. *Phys. Rev. Lett.*, 115:137002, Sep 2015. doi: 10.1103/PhysRevLett.115.137002. URL `https://link.aps.org/doi/10.1103/PhysRevLett.115.137002`.

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. *Journal of Magnetic Resonance*, 172(2):296 – 305, 2005. ISSN 1090-7807. doi: http://dx.doi.org/10.1016/j.jmr.2004.11.004. URL http://www.sciencedirect.com/science/article/pii/S1090780704003696. DOI: 10.1016/j.jmr.2004.11.004.

Nelson Leung, Mohamed Abdelhafez, Jens Koch, and David Schuster. Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Phys. Rev. A*, 95:042318, Apr 2017. doi: 10.1103/PhysRevA.95.042318. URL https://link.aps.org/doi/10.1103/PhysRevA.95.042318.

Mazyar Mirrahimi. Lyapunov control of a quantum particle in a decaying potential. In *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, volume 26, pages 1743–1765, 2009.

Mazyar Mirrahimi, Pierre Rouchon, and Gabriel Turinici. Lyapunov control of bilinear schrödinger equations. *Automatica*, 41(11): 1987–1994, 2005.

José P. Palao and Ronnie Kosloff. Quantum computing by an optimal control algorithm for unitary transformations. *Phys. Rev. Lett.*, 89 (18):188301–, October 2002. URL `http://link.aps.org/doi/10.1103/PhysRevLett.89.188301`.

José P. Palao and Ronnie Kosloff. Optimal control theory for unitary transformations. *Phys. Rev. A*, 68(6):062308–, December 2003. URL `http://link.aps.org/doi/10.1103/PhysRevA.68.062308`.

Yu Pan, V. Ugrinovskii, and M. R. James. Lyapunov analysis for coherent control of quantum systems by dissipation. In *2015 American Control Conference (ACC)*, pages 98–103, July 2015. doi: 10.1109/ACC.2015.7170718.

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*, João Pessoa, Brazil, 2018. Brazilian Control Conference.

S G Schirmer and Pierre de Fouquieres. Efficient algorithms for optimal control of quantum dynamics: the krotov method unencumbered. *New Journal of Physics*, 13(7):073029–, 2011. URL `http://stacks.iop.org/1367-2630/13/i=7/a=073029`.

H. B. Silveira, P. S. Pereira da Silva, and P. Rouchon. Quantum gate generation by *t*-sampling stabilization. *International Journal of Control*, 87(6):1227–1242, 2014.

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. *International Journal of Control*, 89(1):1–16, 2016. DOI:10.1080/00207179.2016.1161830.

Naoki Yamamoto, Koji Tsumura, and Shinji Hara. Feedback control of quantum entanglement in a two-spin system. *Automatica*, 43(6):981 – 992, 2007. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/j.automatica.2006.12.008. URL //www.sciencedirect.com/science/article/pii/S0005109807000672.

Jing Zhang, Yu-xi Liu, Re-Bing Wu, Kurt Jacobs, and Franco Nori. Quantum feedback: theory, experiments, and applications. *arXiv preprint arXiv:1407.8536*, 2014.

Summary

Obtaining the model of System 2

- One obtains the simplified model (16) of Example 2. This model evolves on in SU(2)^n.
- Choose the natural basis of $\mathbb{C}^n \otimes \mathbb{C}^2$ given by

$$\mathcal{B} = \{|k, g\rangle, |k, e\rangle : k = 0, 1, \ldots, n\},$$

  where $|k, g\rangle$ and $|k, e\rangle$ denotes respectively $|k\rangle \otimes |g\rangle$ and $|k\rangle \otimes |e\rangle$.
- By direct computation of the blocks one may show that the matrices H_0, H_1, H_2 of the model (14) are block diagonal and its diagonal is composed by n square blocks of 2-dimensional complex matrices.

- We show now that these matrices are bock diagonal.
- Let $\{e_1, e_2\}$ be the canonical basis of $\mathbb{C}^2$ and let $f_k^1 = |k, g\rangle$ and $f_k^2 = |k, e\rangle$. Then the $\{lk\}$-block of H_p can be computed by the expression $\langle e_i|H_{p\,lk}|e_j\rangle = \langle f_l^i|H_p|f_k^j\rangle, p = 0, 1, 2$.
- Simple computations shows that $H_{p\,lk} = 0$ if $l \neq k$, for $p = 0, 1, 2$. One can also show that $H_{0\,k,k} = \beta(k)I_2 + \alpha(k)E$, for $k = 0, 1, \ldots, n-1$,
- Recall that $E = |e\rangle\langle e|$, $\alpha(k) = \omega_q + \langle k|H_i|k\rangle$, and $\beta(k) = \langle k|H_c|k\rangle\,,\ k = 0, 1, \ldots, n-1$.
- Furthermore $H_{1\,k,k} = \sigma_x$ and $H_{2\,k,k} = \sigma_y$, for $k = 0, 1, \ldots, n-1$.

- One assumes that the propagator $\mathcal{X}$ is also decomposed in 2-dimensional complex blocks $X_{ij} : i, j \in \{1, \ldots, n\}$. Assume that the initial condition $\mathcal{X}(0)$ is also block-diagonal with this same structure.
- For simplicity, denote X_{kk} by X_k. Then it is easy to show that:

$$\begin{aligned}
\dot{X}_{ij}(t) &= 0, i, j \in \{0, \ldots, n-1\}, i \neq j, \\
\dot{X}_k(t) &= -\iota \left\{ \beta(k) I_2 + \alpha(k) E + \sigma_x v_x(t) + \sigma_y v_y(t) \right\} X_k(t), k \in \{0, \ldots
\end{aligned}$$

- Note that $E = \frac{1}{2} \{\sigma_z + I_2\}$. Hence the relevant part of the dynamics reads

$$\dot{X}_k(t) = -\iota \left\{ \frac{\alpha(k)}{2} \sigma_z + \left(\frac{\alpha(k)}{2} + \beta(k) \right) I_2 + \sigma_x v_x(t) + \sigma_y v_y(t) \right\} X_k(t),$$

- Define $\gamma(k) = \left(\frac{\alpha(k)}{2} + \beta(k)\right)$. Consider the phase change (this is not a global phase since γ depends on k):

$$Y_k(t) = \exp(\iota\gamma(k)t)X_k(t).$$

- Then it is easy to prove that

$$\dot{Y}_k(t) = -\iota\left\{\frac{\alpha(k)}{2}\sigma_z + \sigma_x v_x(t) + \sigma_y v_y(t)\right\} Y_k(t), k \in \{0, \ldots, n-1\}.$$

- Fix an angular frequency ω_r and consider the rotating coordinate change:

$$W_k(t) = \exp\left(\frac{\iota\omega_r t}{2}\sigma_z\right) Y_k(t).$$

- Then one shows that

$$\dot{W}_k(t) = -\iota\left\{\frac{(\alpha(k) - \omega_r)}{2}\sigma_z + \widehat{\sigma}_x(t)v_x(t) + \widehat{\sigma}_y v_y(t)\right\} W_k(t), k \in \{0, \ldots$$

- Where$\widehat{\sigma}_x(t) = \exp\left(\frac{\iota\omega_r t}{2}\sigma_z\right)\sigma_x \exp\left(\frac{-\iota\omega_r t}{2}\sigma_z\right)$ and $\widehat{\sigma}_y(t) = \exp\left(\frac{\iota\omega_r t}{2}\sigma_z\right)\sigma_y \exp\left(\frac{-\iota\omega_r t}{2}\sigma_z\right)$.

- Now assume that the inputs $v_x(t)$ and $v_y(t)$ are driven by the auxiliary inputs $u_x(t)$ and $u_y(t)$ as

$$v_x(t) + \iota v_y(t) = \left[u_x(t) + \iota u_y(t)\right] \exp(\iota \omega_r t).$$

- Then substituting this expression in the last equation, and taking into account the trigonometric identities and commutation relations of the Pauli matrices one shows that

$$\begin{aligned} \widehat{\sigma}_x(t) v_x(t) &= \frac{u_x}{2}\sigma_x + \frac{u_y}{2}\sigma_y + G_x(t), \\ \widehat{\sigma}_y(t) v_y(t) &= \frac{u_x}{2}\sigma_x + \frac{u_y}{2}\sigma_y + G_y(t), \end{aligned}$$

where
$G_x(t) = \left\{ \frac{u_x \cos(2\omega_r t) - u_y \sin(2\omega_r t)}{2} \right\} \sigma_x + \left\{ \frac{-u_x \sin(2\omega_r t) - u_y \cos(2\omega_r t)}{2} \right\} \sigma_y$
and $G_y(t) = -G_x(t)$.

REMARK. This means that the expression (16) is exact (it is not a Rotating Wave Approximation).

REMARK: Note that the dynamics of $W_k(t)$ and $Y_k(t)$ evolves on SU(2). Hence, one may show that the choice of (17) is unique for assuring that the construction of the desired SNAP gate respects also the fact that $W_{goal,k}$ must be in SU(2).

Summary

Gate Fidelity

- Fidelity is a measure of distance between quantum states.
- It is originally defined for density matrices ρ, σ by $\mathcal{F}(\rho, \sigma) = \text{trace}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)$.
- For pure normalized states $\rho = |\phi\rangle\langle\phi|$ and $\sigma = |\psi\rangle\langle\psi|$ in $\mathbb{C}^n$ (in bra-ket notation, one denotes $\langle\phi| = |\phi\rangle^\dagger$)

$$\mathcal{F}(\rho, \sigma) = \|\langle\psi|\phi\rangle\|$$

.

- In particular the square of the fidelity may be interpreted as the probability of state transition.
- Assume that $X_{goal} \in \mathrm{U}(n)$ is the desired gate and $X_f = X(T_f)$ is the obtained gate. The worst case fidelity $\mathcal{F}(X_{goal}, X_f)$ is defined by

$$\mathcal{F}(X_{goal}, X_f) = \min_{|\phi\rangle \in S_{\mathbb{C}}} \left\| \langle\phi| X_{goal}^\dagger X_f |\phi\rangle \right\|,$$

where $S_{\mathbb{C}}$ is the set of normalized (unitary) vectors of $\mathbb{C}^n$.

Computing the worst-case gate fidelity

The following result may be proved by the method of Lagrange multipliers:

Theorem

Let $\widetilde{X} = X_{goal}^{\dagger} X_f$. *Assume that the Schur decomposition of* $\widetilde{X}$ *(eigenstructure in this case) is* $\widetilde{X} = U\dagger\Sigma U$, *where* $\Sigma = diag[\exp(\iota\theta_1), \ldots, (\iota\theta_n)]$ *and* $U \in U(n)$. *Let* $\{e_1, \ldots, e_n\}$ *be the canonical basis of* $\mathbb{C}^n$. *Consider the subset* $\mathcal{V} = \{v_{ij} \in \mathbb{C} \mid v_{ij} = \frac{\sqrt{2}}{2} e_i + \frac{\sqrt{2}}{2} e_j, i = 1, \ldots, n-1, j = i+1, \ldots, n\}$. *Let* $\mathcal{F}_e = \min_{v_{ij} \in \mathcal{V}} \{\| v_{ij}^{\dagger} \Sigma v_{ij} \|\}$. *Then,* $0 \leq \mathcal{F}(X_{goal}, X_f) \leq \mathcal{F}_e$. *If there exits an interval* $I = [\theta_a, \theta_b] \subset \mathbb{R}$ *with* $\theta_b - \theta_a < \pi$ *such that, for all* $i \in \{1, \ldots, n\}$, *there exists* $k_i \in \mathbb{Z}$ *such that* $\theta_i + 2k_i\pi$ *is inside* I, *then* $\mathcal{F}(X_{goal}, X_f) = \mathcal{F}_e$. *In particular, the last equality is true if* $-\pi/2 < \theta_i < \pi/2$ *for all* $i = 1, \ldots, n$.