Quantum Inf Process (2016) 15:1937–1953
DOI 10.1007/s11128-016-1242-y

Various methods of optimizing control pulses for quantum systems with decoherence

Łukasz Pawela[1] · **Przemysław Sadowski**[1]

Received: 22 July 2014 / Accepted: 5 January 2016 / Published online: 25 January 2016

Abstract We design control setting that allows the implementation of an approximation of an unitary operation of a quantum system under decoherence using various quantum system layouts and numerical algorithms. We focus our attention on the possibility of adding ancillary qubits which help to achieve a desired quantum map on the initial system. Furthermore, we use three methods of optimizing the control pulses: genetic optimization, approximate evolution method and approximate gradient method. To model the noise in the system we use the Lindblad equation. We obtain results showing that applying the control pulses to the ancilla allows one to successfully implement unitary operation on a target system in the presence of noise, which is not possible which control field applied to the system qubits.

Keywords Quantum information · Quantum computation · Control in mathematical physics

1 Introduction

One of the fundamental issues of quantum information science is the ability to manipulate the dynamics of a given complex quantum system. Since the beginning of quantum mechanics, controlling a quantum system has been an implicit goal of quantum physics, chemistry and implementations of quantum information processing [1].

✉ Przemysław Sadowski
psadowski@iitis.pl

Łukasz Pawela
lpawela@iitis.pl

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

If a given quantum system is controllable, i.e. it is possible to drive it into a previously fixed state, it is desirable to develop a control strategy to accomplish the desired control task. In the case of finite dimensional quantum systems, the criteria for controllability can be expressed in terms of Lie-algebraic concepts [2–4]. These concepts provide a mathematical tool, in the case of closed quantum systems, i.e. systems without external influences.

A widely used method for manipulating a quantum system is a coherent control strategy, where the manipulation of the quantum states is achieved by applying semi-classical potentials in a fashion that preserves quantum coherence. In the case when a system is controllable, it is a point of interest what actions must be performed to control a system most efficiently, bearing in mind limitations imposed by practical restrictions [5–9]. The always present noise in the quantum system may be considered such a constraint [10–17]. Therefore it is necessary to study methods of obtaining piecewise constant control pulses which implement the desired quantum operation on a noisy system.

It is an important question whether the system is controllable with a control applied only on a subsystem. This kind of approach is called a *local-controllability* and can be considered only in the case when the subsystems of a given system interact. Coupled spin chains or spin networks [4,18–20] may serve as examples. Local control has a practical importance in proposed quantum computer architectures, as its implementation is simpler and the effect of decoherence is reduced by decreased number of control actuators [21,22].

In this paper we study various methods of adding ancillary qubits to a given quantum system, in order to overcome the interaction with an external environment. This means, we wish to perform a time evolution on a greater system than the target one and discard the ancilla afterward. This scheme should implement a unitary transformation on the target subsystem.

2 Model of the quantum system

We test our approach on a toy model and implement the unitary operations $U_{\mathrm{NOT}} = \sigma_x \otimes \mathbb{1}$ and $U_{\mathrm{SWAP}} = \mathbb{1} \otimes \sum_{ij} |i\rangle\langle j| \otimes |j\rangle\langle i|$ on a quantum system modeled as a an isotropic Heisenberg spin-1/2 chain of a finite length N. We will study two- and three-qubit systems. The total Hamiltonian of the aforementioned quantum control system is given by

$$H(t) = H_0 + H_c(t), \tag{1}$$

where

$$H_0 = J \sum_{i=1}^{N-1} S_x^i S_x^{i+1} + S_y^i S_y^{i+1} + S_z^i S_z^{i+1} \tag{2}$$

is a drift part given by the Heisenberg Hamiltonian. The control is performed only on the ith spin and is Zeeman-like, i.e.

$$H_c(t) = h_x(t) S_x^i + h_y(t) S_y^i. \tag{3}$$

In the above S_k^i denotes k^{th} Pauli matrix acting on the spin i. Time-dependent control parameters $h_x(t)$ and $h_y(t)$ are chosen to be piecewise constant. For notational convenience, we set $\hbar = 1$ and after this rescaling frequencies and control-field amplitudes can be expressed in units of the coupling strength J, and on the other hand all times can be expressed in units of $1/J$ [23].

We model the noisy quantum system dynamics using the Markovian approximation with the master equation in the Kossakowski–Lindblad form

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\mathrm{i}[H(t), \rho] + \sum_j \gamma_j \left(L_j \rho L_j^\dagger - \frac{1}{2}\{L_j^\dagger L_j, \rho\} \right), \tag{4}$$

where L_j are the *Lindblad* operators, representing the environment influence on the system [24] and ρ is the state of the system.

The main goal of this paper is to compare various methods for optimizing control pulses $h_x(t), h_y(t)$ for the model introduced above. As we are working within the Markovian approximation of the quantum system, we need to assume the environment is fast. Strictly speaking, this means that the environment autocorrelation functions are δ-functions in time. Another way to view this approximation is that the environment has no memory.

It is a widely known fact, that coherent control in the form of pulses on the system's qubits cannot overcome decoherence. This is intuitively true, because such control results in an unitary evolution of the system and this cannot fight the non-unitary dynamics. However, if we apply the control pulses to an acilla, we are effectively creating an additional environment and as a result produce non-unitary evolution which can be viewed as a damping on the system. This evolution smooths the effect of the piecewise constant control pulses and in turn validates the singular coupling limit on the noisy environment.

In order to show that the singular coupling limit [25,26] is valid in the case of piecewise constant control, we calculate the derivative of the reduced state and show that it is continuous. We consider the state $\rho_S = \mathrm{Tr}_A(\rho)$ after tracing out the ancilla

$$\begin{aligned} \frac{\mathrm{d}\rho_S}{\mathrm{d}t} &= \mathrm{Tr}_A \left(-\mathrm{i}[H(t), \rho] + \sum_j \gamma_j (L_j \rho L_j^\dagger - \frac{1}{2}\{L_j^\dagger L_j, \rho\}) \right) \\ &= \mathrm{Tr}_A \left(-\mathrm{i}[H_0, \rho] + \sum_j \gamma_j (L_j \rho L_j^\dagger - \frac{1}{2}\{L_j^\dagger L_j, \rho\}) \right) + \mathrm{Tr}_A(-\mathrm{i}[H_c(t), \rho]). \end{aligned} \tag{5}$$

Here the index A denotes the ancilla and index S denotes the system on which we have the target operation. The part corresponding to the drift Hamiltonian and environment influence is time independent. Because the control Hamiltonian is a combination of unitary operators and acts non-trivially on the ancilla only, the time-dependent term vanishes after tracing out the ancilla,

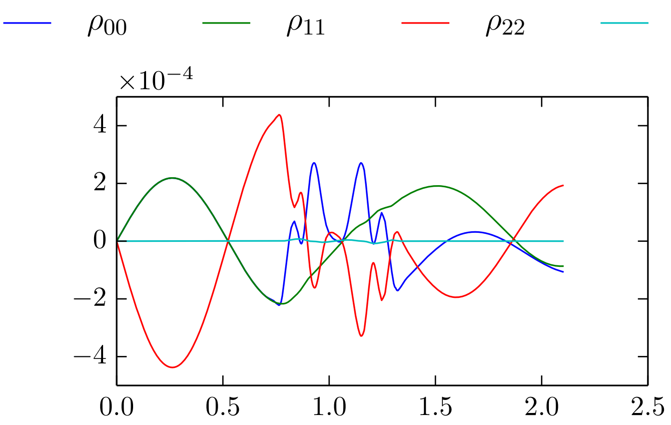

Fig. 1 An example of the derivatives of the diagonal elements of the system with the ancilla traced out

$$\begin{aligned}\mathrm{Tr}_A(-\mathrm{i}[H_c(t),\rho]) &= -\mathrm{i}\sum_{k\in\{x,y\}} h_k(t)\mathrm{Tr}_A([S_k^A,\rho])\\ &= -\mathrm{i}\sum_{k\in\{x,y\}} h_k(t)\Bigg(\sum_i(\langle i|_A\otimes\mathbb{1}_S)(S_k^A\otimes\mathbb{1}_S)\rho(|i\rangle_A\otimes\mathbb{1}_S)\\ &\quad -\sum_i(\langle i|_A S_k^A\otimes\mathbb{1}_S)\rho(S_k^A\otimes\mathbb{1}_S)(S_k^{A\dagger}|i\rangle_A\otimes\mathbb{1}_S)\Bigg)=0, \quad (6)\end{aligned}$$

where the second equation results from the fact that the partial trace result is the same for any basis selection. Thus, provided that control is applied on the ancilla register only, evolution of the rest of the system is smooth, even in the case of non-continous control pulse functions.

We show the derivatives of the diagonal elements of an example density matrix of the state of the system with the ancilla traced out. The derivatives for are shown in Fig. 1. They are obtained for the system (c) shown in Fig. 2. The damping parameter is $\gamma = 0.01$. This is an illustration of the result given in Eq. (5) showing that the derivative of the reduced system is continuous, despite the control pulses on the ancilla being piecewise constant.

In the next section we present three methods for the purpose of the comparison. The comparison of the control pulses obtained by different methods is done by applying these pulses into the above model and analysis of the obtained results.

3 Various approaches to fidelity maximization

In this Section, we describe three methods we used to optimize control pulses. The first method is based on an approximate method for obtaining a mapping which is close to a unitary one. In order to obtain the approximation, we expand the superoperator into the Taylor series up to a linear term. After this we calculate the exact gradient of the fidelity function with respect to the control pulses. The second one uses an exact formula for

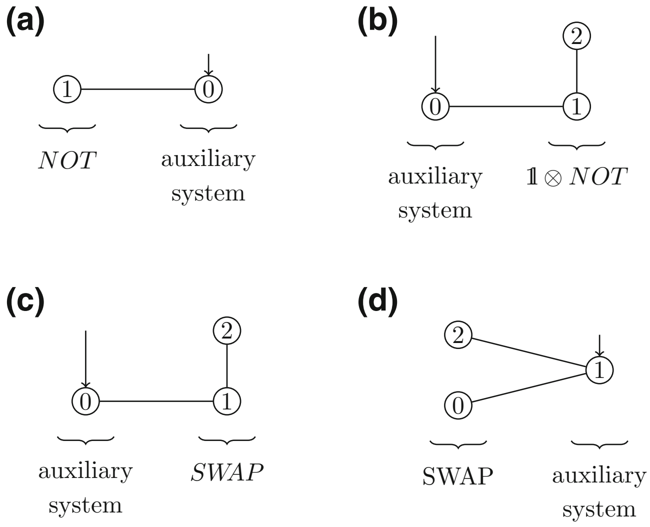

Fig. 2 Systems used for numerical simulation

the time evolution of the system, but we approximate the derivative of the mapping with respect to control pulses. In our numerical research, we perform optimization with use of the L-BFGS-B optimization algorithm [27]. Its main advantages lies in harnessing the approximation of Hessian of fitness function (we refer to the section 3 of [28] for details). Finally, we use genetic programming to optimize control pulses without the need for computing gradient of the fidelity function.

3.1 Approximate mapping method

Assuming piecewise constant control pulses the Hamiltonian in Eq. (4) becomes independent of time during the duration of the pulse. This allows us to simplify the master equation.

For notational convenience, let us write the decoherence part of the Eq. (4) in the following form [29] that allows us to act in the vectorized states space

$$-\mathcal{G} = \sum_j \gamma_j \left(L_j \otimes \overline{L_j} - \frac{1}{2} \left[\left(L_j^\dagger L_j \right) \otimes \mathbb{1} + \mathbb{1} \otimes \left(\overline{L_j^\dagger L_j} \right) \right] \right), \tag{7}$$

where $\overline{L_j}$ denotes the conjugate of L_j. The first term in Eq. 4 can be transformed into

$$-\mathrm{i}\mathcal{H} = -\mathrm{i}\left(\mathbb{1} \otimes \overline{H} - H \otimes \mathbb{1} \right). \tag{8}$$

These observations allow us to write a mapping representing the evolution of a system in an initial state ρ under Eq. (4) for time t as

$$\mathcal{S} = \exp(-t\mathcal{F}), \tag{9}$$

where $-\mathcal{F} = -\mathcal{G} - \mathrm{i}\mathcal{H}$. The final state of the evolution is

$$\mathrm{res}(\rho_f) = \mathcal{S}\mathrm{res}(\rho), \tag{10}$$

where $\mathrm{res}(\cdot)$ is a linear mapping defined for dyadic operators as

$$\mathrm{res}(|\phi\rangle\langle\psi|) = |\phi\rangle\overline{|\psi\rangle}. \tag{11}$$

The extension to all other operators follows from linearity.

We can approximate the superoperator A as

$$\begin{aligned}\mathcal{S} = \exp&\left[-\frac{1}{2}t\sum_j \left(L_j^\dagger L_j\right)\otimes \mathbb{1} + \mathbb{1}\otimes\left(\overline{L_j^\dagger L_j}\right)\right]\\ &\times\exp\left(t\sum_j L_j\otimes\overline{L_j}\right)\times\exp\left(-t\mathrm{i}\mathcal{H}\right) + O(t^2)\\ =\,& \mathcal{A}(t)\mathcal{B}(t)\mathcal{C}(t) + O(t^2).\end{aligned} \tag{12}$$

Note that, only the $\mathcal{C}(t)$ term depends on the control pulses. Assuming piecewise constant control pulses, total evolution time T and setting n as the total number of pulses, we can write the resulting approximate superoperator as

$$\tilde{\mathcal{S}} = \prod_{k=1}^{n}\mathcal{A}(\Delta t_k)\mathcal{B}(\Delta t_k)\mathcal{C}(\Delta t_k), \tag{13}$$

where Δt_k is the length of the kth time interval. As we assume all the control pulses are of the same length, we will write t_k instead of Δt_k.

The derivative of the superoperator with respect to a control pulse $h_j(t_l)$ is

$$\begin{aligned}\frac{\partial\tilde{\mathcal{S}}}{\partial h_j(t_l)} =& \left(\prod_{k=1}^{l-1}\mathcal{A}(t_k)\mathcal{B}(t_k)\mathcal{C}(t_k)\right)\times\left(\mathcal{A}(t_l)\mathcal{B}(t_l)\frac{\partial\mathcal{C}(t_l)}{\partial h_j(t_l)}\right)\\ &\times\left(\prod_{k=l+1}^{n}\mathcal{A}(t_k)\mathcal{B}(t_k)\mathcal{C}(t_k)\right).\end{aligned} \tag{14}$$

We use the fidelity as the figure of merit

$$F = \frac{1}{2^{2N}}\Re(\mathrm{Tr}\mathcal{S}_{\mathrm{target}}^\dagger\tilde{\mathcal{S}}), \tag{15}$$

where N is the number of qubits in the system and $\mathcal{S}_{\mathrm{target}}$ is the target superoperator. The derivative of the fidelity with respect to the control pulse in the jth direction and in the lth time interval is given by:

$$\frac{\partial F}{\partial h_j(t_l)} = \frac{1}{2^{2N}} \Re \left(\mathrm{Tr} \left(\mathcal{S}_{\text{target}} \frac{\partial \tilde{\mathcal{S}}}{\partial h_j(t_l)} \right) \right). \tag{16}$$

3.2 Approximate gradient method

In this section we follow the results by Machnes et al. [30]. In order to introduce the approximate gradient method, we introduce the following notation

$$\hat{H}(\cdot) = [H(t), \cdot],\ \hat{H}_i(\cdot) = [H_0 + H_i(t), \cdot],\ \hat{L}(\cdot) = \sum_j \gamma_j (L_j \cdot L_j^\dagger - \frac{1}{2}\{L_j^\dagger L_j, \cdot\}). \tag{17}$$

This allows us to write Eq. (4) in the form

$$\frac{\partial \rho(t)}{\partial t} = -(\mathrm{i}\hat{H} + \hat{L})(\rho(t)). \tag{18}$$

The evolution of a quantum map X under this equation is given by

$$\frac{\partial X(t)}{\partial t} = -(\mathrm{i}\hat{H} + \hat{L})X(t). \tag{19}$$

In order to perform numerical simulations, Eq. (19) needs to be discretized. Given a total evolution time T, we divide it into n small intervals, each of length $\Delta t = T/n$. Hence, the quantum map in the kth time interval is given by

$$X_k = \exp\left[-\Delta t(\mathrm{i}\hat{H}(t_k) + \hat{L}(t_k))\right]. \tag{20}$$

We utilize the trace fidelity as the figure of merit for this optimization problem

$$F = \frac{1}{2^{2N}} \Re \mathrm{Tr}\left[X_{\text{target}}^\dagger X(T)\right] = \frac{1}{2^{2N}} \Re \mathrm{Tr}\left[\Lambda^\dagger(t_k) X_k X(t_{k-1})\right], \tag{21}$$

where operators $X(t_{k-1}) = X_{k-1}X_{k-2}\dots X_1 X_0$ and $\Lambda^\dagger(t_k) = X^\dagger X_n X_{n-1}\dots X_{k+2}X_{k+1}$ are introduced in order to split the time evolution at any given time k and highlight the term only depending on $h_j(t_k)$, $j \in \{x, y\}$. This allows us to write the derivative of the figure of merit with respect to the control pulses as

$$\frac{\partial F}{\partial h_j(t_k)} = \frac{1}{2^{2N}} \Re \mathrm{Tr}\left[\Lambda^\dagger(t_k) \left(\frac{\partial X_k}{\partial h_j(t_k)}\right) X(t_{k-1})\right]. \tag{22}$$

Since $\hat{L}$ and $\mathrm{i}\hat{H}$ need not commute, we cannot calculate the derivative $\frac{\partial X_k}{\partial h_j(t_k)}$ using exact methods. The best approach is to use the following approximation for the gradient (see [30], section III.B)

$$\frac{\partial X_k}{\partial h_j(t_k)} \approx -\Delta t \left(\mathrm{i}\hat{H}_i + \frac{\partial \hat{L}(h_j(t_k))}{\partial h_j(t_k)} \right) X_k. \tag{23}$$

In our test cases, the damping does not depend on the control pulses; hence, we may rewrite Eq. (23) as:

$$\frac{\partial X_k}{\partial h_j(t_k)} \approx -\mathrm{i}\Delta t \hat{H}_i X_k. \tag{24}$$

This approximation is valid provided that

$$\Delta t \ll \frac{1}{||\mathrm{i}\hat{H} + \hat{L}||_2}. \tag{25}$$

3.3 Genetic programming

Genetic programming (GP) is a numerical method based on the evolutionary mechanisms [31,32]. There are two main reasons for using GP for finding optimal control pulses. First of all it enables us to perform numerical optimization in the case of a complicated fitness function. On the other hand, one should note that the values of control pulses in different time intervals can be set independently. Thus the idea of genetic code fits well as a model for a control setting. Thanks to such a representation, genetic programming enables us to exchange values of control pulses in some fixed intervals between control settings that result in most accurate approximation of the desired evolution.

Genetic programming belongs to the family of search heuristics inspired by the mechanism of natural evolution. Each element of a search space being candidate for a solution is identified with a representative of a population. Every member of a population has its unique genetic code, which is its representation in optimization algorithm. In most of the cases, genetic code is a sequence of values from a fixed set Σ of possible values of all the features that characterize a potential solution $x \in \Sigma^n$ in the search space. Searching for the optimal solution is done by the systematic modification and evaluation of genetic codes of population members due to the rules of the evolution such as mutation, selection, crossing-over and inheritance.

Mutators are functions that change single elements of a genetic code randomly. A basic example of a mutator is a function that randomly changes values of a representative x at all positions with some non-zero probability

$$M(x)_i = \begin{cases} x_i, & \text{probability } p, \\ \mathrm{rand}(\Sigma), & \text{probability } 1 - p. \end{cases} \tag{26}$$

Crossovers implement the mechanism of inheritance. This function divides given parental genetic code and creates a new genetic code. Commonly two new codes are created at the same time from two parental codes. An example of such crossover is so-called two point cut, where both parental codes (x_i, y_i) are cut into three regions and the middle segments are interchanged

$$x_i' = \begin{cases} x_i, \; i \leq c_1 \vee c_2 \leq i, \\ y_i, \quad c_1 < i < c_2, \end{cases} \; y_i' = \begin{cases} x_i, \quad c_1 < i < c_2, \\ y_i, \; i \leq c_1 \vee c_2 \leq i, \end{cases} \tag{27}$$

where $c_1 < c_2$ are randomly chosen indices. In every iteration of the algorithm, all members of the population are evaluated using fitness function $f : \Sigma^n \to \Re$ which enables elements ordering. Then, using a selector function, the set of the best members is obtained and used to create a new generation of the population using mutation and crossover functions. There is a number of strategies for defining selector function—from completely random choices to the deterministic choice of best representatives.

Strategy based on evolution mechanism makes genetic programming especially useful when parts of genetic code represent features of elements of a search space that can be interchanged between elements independently. In such case GA is expected to find the features that occur in well-fitted representatives and mix them in order to find the best possible combination. Pseudocode representing this approach is presented in Listing 1.

```
population = RandomPopulation()
for( generationsNumber ){
    newPopulation = []
    for(i = 0; i<population.size()/2; i++){
        mom = Selector(population)
        dad = Selector(population)
        (sister, brother) = CrossOver(mom, dad)
        Mutator(sister)
        Mutator(brother)
        newPopulation.append(sister)
        newPopulation.append(brother)
    }
    population = newPopulation
}
```

Listing 1: Pseudocode representing the algorithm of genetic programming. Functions `Selector`, `Mutator` and `CrossOver` work as defined in Sect. 3.3.

While the customization of population representation and fitness function unavoidably relies on the optimization problem, other parameters of genetic programming such as crossover and mutation methods are universal.

In this case the search space is the space of control pulse sequences Σ^n for n time intervals, where each control pulse has bounded absolute value $\Sigma = [-100, 100]$. In order to optimize a controlled evolution of a system governed by the Lindblad equation, we perform optimization of the average distance between target state operator and the resulting state for each basis matrix of the space of input states. For each basis matrix ρ_0^i in the space of the system joined with ancilla, we compare the reduced resulting state $\mathrm{Tr}_A(\rho_T^i)$ with the target one $\rho_{\mathrm{target}}^i = U_{\mathrm{target}} \mathrm{Tr}_A(\rho_0^i) U_{\mathrm{target}}^\dagger$. Our fitness function is defined as

$$F = \frac{1}{2^{2N}} \sum_{i=1}^{2^{2N}} \mathrm{Tr}\left(\rho^i_{\mathrm{target}} \mathrm{Tr}_A\left(\rho^i_T\right)\right), \tag{28}$$

where Tr_A denotes tracing out the ancilla and N is the total number of qubits in the system.

4 Results and discussion

Our goal in this section is to study the impact of different spin chain configurations on the final fidelity of the operation. To find the best spin chain configuration, we study the following systems:

(a) One-qubit system with one-qubit ancilla. The control is performed on the ancillary qubit, and the target is a NOT operation on the system qubit.
(b) Two-qubit system with one-qubit ancilla. The control is performed on the ancillary qubit, and the target is a $\mathbb{1} \otimes NOT$ operation on the system qubits
(c) Two-qubit system with one-qubit ancilla. The control is performed on the ancillary qubit, and the target is a $SWAP$ operation on the system qubits

In all of the simulations, we set the number of control pulses to 32 for two-qubit systems and 128 for the three-qubit systems. We limit the strength of the control pulses to $h_{\max} = 100$. We set $J = 1$. In each case we split all of the qubits forming the system into two subsystems: the one that performs some fixed evolution and the auxiliary one. Each data point is the result of 72 simulations, and we chose the result with the highest fidelity. The initial control pulses were random vectors with elements sampled independently from a uniform distribution on $[-100, 100]$.

Table 1 Number of time steps and corresponding Δt for different simulation setups in the two-qubit scenario

	Genetic optimization		Approximate evolution		Approximate gradient	
	n	Δt (10^{-3})	n	Δt (10^{-3})	n	Δt (10^{-3})
Equal number of steps	32	65.625	32	65.625	32	65.625
Equal computation time	32	65.625	128	16.406	128	16.406

Table 2 Number of time steps and corresponding Δt for different simulation setups in the three-qubit scenario

	Genetic optimization		Approximate evolution		Approximate gradient	
	n	Δt (10^{-3})	n	Δt (10^{-3})	n	Δt (10^{-3})
Equal number of steps	128	16.406	128	16.406	128	16.406
Equal computation time	128	16.406	512	4.102	512	4.102

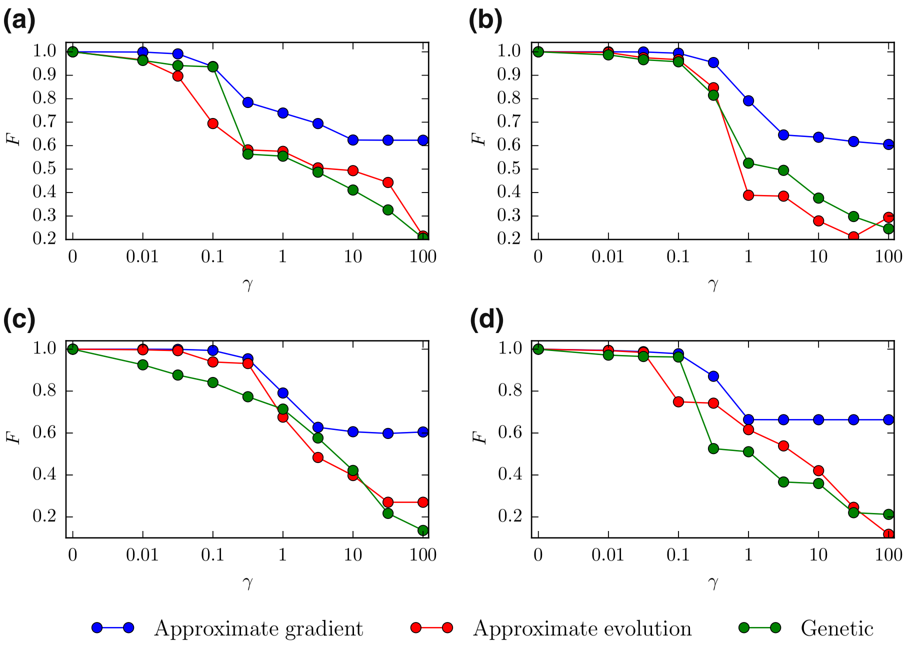

Fig. 3 Simulation results for the phase damping channel, **a** 2-qubit system, target NOT gate, **b** 3-qubit system, target $\mathbb{1} \otimes NOT$ gate, **c** 3-qubit, target $SWAP$ gate, **d** 3-qubit, target $SWAP$ gate between first and last qubit

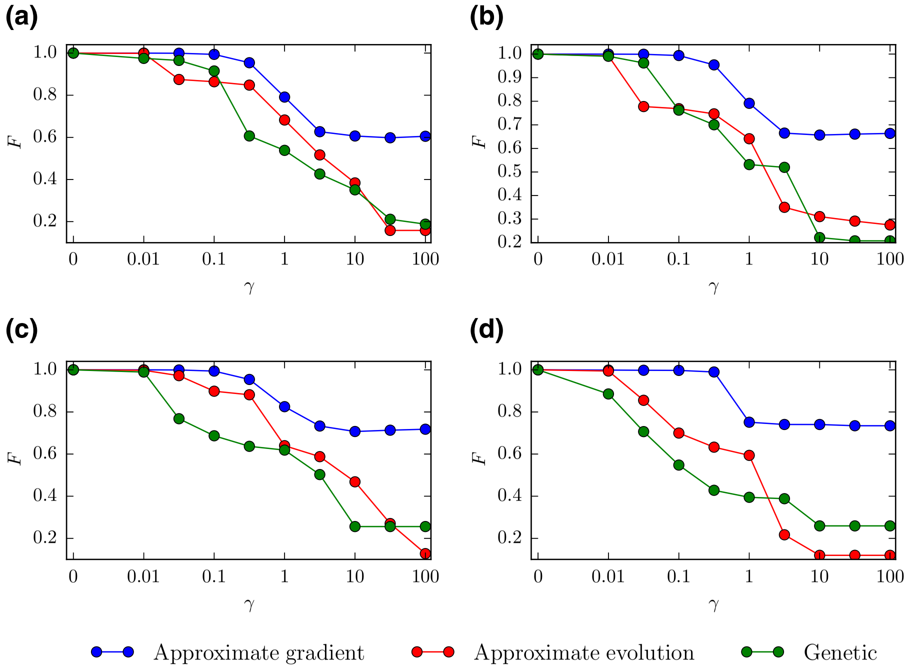

Fig. 4 Simulation results for the amplitude damping channel, **a** 2-qubit system, target NOT gate, **b** 3-qubit system, target $\mathbb{1} \otimes NOT$ gate, **c** 3-qubit, target $SWAP$ gate, **d** 3-qubit, target $SWAP$ gate between first and last qubit

Table 3 Average computation times in seconds for different simulation setups

	Genetic optimization		Approximate evolution		Approximate gradient	
Number of qubits	2	3	2	3	2	3
Equal number of steps	766	15741	521	1931	687	1745
Equal computation time	4248	24531	3875	21478	4026	22167

In our study we choose the phase damping and the amplitude damping channels as noise models. The former is given by the Lindblad operator $L = \sigma_z$, the latter is given by the operator $L = \sigma_- = |0\rangle\langle 1|$. In order to objectively compare the control methods, we study two setups: with equal number of steps of the algorithm and with equal computation time. The number of steps and length of these intervals are shown in Table 1 and Table 2.

Figure 3 shows the results for the phase damping channel. In the case of the NOT target gate, shown in Fig. 3a, b, we obtain the best results performing optimization with the approximate gradient method. In this setup the control is performed on the ancillary qubit, which leads to a smoother evolution of the target qubit, as the interaction between the two-qubits smooths impact of the control fields. Another feature of the result is the fact that the fidelity of the optimized operation decreases significantly when $\gamma \approx J$.

In the case of the SWAP target gate, shown in Fig. 3c, d, we get similar results. This is consistent with the results for the two qubit systems. In this case the genetic optimization and approximate evolution perform poorer compared to the approximate gradient method.

Next, in Fig. 4 we show the results for the amplitude damping channel. These results are similar to the phase damping case. Again there is a drop in the fidelity of the operations when $\gamma \approx J$.

Finally, we focus on comparing the algorithms when the computation times are on the same order of magnitude. To achieve this, we added more control pulses in the gradient-based methods. In the case of both gradient-based methods, we used 128 pulses for the two-qubit scenario and 512 for the three-qubit scenario. The computation times are summarized in Table 3. Results are presented in Figs 5 and 6. The obtained results show that using the method based on approximate gradient of the fitness function, we get control pulses providing the most accurate evolution.

5 Conclusions

We studied various methods of obtaining piecewise constant control pulses that implement an unitary evolution on a system governed by Kossakowski–Lindblad equation in the restrictive singular coupling limit. The studied methods included genetic optimization and the L-BFGS-B algorithm with the use of fidelity gradient based on an approximate evolution of the quantum system and an approximate gradient method for the exact evolution case.

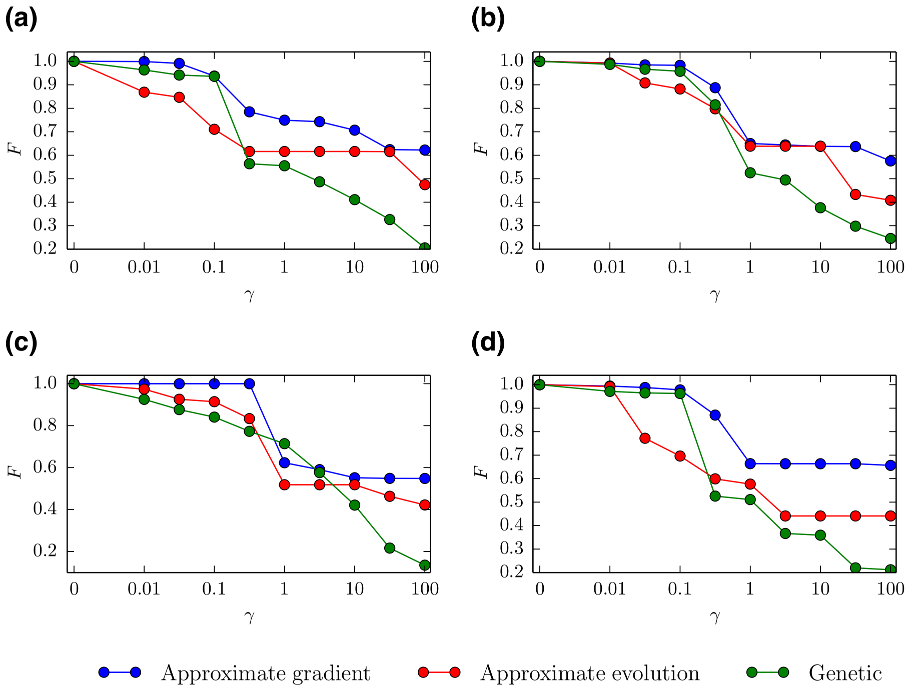

Fig. 5 Simulation results for the phase damping channel with equal computation time, **a** 2-qubit system, target NOT gate, **b** 3-qubit system, target $\mathbb{1} \otimes NOT$ gate, **c** 3-qubit, target $SWAP$ gate, **d** 3-qubit, target $SWAP$ gate between first and last qubit

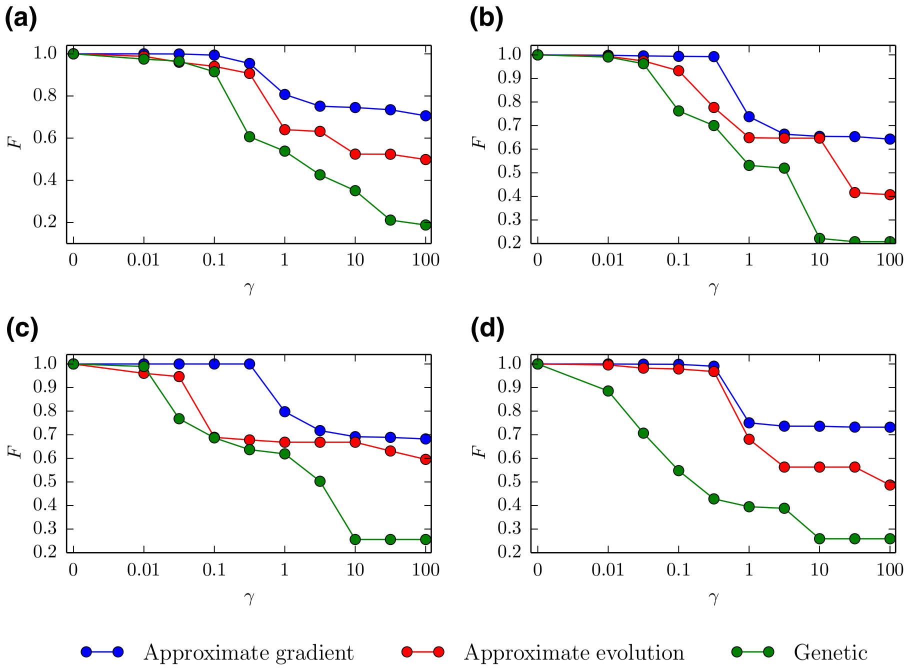

Fig. 6 Simulation results for the amplitude damping channel with equal computation time, **a** 2-qubit system, target NOT gate, **b** 3-qubit system, target $\mathbb{1} \otimes NOT$ gate, **c** 3-qubit, target $SWAP$ gate, **d** 3-qubit, target $SWAP$ gate between first and last qubit

Our results show that, by adding an ancilla, it is possible to implement a unitary evolution on a system under the Markovian approximation. What we have found is that it is sufficient to apply the control pulses to the ancillary qubits. This is caused by the fact that the interaction between the two qubits smooths impact of the control fields. This is limited to the case when the environment is slower than the frequency of oscillation of the qubits. In other words we require that $\gamma < J$.

The comparison of the numerical optimization methods used for control design led to the conclusion that using approximate gradient-based approach one gets the best results in terms of fidelity of the target unitary and the operator resulting from obtained control pulses.

Acknowledgments We would like to thank an anonymous reviewer for his/hers insightful comments. ŁP was supported by the Polish National Science Centre under the Grant Number DEC-2012/05/N/ST7/01105. PS was supported by the Polish National Science Centre under the Grant Number N N514 513340.

References

1. Cheng, C.-J., Hwang, C.-C., Liao, T.-L., Chou, G.-L.: Optimal control of quantum systems: a projection approach. J. Phys. A Math. Gen. **38**, 929 (2005)
2. Albertini, F., D'Alessandro, D.: The Lie algebra structure and controllability of spin systems. Linear Algebra Appl. **350**, 213 (2002). ISSN 0024-3795
3. Elliott, D.: Bilinear Control Systems: Matrices in Action. Springer, Berlin (2009). ISBN 1402096127
4. d'Alessandro, D.: Introduction to quantum control and dynamics. Chapman & Hal, London (2008)
5. Morzhin, O.V.: Nonlocal improvement of controlling functions and parameters in nonlinear dynamical systems. Autom. Remote Control **73**, 1822 (2012)
6. Dominy, J.M., Paz-Silva, G.A., Rezakhani, A., Lidar, D.: Analysis of the quantum Zeno effect for quantum control and computation. J. Phys. A Math. Theor. **46**, 075306 (2013)
7. Qi, B.: A two-step strategy for stabilizing control of quantum systems with uncertainties. Automatica **49**, 834 (2013)
8. Pawela, Ł., Puchała, Z.: Quantum control with spectral constraints. Quantum Inf. Process. **13**, 227 (2014)
9. Pawela, Ł., Puchała, Z.: Quantum control robust with respect to coupling with an external environment. Quantum Inf. Process. **14**, 437 (2015)
10. Li, J.-S., Ruths, J., Stefanatos, D.: A pseudospectral method for optimal control of open quantum systems. J. Chem. Phys. **131**, 164110 (2009)
11. Khodjasteh, K., Lidar, D.A., Viola, L.: Arbitrarily accurate dynamical control in open quantum systems. Phys. Rev. Lett. **104**, 090501 (2010)
12. Schmidt, R., Negretti, A., Ankerhold, J., Calarco, T., Stockburger, J.T.: Optimal control of open quantum systems: cooperative effects of driving and dissipation. Phys. Rev. Lett. **107**, 130404 (2011)
13. Gawron, P., Kurzyk, D., Pawela, Ł.: Decoherence effects in the quantum qubit flip game using Markovian approximation. Quantum Inf. Process. **13**, 665 (2014)
14. Schulte-Herbrüggen, T., Spörl, A., Khaneja, N., Glaser, S.: Optimal control for generating quantum gates in open dissipative systems. J. Phys. B At. Mol. Opt. Phys. **44**, 154013 (2011)
15. Hwang, B., Goan, H.-S.: Optimal control for non-Markovian open quantum systems. Phys. Rev. A **85**, 032321 (2012)
16. Clausen, J., Bensky, G., Kurizki, G.: Task-optimized control of open quantum systems. Phys. Rev. A **85**, 052105 (2012)

17. Floether, F.F., de Fouquieres, P., Schirmer, S.G.: Robust quantum gates for open systems via optimal control: Markovian versus non-Markovian dynamics. New J. Phys. **14**, 073023 (2012)
18. Burgarth, D., Giovannetti, V.: Full control by locally induced relaxation. Phys. Rev. Lett. **99**, 100501 (2007)
19. Burgarth, D., Bose, S., Bruder, C., Giovannetti, V.: Local controllability of quantum networks. Phys. Rev. A **79**, 60305 (2009). ISSN 1094–1622
20. Puchała, Z.: Local controllability of quantum systems. Quantum Inf. Process. **12**, 459 (2013)
21. Montangero, S., Calarco, T., Fazio, R.: Robust optimal quantum gates for Josephson charge qubits. Phys. Rev. Lett. **99**, 170501 (2007)
22. Fisher, R., Helmer, F., Glaser, S.J., Marquardt, F., Schulte-Herbrüggen, T.: Optimal control of circuit quantum electrodynamics in one and two dimensions. Phys. Rev. B **81**, 085328 (2010)
23. Heule, R., Bruder, C., Burgarth, D., Stojanović, V.M.: Local quantum control of Heisenberg spin chains. Phys. Rev. A **82**, 052333 (2010)
24. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000). ISBN 9780521635035
25. Davies, E.B.: Markovian master equations. Commun. Math. Phys. **39**, 91 (1974)
26. DAlessandro, D., Jonckheere, E., Romano, R. : in 21st International Symposium on the Mathematical Theory of Networks and Systems (MTNS) (2014), pp. 1677–1684
27. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J., Trans, A.C.M.: Algorithm 778: L-BFGS-B: fortran subroutines for large-scale bound-constrained optimization. Math. Softw. **23**, 550 (1997). ISSN 0098–3500
28. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM J. Sci. Comput. **16**, 1190 (1995)
29. Havel, T.F.: Robust procedures for converting among Lindblad, Kraus and matrix representations of quantum dynamical semigroups. J. Math. Phys. **44**, 534 (2003)
30. Machnes, S., Sander, U., Glaser, S.J., de Fouquires, P., Gruslys, A., Schirmer, S., Schulte-Herbrggen, T.: Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework. Phys. Rev. A **84**, 022305 (2011)
31. Spector, L.: Automatic Quantum Computer Programming: A Genetic Programming Approach. Springer, Berlin (2004)
32. Gepp, A., Stocks, P.: A review of procedures to evolve quantum algorithms. Genet. Program. Evol. Mach. **10**, 181–228 (2009)