MOX-Report No. 81/2023

A SPIRED code for the reconstruction of spin distribution

Buchwald, S.; Ciaramella, G; Salomon, J.; Sugny, D.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

A SPIRED code for the reconstruction of spin distribution

Simon Buchwald[a], Gabriele Ciaramella[b], Julien Salomon[c], Dominique Sugny[d]

[a] *Department of Mathematics, Universität Konstanz, Universitätsstr. 10, Konstanz, 78464, Germany*
[b] *MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, 20133, Italy*
[c] *INRIA Paris, ANGE team, 2 rue Simone Iff, Paris, 75589, France*
[d] *Laboratoire Interdisciplinaire Carnot de Bourgogne (ICB), UMR 6303 CNRS–Université de Bourgogne, 9 Av. A. Savary, B.P. 47 870, Dijon Cedex, F-21078, France*

Abstract

In Nuclear Magnetic Resonance (NMR), it is of crucial importance to have an accurate knowledge of the sample probability distribution corresponding to inhomogeneities of the magnetic fields. An accurate identification of the sample distribution requires a set of experimental data that is sufficiently rich to extract all fundamental information. These data depend strongly on the control fields (and their number) used experimentally. In this work, we present and analyze a greedy reconstruction algorithm, and provide the corresponding SPIRED code, for the computation of a set of control functions allowing the generation of data that are appropriate for the accurate reconstruction of a sample distribution. In particular, the focus is on NMR and the Bloch system with inhomogeneities in the magnetic fields in all spatial directions. Numerical examples illustrate this general study.

Keywords: Quantum control, Greedy reconstruction algorithm, spin distribution, Nuclear Magnetic Resonance

PROGRAM SUMMARY

Program Title: `SPIRED`
CPC Library link to program files: (to be added by Technical Editor)
Code Ocean capsule: (to be added by Technical Editor)
Licensing provisions(please choose one): CC0 1.0/CC By 4.0/MIT/Apache-2.0/BSD

3-clause/BSD 2-clause/GPLv3/GPLv2/LGPL/CC BY NC 3.0/MPL-2.0
Programming language: MATLAB
Nature of problem: Identify the sample probability distribution corresponding to inhomogeneties of the magnetic field in Nuclear Magnetic Resonance from experimental data. The data depends strongly on the control fields and their number, and needs to be sufficiently rich in order to extract all fundamental information.
Solution method: Use greedy reconstruction algorithms to compute a set of control functions that allows the generation of data that are appropriate for an accurate reconstruction of the sample distribution.
Additional comments including restrictions and unusual features: Some routines in the `SPIRED` use MATLAB's *fmincon*-solver, which requires MATLAB's Optimization Toolbox to be installed.

1. Introduction

Quantum Control (QC) is nowadays a well-recognized area of research [1, 2, 3, 4, 5] with many applications ranging from magnetic resonance [6, 7, 8] and atomic and molecular physics [9, 10, 11, 12] to quantum technologies [7, 13, 14]. Its goal is generally to design external control fields to perform quantum operations on the studied system. A severe limitation of QC comes from measurement processes which are much more difficult to account for than their classical counterpart. This explains that a majority of QC protocols are performed in an open-loop framework without any feedback from the experiment when applying the control. A good agreement between theory and experiment is achieved if all the parameters of the model system are perfectly known within a given range of precision. The values of such parameters can be estimated experimentally but can also be actively found by using specifically adapted controls. To this aim, different approaches using quantum features have been developed recently with success [15, 16, 17]. Among others, we can mention inversion techniques [18], selective controls [19, 20, 21, 22], the maximization of quantum Fischer information [23, 24, 25, 26, 27] and the fingerprinting approach [28, 29]. Such methods allow one to estimate the value of the Hamiltonian parameter as well as its variation range. However, this latter is not the only interesting quantity and the probability distribution is also a key feature of the experimental sample. When controlling an ensemble of quantum systems, this distribution can be interpreted as the number of individual systems having a given value of the parameter. The distribution can have a simple form such as a Gaussian or a Lorentzian one. In this case, the identification is quite straightforward and can be done using standard techniques. However, the identification is much more difficult when the distribution has a complex structure with, e.g., several peaks.

In a previous work [30], we introduced a Greedy Reconstruction Algorithm (GRA) to identify in a systematic way the probability distribution of one given Hamiltonian parameter. This was based on the framework presented in [31, 18]. In particular, we focused on an ensemble of spin 1/2 particles in Nuclear Magnetic Resonance (NMR) subjected to an inhomogeneous radio-frequency magnetic field [32, 6, 33, 8, 34, 35], where the algorithm was successfully applied to identify the distribution of the scaling factor corresponding to the sample inhomogeneity. Notice that a convergence analysis was only briefly sketched in [30], without rigorous proof. The goal of the present paper is to extend the work [30] from different points of view. First, we extend the GRA for the reconstruction of joint distributions of two distinct inhomogeneous Hamiltonian parameters. Second, we provide full MATLAB codes implementing our GRA and its optimized version (called OGRA) to find spin distribution. Such codes can be directly used to solve the problems presented in [30] and those investigated in this study. Third, we take also the opportunity of this paper to prove theoretical results covering also the ones only stated in [30]. As a result, this paper not only considers a more general problem than the one presented in [30], but also provides a full MATLAB code and detailed and rigorous convergence analysis.

The paper is organized as follows. The identification problem of the spin distribution in NMR is presented in Sec. 2. The different variants of the greedy reconstruction algorithm are described in Sec. 3. Section 4 is dedicated to the description of the structure of the code `SPIRED` and its use. A convergence analysis of the algorithm is provided in Sec. 5. Numerical results are presented in Sec. 6. Conclusion and prospective views are drawn in Sec. 7. Additional results are presented in Appendix A.

2. Identification of spin distribution

The framework of our `SPIRED` code is illustrated in a standard control problem in NMR, i.e. a spin ensemble subjected to inhomogeneous radio-frequency magnetic fields [32, 36, 8]. In a given rotating frame, each isochromat is characterized by a Bloch vector $\mathbf{M} = [M_x, M_y, M_z]^\top$, evolving in time according to the equations

$$\begin{cases} \dot{M}_x = -\omega M_y + (1+\alpha)\omega_y M_z, \\ \dot{M}_y = \omega M_x - (1+\alpha)\omega_x M_z, \\ \dot{M}_z = (1+\alpha)\omega_x M_y - (1+\alpha)\omega_y M_x. \end{cases}$$

Notice that the components of M satisfy $M_x^2 + M_y^2 + M_z^2 = M_0^2$, with M_0 the equilibrium magnetization. Here, ω_x and ω_y are time-dependent controls corresponding to the components of the magnetic field along the x- and y- directions. The parameters ω and α correspond to offset and control field inhomogeneities, respectively [6]. In standard experiments, we have $\frac{\omega}{2\pi} \in [-20, 20]$ Hz and $\alpha \in [-0.2, 0.2]$. For the purpose of this paper, we assume that the probability densities of ω and α are unknown. The controls $\frac{\omega_x}{2\pi}$ and $\frac{\omega_y}{2\pi}$ are expressed in Hz. We consider a typical field amplitude ω_0 that can be fixed, for instance, to $\omega_0 = 2\pi \times 100$ Hz. We introduce normalized coordinates as follows:

$$u_x = 2\pi\frac{\omega_x}{\omega_0};\ u_y = 2\pi\frac{\omega_y}{\omega_0};\ t' = \frac{\omega_0}{2\pi}t;\ \Delta = 2\pi\frac{\omega}{\omega_0}; \mathbf{X} = \frac{\mathbf{M}}{M_0}.$$

In what follows, we omit the prime to simplify the notations. We deduce that the differential system can be expressed in normalized units as:

$$\begin{cases} \dot{x} = -\Delta y + (1+\alpha)u_y z \\ \dot{y} = \Delta x - (1+\alpha)u_x z \\ \dot{z} = (1+\alpha)u_x y - (1+\alpha)u_y x \end{cases} \tag{1}$$

with $x^2 + y^2 + z^2 = 1$. The initial state of the dynamics for each spin is the thermal equilibrium point, i.e. $\mathbf{X}_0 = [0, 0, 1]^\top$. We consider a control time of the order of 100 ms, that corresponds to a normalized time t'_f of the order of 10. The range of variation of the parameter Δ is $\Delta_0 + 2\pi[-0.2, 0.2]$, where Δ_0 is a frequency value that can be used to shift arbitrarily the interval. For the purpose of this paper, we assume that $\Delta_0 \geq 0.4\pi$, meaning that $\Delta \geq 0$.

The goal of our `SPIRED` code is to estimate simultaneously the distributions for the parameters α and Δ by designing specific controls (u_x, u_y). We consider an ensemble of N spins whose dynamics are governed by Eq. (1). We assume that the control amplitudes (u_x, u_y) belong to the admissible set $\mathcal{U} = \{(u_x, u_y) \in \mathbb{R}^2 \mid |u_x| \leq u_m, |u_y| \leq u_m\}$, where u_m is the maximum amplitude of each component. A simple way to proceed can be described as follows. We consider that the system of N spins is divided into K_Δ groups, and we associate with the ℓ-th subgroup a certain value Δ_ℓ and the corresponding probability $P_\star^\Delta(\ell) = \frac{N_{\Delta,\ell}}{N}$, $\ell = 1, \dots, K_\Delta$, with $\sum_{\ell=1}^{K_\Delta} P_\star^\Delta(\ell) = 1$. The probability $P_\star^\Delta(\ell)$ is unknown, which means that the number of elements $N_{\Delta,\ell}$ of each group is to be found. Similarly, for the parameter α, we have K_α groups with the probabilities $P_\star^\alpha(\ell) = \frac{N_{\alpha,\ell}}{N}$, $\ell = 1, \dots, K_\alpha$, with $\sum_{\ell=1}^{K_\alpha} P_\star^\alpha(\ell) = 1$ to estimate.

This problem can be viewed as a natural extension of the work [30] and leads to the identification of two independent discrete distributions. However, this approach has two main drawbacks. First, the two random variables Δ and α are assumed to be independent. This is a limitation when trying to reconstruct the two unknown distributions, since any possible correlation is a priori neglected. Second, the final identification problem is nonlinear, since the product of the two distributions would appear. This is in contrast with the case of the reconstruction of one single distribution, where the identification problem is quadratic [30]. For these reasons, rather than considering two independent distributions, we work directly with the joint distribution, i.e. the system of N spins is divided into K groups and we associate to each subgroup a pair $(\alpha, \Delta)_\ell$ and the corresponding joint probability $P_\star(\ell) = \frac{N_\ell}{N}$, $\ell = 1, \dots, K$, with $\sum_{\ell=1}^{K} P_\star(\ell) = 1$. Now, the joint probability $P_\star(\ell)$ is unknown, namely the number of elements N_ℓ affected by the pairs $(\alpha, \Delta)_\ell$. This approach has the advantage of taking into account correlation effects and the final identification problem remains quadratic. It should be noted that these are acquired at the cost of an increase in the dimension of the unknown object(s), i.e. from two one-dimensional functions to a two-dimensional function. Finally, we point out that two independent distributions can also be treated as a specific case of joint distributions.

Since we are dealing with an inverse problem, we need to define what quantities can be observed in an experimental setting. In NMR, only the first two coordinates of the magnetization vector can be directly measured. We do not have accessed directly to the z component due to the strong constant magnetic field applied along this direction [6]. We denote by $\mathbf{Y}_{\mathbf{u},(\Delta,\alpha)}(t) = [x(t), y(t)]^\top$ the projection of the Bloch vector onto the first two coordinates. Here, the dependence on $\mathbf{u}$ and (Δ, α) has been explicitly mentioned. The corresponding experimental realization of this controlled dynamic is obtained at $t = t_f$ and leads to $\mathbf{Y}_{\mathbf{u}}^{\exp}(t_f) = [x_{\mathbf{u}}^{\exp}(t_f), y_{\mathbf{u}}^{\exp}(t_f)]^\top$, where $\mathbf{Y}_{\mathbf{u}}^{\exp}(t_f)$ can be viewed as the average at time t_f of the experimental measures of all the spins of the set subjected to the control $\mathbf{u}$. The coordinates $x_{\mathbf{u}}^{\text{exp}}$ and $y_{\mathbf{u}}^{\text{exp}}$ are those of this measured magnetization vector.

The relation between the theoretical description of the dynamical system to the experimental outcome can be expressed as:

$$\mathbf{Y}_{\mathbf{u}}^{\text{exp}}(t_f) = \sum_{\ell=1}^{K} P_\star(\ell)\mathbf{Y}_{\mathbf{u},(\Delta,\alpha)_\ell}(t_f), \tag{2}$$

in which the two sides of the equation crucially depend on the control $\mathbf{u}$.

In general, one control protocol is not sufficient to obtain an appropriate identification of the unknown $P_\star$, but a set of $\widetilde{K}$ control processes with $\widetilde{K}$ different control functions denoted $\mathbf{u}_k$, $k = 1, \cdots, \widetilde{K}$, needs to be used. On the basis of the experimental outputs, a straightforward way to determine $P_\star$ is to solve the following minimization problem:

$$\min_{P\in\mathbb{P}} \sum_{k=1}^{K} \|\mathbf{Y}_{\mathbf{u}_k}^{\text{exp}}(t_f) - \sum_{\ell=1}^{K} P(\ell)\mathbf{Y}_{\mathbf{u}_k,(\Delta,\alpha)_\ell}(t_f)\|^2, \tag{3}$$

where $\|\cdot\|$ denotes the standard Euclidean vector norm, and $\mathbb{P}$ is the convex and closed set of all the possible probability distributions P that satisfy $P(\ell) \geq 0$ for $1 \leq \ell \leq K$ and $\sum_{\ell=1}^{K} P(\ell) = 1$. At this point, it is clear that a key ingredient of the accuracy of the identification process rests on the choice of a set of $\widetilde{K}$ controls $\mathbf{u}_k$. The identification of the number $\widetilde{K}$ of control functions is a difficult task. The theoretical analysis presented in Sec. 5 shows that the choice $\widetilde{K} = K$ is sufficient. The GRA algorithm computes exactly $\widetilde{K} = K$ control fields. However, we will show that OGRA is capable of reducing (halving) the number $\widetilde{K}$ of control fields while guaranteeing an accurate identification.

Let us now rewrite (3) in a form that we consider in our `SPIRED` implementation. We introduce a set $\Phi := \{\phi_j\}_{j=1}^{K}$ of linearly independent functions $\phi_j : \{1, \ldots, K\} \to \mathbb{R}$ such that $\mathbb{P} \subset \text{span}(\Phi)$, where span denotes the vector space generated by the functions. Expressing P as $P(\ell) = \sum_{j=1}^{K} \beta_j\phi_j(\ell)$, the minimization problem (3) becomes:

$$\min_{\beta\in\widehat{\mathbb{R}}^K} \sum_{k=1}^{K} \|\mathbf{Y}_{\mathbf{u}_k}^{\text{exp}}(t_f) - \sum_{\ell,j=1}^{K} \beta_j\phi_j(\ell)\mathbf{Y}_{\mathbf{u}_k,(\Delta,\alpha)_\ell}(t_f)\|^2, \tag{4}$$

where the vector $\beta = (\beta_j)_{j=1}^{K}$ is taken in $\widehat{\mathbb{R}}^K$, a subset of $\mathbb{R}^K$, so that $P = \sum_j \beta_j\phi_j$ is a probability distribution.

We show in this study that GRA allows us to design a set of controls $\mathbf{u}_k$ that makes (4) solvable and well conditioned. The algorithm is composed of two steps, namely an offline and an online steps. In the offline step, GRA computes the controls $\mathbf{u}_k$. In this step, only the theoretical model is needed without any experimental input. The derived controls are used in the online step in which the different magnetization vectors are measured and the minimization problem (3) is solved. Note that the controls are the same for any probability distribution to identify and only depend on the model system under study. Finally, we point out that in our algorithms the duration of each control pulse is considered as a variable to be optimized together with its amplitude. In particular, we assume that the controls are constant in time, i.e. $\mathbf{u}(t) \equiv \mathbf{u} \in \mathbb{R}^2$, and that we can freely choose the control time up to a fixed maximum value t_f. Since the initial state is an equilibrium point, this is equivalent to turning on the control at a time $t \geq 0$. We show in Sec. 5 that these hypotheses are sufficient for the different examples to identify the probability distributions. The generality of GRA allows one to tackle this situation in a straightforward manner.

3. Greedy reconstruction algorithms

We present in this section the GRA in its classical form and in an optimized extension called optimized GRA (OGRA).

GRA computes the controls $\mathbf{u}_k$ and the corresponding control times t_k by solving a sequence of fitting-step and discriminatory-step problems. The goal of the fitting step is to identify a defect of the system, namely a nontrivial kernel of a certain matrix W introduced below, while the discriminatory step designs a new control which is aimed to correct this discrepancy and to eliminate the identified nontrivial kernel. The explicit formulation of GRA is presented in Alg. 1 and is given in terms of the function $\mathbf{h}^{(k)}$ defined by:

$$\mathbf{h}^{(k)}(\beta, \mathbf{u}, t) = \sum_{\ell=1}^{K} \sum_{j=1}^{k} \beta_j \phi_j(\ell) \mathbf{Y}_{\mathbf{u},(\Delta,\alpha)_\ell}(t), \tag{5}$$

for any β in $\mathbb{R}^k$. Notice that the fitting step minimizes over the full space $\mathbb{R}^k$, meaning that $\sum_j \beta_j \phi_j$ does not have to be a probability distribution. However, this is a restrictive condition. On the contrary, it allows the algorithm to find and correct more nontrivial kernels than might be necessary.

Algorithm 1 Greedy Reconstruction Algorithm (GRA)

```
Require: A set of $K$ linearly independent functions $\Phi = \{\phi_1, \dots, \phi_K\}$.
Compute the control $\mathbf{u}_1$ and the control time $t_1$ by solving
    $\max_{\substack{\mathbf{u}\in\mathcal{U} \\ t\in[0,t_f]}} \|\mathbf{h}^{(1)}(1, \mathbf{u}, t)\|^2,$    (6)
for  $k = 1, \dots, K-1$  do
    Fitting step: Find $\beta^k = (\beta^k_j)_{j=1,\dots,k}$ that solves
        $\min_{\beta\in\mathbb{R}^k} \sum_{m=1}^{k} \|\mathbf{h}^{(K)}(\boldsymbol{e}_{k+1}, \mathbf{u}_m, t_m) - \mathbf{h}^{(k)}(\beta, \mathbf{u}_m, t_m)\|^2,$    (7)
    where $\boldsymbol{e}_{k+1}$ is the $(k+1)$-th canonical vector in $\mathbb{R}^K$.
    Discriminatory step: Find $\mathbf{u}_{k+1}$ and $t_{k+1}$ that solves
        $\max_{\substack{\mathbf{u}\in\mathcal{U} \\ t\in[0,t_f]}} \|\mathbf{h}^{(K)}(\boldsymbol{e}_{k+1}, \mathbf{u}, t) - \mathbf{h}^{(k)}(\beta^k, \mathbf{u}, t)\|^2.$    (8)
end for
```

One main characteristic of GRA is that the set Φ and its order have to be fixed a-priori. However, the choice and order of Φ can have a crucial impact on the outcome of the algorithm as shown in [31, Sec. 5.3]. Hence, the idea of OGRA, which is stated in Algorithm 2, is to make the algorithm independent of the choice and order of the set Φ. Additionally, it aims at avoiding the computation of unnecessary control functions. This is achieved by two adaptations in GRA. The first one is that in each step one does not only consider the next element (the next canonical vector $\boldsymbol{e}_{k+1}$) in the set, but all remaining elements (the canonical vectors $\boldsymbol{e}_{k+\ell}$ for all $1 \leq \ell \leq K-k$) in parallel. Hence, it is also possible to enlarge the set Φ (and thus enlarge K) by additional functions ϕ_k which do not have to be linearly independent. In order to progressively remove linearly dependent functions in the set and to avoid scaling issues, all remaining basis elements are orthonormalized against the already selected ones after each iteration. The second adaptation is the introduction of two tolerances $\mathrm{tol}_1, \mathrm{tol}_2 > 0$. The first tolerance tol_1 is used as a stopping criterion.

Algorithm 2 Optimized Greedy Reconstruction Algorithm (OGRA)

```
Require: A set of $K$ functions $\Phi = \{\phi_1, \dots, \phi_K\}$ and two tolerances $\mathrm{tol}_1 > 0$ and $\mathrm{tol}_2 > 0$.
Compute $\mathbf{u}_1$, $t_1$ and the index $\ell_1$ by solving the initialization problem
    $\max_{\ell \in \{1,\dots,K\}} \max_{\substack{\mathbf{u} \in \mathcal{U} \\ t \in [0,t_f]}} \|\mathbf{h}^{(1)}(\boldsymbol{e}_\ell, \mathbf{u}, t)\|^2,$
Swap $\phi_1$ and $\phi_{\ell_1}$ in $\Phi$, and set $k = 1$, $\widetilde{K} = 1$, and $f_{max} = \|\mathbf{h}^{(1)}(\boldsymbol{e}_\ell, \mathbf{u}_1, t_1)\|^2$.
while $k \leq K - 1$ and $f_{max} > \mathrm{tol}_1$ do
    for $\ell = k+1, \dots, K$ do
        Orthonormalize all elements $(\phi_{k+1}, \dots, \phi_K)$ with respect to $(\phi_1, \dots, \phi_k)$, remove any that are linearly dependent and update $K$ accordingly.
        Fitting step: Find $(\beta_j^\ell)_{j=1,\dots,k}$ that solve the problem
            $\min_{\beta \in \mathbb{R}^k} \sum_{m=1}^{k} \|\mathbf{h}^{(K)}(\boldsymbol{e}_{k+\ell}, \mathbf{u}_m, t_m) - \mathbf{h}^{(k)}(\beta, \mathbf{u}_m, t_m)\|^2,$
        and set $f_\ell = \sum_{m=1}^{k} \|\mathbf{h}^{(K)}(\boldsymbol{e}_{k+\ell}, \mathbf{u}_m, t_m) - \mathbf{h}^{(k)}(\beta^\ell, \mathbf{u}_m, t_m)\|^2$.
    end for
    if $\max_{\ell=k+1,\dots,K} f_\ell > \mathrm{tol}_2$ then
        Set $\ell_{k+1} = \arg\max_{\ell=k+1,\dots,K} f_\ell$.
    else
        Extended discriminatory step: Find $\mathbf{u}_{k+1}$, $t_{k+1}$ and $\ell_{k+1}$ that solve
            $\max_{\ell \in \{k+1,\dots,K\}} \max_{\substack{\mathbf{u} \in \mathcal{U} \\ t \in [0,t_f]}} \|\mathbf{h}^{(K)}(\boldsymbol{e}_{k+\ell}, \mathbf{u}, t) - \mathbf{h}^{(k)}(\beta^\ell, \mathbf{u}, t)\|^2.$
        Set $\widetilde{K} = \widetilde{K} + 1$.
    end if
    Swap $\phi_{k+1}$ and $\phi_{\ell_{k+1}}$ in $\Phi$.
    Set $f_{max} = \|\mathbf{h}^{(K)}(\boldsymbol{e}_{k+\ell_{k+1}}, \mathbf{u}_{k+1}, t_{k+1}) - \mathbf{h}^{(k)}(\beta^{\ell_{k+1}}, \mathbf{u}_{k+1}, t_{k+1})\|^2$.
    Set $k = k + 1$.
end while
```

The algorithm terminates if the function value in the initialization or any of the discriminatory steps (denoted by f_ℓ in Alg. 2) is too small, thus not adding new information. The second tolerance tol_2 is used to skip the computation of a new control field in the discriminatory step, if the minimum

GRA routines	**Description**
`main`	Main function used to run the code.
`GRA`	Greedy reconstruction algorithm.
`OGRA`	Optimized greedy reconstruction algorithm.
`discriminatory_step`	Routine that solves the initialization and discriminatory step problem using MATLAB's *fmincon*-solver.
`fitting_step`	Routine that solves the fitting step problem.
`orthonormalize`	Routine that orthonormalizes all remaining basis elements after each iteration of OGRA.
`SVD_solver`	Routine that solves the fitting step problem using the singular value decompostion (SVD).

Tab. 1: Routines related to the greedy reconstruction algorithm.

cost function value computed by the fitting step is not small enough. If this function value is large, then there already exists a control function that discriminates between the two distributions $\phi_{k+\ell}$ and $\sum_{j=1}^{\ell} \beta_j^{\ell} \phi_j$. Notice that setting tol_2 to a very small value is reasonable if the final identification problem is quadratic. In this case, one can prove that a nonzero cost function value in the fitting step implies that one does not need to compute a new control for the corresponding set element (compare with [31]). However, if the final identification problem is not quadratic, then it can make sense to set tol_2 to a larger value. In conclusion, the main adaptations of OGRA in lines 3 and 8-9 allow the algorithm to reduce the number of computed controls $\widetilde{K}$, meaning that $\widetilde{K} < K$. On the other hand, as we mentioned before, GRA is designed to always compute exactly $\widetilde{K} = K$ controls. The numerical implementation of GRA and OGRA is presented and discussed in the following sections.

4. The `SPIRED` code

4.1. Structure of the code

In this section, we provide a full list of all MATLAB functions contained in the `SPIRED` code. Inside the `SPIRED` folder the user can find the `main` routine used to run the `SPIRED` code, as well as the routines that run GRA and OGRA, and that solve their sub-problems (see Tab. 1). Additionally,

Reconstruction routines	**Description**
`generate_data`	Routine that generates the experimental realizations for all computed controls.
`reconstr`	Routine that either solves the final identification problem (4) using a second order interior-point algorithm, or the compact form (compare (9) in Section 5) using a solver based on the SVD.

Tab. 2: Routines related to the reconstruction of the probability distribution.

Test routines	**Description**
`starting`	Input function.
`fun_discriminatory`	Function computing the cost functional and gradient for the discriminatory step problem.
`NMR_solver`	Routine that solves the (normalized) dynamical system via direct calculations of the exponential matrix (compare the proof of Theorem 2).

Tab. 3: Routines related to the test problems.

the `SPIRED` folder contains the routines that generate the synthetic experimental data for the true parameter probability distribution, and that solve the final identification problem (3) (see Tab. 2). Notice that the both the `discriminatory_step` and the `reconstr` routine use MATLAB's *fmincon*-solver, which requires MATLAB's Optimization Toolbox to be installed.

There are also three subfolders labeled "Test1", "Test2" and "Test3". These contain three test problems the user can choose from. "Test1" corresponds to the problem discussed in this paper. "Test2" is the same as "Test1", but only considers a control in the x direction (in other words $\mathbf{u}_y = 0$ for all control fields). Finally, "Test3" corresponds to the problem investigated in [30], where the resonance offset Δ is fixed and one attempts to reconstruct only the control inhomogeneity parameter α. Each of these "Test" folders contains routines to set the input variables, describe the cost function and its gradient for the discriminatory-step problem, and solve the corresponding dynamical system (see Tab. 3). They also each contain two routines used to plot the results and condition number of the reconstruction process (see Tab. 4).

Plotting routines	Description
`plot_reconstr`	Routine that plots the true and reconstructed probability distributions for the two control sets.
`plot_condition`	Routine that produces a table containing the condition numbers corresponding to the reconstruction process.

Tab. 4: Routines plotting the results for the test problem.

4.2. Usage of the code

Here we illustrate the working procedure of the `SPIRED` code with an example. The user needs to define the test problem in the function `main`, which is used to initialize the procedure.

```
function [ controls, bases, model, results ] = main
% STEP 1: Choice of the Problem;
addpath('Test1');
% STEP 2: Assemble problem variables;
[ model, bases, options ] = starting ( );
% STEP 3: Run!
[controls.GRA, results.GRA] = GRA( bases.GRA, model,
    options);
[controls.OGRA, bases.OGRA, results.OGRA] = OGRA( bases.
    OGRA, model, options);
% STEP 4: Compute (synthetic) experimental data;
Y_exp.GRA  = generate_data( controls.GRA, model );
Y_exp.OGRA = generate_data( controls.OGRA, model );
% STEP 5: Solve the final identification problem;
...
```

In particular, at the “STEP 1” the user needs to define the path of the folder containing the test routines. Then, the user can define the input variables in the function `starting`, which is listed exemplary for the first test problem in the following.

```
function [ model, bases, options ] = starting ( )
% STEP 1: Input variables;
% control bounds and maximum control time;
um  = 10;
tf  = 16;
% variables for the unknown parameters
```

```
Delta0              = 4*pi;
Delta1              = 0.2;
Delta_interval      = Delta0 + 2*pi.*[-Delta1, Delta1];
alpha_interval      = [-0.2, 0.2];
% number of grid points for the unknown parameters
nr_alphas           = 10;
nr_Deltas           = 10;
% open the file to get the input probability distribution
load('Test1/Distributions/Gaussian.mat', 'P_star')
% number of spins;
nr_spins = 100000;
% options for GRA and OGRA
iterations          = nr_alphas*nr_Deltas;
Display_GRA         = 'off';
flag_orth           = 1;
% numerical parameters for OGRA;
tol_OGRA_fit        = 1e-4;
tol_OGRA_discr      = 1e-14;
% tolerance for the SVD solver in the fitting step (and
   optionally for the reconstruction solver)
tol_svd             = 1e-10;
% optimization method for the final identification problem
solver              = 'fmincon';
```

At “STEP 1” in this function, the user can define the input variables and the path to the .mat file containing the true probability distribution $P_\star$. The input parameters related to the problem are

- **um:** bound u_m for the absolute value of the control amplitudes;
- **tf:** maximum normalized control time t_f;
- **Delta0:** frequency shift Δ_0 for the normalized resonance offset interval;
- **Delta1:** width Δ_1 of the normalized resonance offset interval;
- **Delta_interval:** interval boundaries for the normalized resonance offset Δ;
- **alpha_interval:** interval boundaries for the control field inhomogeneity parameter α;
- **nr_alphas:** number of grid points in the direction of α for the joint discrete parameter probability distribution of α and Δ;

- `nr_Deltas:` number of grid points in the direction of Δ for the joint discrete parameter probability distribution of α and Δ;
- `nr_spins:` number of spins in the system;
- `iterations:` (maximum) number of iterations performed by GRA and OGRA; for any full basis of the discrete parameter space, the obvious choice is the total number of discretization points, which is the product of `nr_alphas` and `nr_Deltas`;
- `Display_GRA` Display option to print information about the current iteration of GRA and OGRA in the command window; can be set to `'off'` to display no output, `'iter'` to show the current substep of GRA and OGRA, or `'iter-detailed'` to also show the current optimization problem during the substeps of OGRA;
- `flag_orth:` flag variable that turns the orthonormalization of the remaining basis elements during OGRA on or off;
- `tol_OGRA_fit:` tolerance tol_2 for OGRA;
- `tol_OGRA_discr:` tolerance tol_1 for OGRA;
- `tol_svd:` tolerance for the SVD solver, used in the fitting step and (optionally) for the final identification problem;
- `solver:` optimization method used to solve the final identification problem (4); can be set to `'fmincon'` to solve (4) using the second-order interior-point algorithm of MATLAB's *fmincon*-solver, or to `'svd'` to solve a compact form of the problem (compare (19) in Section 6) using the SVD solver;

The .mat file has to contain the variable `P_star`, which is the vectorized true joint probability distribution $P_\star$. If the user is considering a true experimental (laboratory) setup, meaning that they perform real experiments for the different controls to obtain the experimental data and that the true probability distribution is truly unknown, they should replace "STEP 4" in the "main.m" file with a load command to fetch the real experimental data.

Finally, to run the code the user has to write on the MATLAB prompt the following

```
>> [ controls , bases , model , results ] = main
```

After the computations, the routine saves the results in the MATLAB workspace (as documented in the code) and plots the reconstructed probability

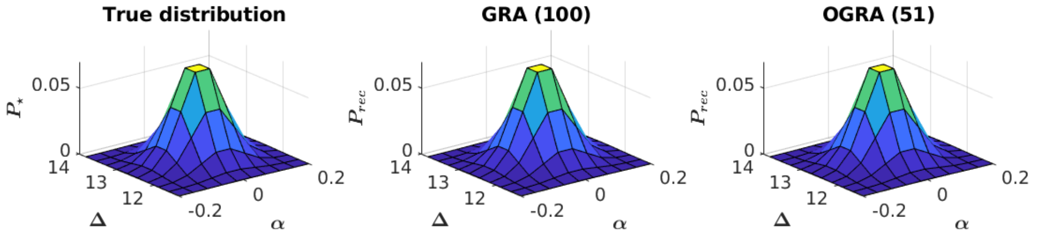

Fig. 1: The plot on the left shows the true Gaussian probability distribution for $K = 100$ uniform mesh points. The plots in the middle and on the right contain the reconstructed distributions for the control sets generated by GRA (containing 100 control fields) and OGRA (containing 51 control fields).

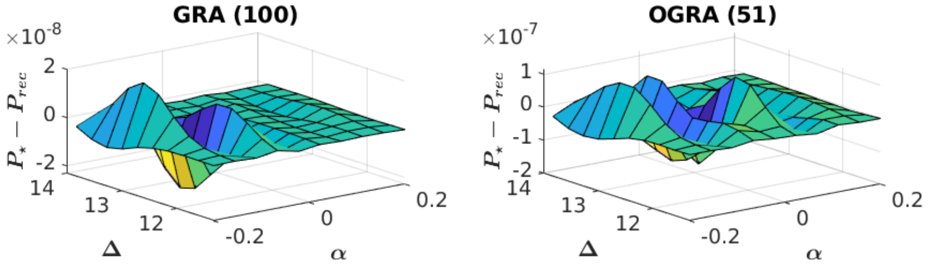

Fig. 2: Difference between the true probability distribution $P_\star$ and the distributions P_{rec} reconstructed using the control sets generated by GRA (left) and OGRA (right). In brackets are the number of control fields for each set.

distributions and their difference to the true one, as well as the condition numbers for different mesh sizes.

In particular, the results obtained by running “Test1” are the true and reconstructed probability distributions for the control fields generated by GRA and OGRA, shown in Fig. 1. In Fig. 2 we show the difference between the true and reconstructed distributions for the two control field sets. Additionally, the routine provides a table containing the exact condition numbers corresponding to GRA and OGRA. Since the solver for the discriminatory step problem is initialized with a random vector, there may be small variations in some results, without changing the overall outcome.

Examples of all figures produced by the different test problems are also provided in the folder “Results” that can be found in the corresponding “Test”-folder. There one can also find the .mat files containing the set of random controls for each test problem, loaded in the `main`.

5. Convergence analysis

In this section, we prove that the controls generated by GRA and OGRA make possible the identification of the unknown probability distributions of the parameters Δ and α, i.e. they make problem (3) uniquely solvable.

We start by recalling that problem (3) is equivalent to (4). Assuming that $P_\star$ can be written as $P_\star(\ell) = \sum_{j=1}^{K} \beta_{\star,j}\phi_j(\ell)$, we can write equation (4) in a compact form as follows:

$$\min_{\beta \in \widehat{\mathbb{R}}^K} \langle \beta_\star - \beta | W | \beta_\star - \beta \rangle, \tag{9}$$

where $W := \sum_k W(\mathbf{u}_k, t_k)$ is the sum of symmetric and positive semi-definite $K \times K$- matrices whose elements are defined as:

$$[W(\mathbf{u}_k, t_k)]_{\ell,j} := \langle \gamma_\ell(\mathbf{u}_k, t_k) | \gamma_j(\mathbf{u}_k, t_k) \rangle \tag{10}$$

with

$$\gamma_j(\mathbf{u}_k, t_k) := \sum_\ell \phi_j(\ell) \mathbf{Y}_{\mathbf{u}_k,(\alpha_\ell,\Delta_\ell)}(t_k). \tag{11}$$

Since the set of vectors β is a convex subset of $\mathbb{R}^K$, we deduce that the problem is uniquely solvable if the matrix W is positive definite. In the case W has a non-trivial kernel, infinitely many solutions may exist which lead to wrong probability distributions different from the experimental one $P_\star$. We stress that the non-triviality of the kernel depends completely on the choice of the controls $\mathbf{u}_k$ and the corresponding control times t_k.

Using the notation (10)-(11), we can now also rewrite the subproblems of GRA in terms of the matrix W. The initialization problem (6) can be written as

$$\max_{\substack{\mathbf{u} \in \mathcal{U} \\ t \in [0,t_f]}} |[W(\mathbf{u}, t)]_{1,1}|^2. \tag{12}$$

The fitting step problem (7) is equivalent to

$$\min_{\beta \in \mathbb{R}^k} \langle \mathbf{v}_\beta | [W^k]_{[1:k+1,1:k+1]} | \mathbf{v}_\beta \rangle, \tag{13}$$

where $W^k := \sum_{m=1}^{k} W(\mathbf{u}_m, t_m)$ and $\mathbf{v}_\beta := [\beta^\top, -1]^\top$. Finally, the discriminatory step problem (8) can be written as

$$\max_{\substack{\mathbf{u} \in \mathcal{U} \\ t \in [0,t_f]}} \langle \mathbf{v}_{\beta^k} | [W(\mathbf{u}, t)]_{[1:k+1,1:k+1]} | \mathbf{v}_{\beta^k} \rangle. \tag{14}$$

A direct interpretation of these reformulated problems is that each control $\mathbf{u}_k$ generated by GRA at iteration k ensures that $\langle \boldsymbol{e}_k|W|\boldsymbol{e}_k\rangle > 0$. Iteratively, this implies that $\langle \mathbf{w}|W|\mathbf{w}\rangle > 0$ for any $\mathbf{w} \in \mathbb{R}^K$ which is equivalent to W being positive definite.

In more details, the first control $\mathbf{u}_1$ and the control time t_1 are chosen by the initialization (12) such that the first upper left entry of $W(\mathbf{u}_1, t_1)$ is positive. This guarantees that $\langle \boldsymbol{e}_1|W|\boldsymbol{e}_1\rangle > 0$ since

$$\langle \boldsymbol{e}_1|W|\boldsymbol{e}_1\rangle = \sum_{j=1}^{K} \langle \boldsymbol{e}_1|W(\mathbf{u}_j, t_j)|\boldsymbol{e}_1\rangle \geq [W(\mathbf{u}_1, t_1)]_{1,1} > 0,$$

where we used that $W(\mathbf{u}, t)$ is positive semi-definite for any $\mathbf{u}$ and t. Assume now that the upper left 2×2-submatrix of $W^1 = W(\mathbf{u}_1, t_1)$ is not positive definite. Then it has a one-dimensional kernel spanned by a vector $\mathbf{v}_{\beta^1} := [\beta^1, -1]^\top \in \mathbb{R}^2$ (see [31, Lem. 5.3]). The corresponding scalar β^1 is clearly the unique solution to the fitting step problem (13) for $k = 1$. Now, the discriminatory step problem (14) attempts to find a control $\mathbf{u}_2 \in \mathcal{U}$ and a control time $t_2 \in [0, t_f]$ such that the vector $\mathbf{v}_{\beta^1}$ is not in the kernel of the upper left 2×2-submatrix of $W(\mathbf{u}_2, t_2)$. If this is successful then the upper left 2×2-submatrix of $W^2 = W(\mathbf{u}_1, t_1) + W(\mathbf{u}_2, t_2)$ is positive definite. This also implies that $\langle \boldsymbol{e}_2|W|\boldsymbol{e}_2\rangle > 0$. Repeating this procedure for $k = 2, \ldots, K-1$, we obtain $\langle \boldsymbol{e}_k|W|\boldsymbol{e}_k\rangle > 0$ for all $k \in 1, \ldots, K$, which guarantees that W is positive definite. We summarize the arguments above in the following theorem.

Theorem 1. *Let $\{(\boldsymbol{u}_k, t_k)\}_{k=1}^K$ be a set of controls and corresponding control times generated by GRA, such that $[W(\boldsymbol{u}_1, t_1)]_{1,1} > 0$. Let β^k be the solution to the fitting step problem* (13) *for $k = 1, \ldots, K-1$, such that the vectors $\boldsymbol{v}_{\beta^k} = [(\beta^k)^\top, -1]^\top$ are not in the kernel of $[W(\boldsymbol{u}_{k+1}, t_{k+1})]_{[1:k+1,1:k+1]}$. Then the matrix $W = \sum_k W(\boldsymbol{u}_k, t_k)$ is positive definite.*

It remains to show that the discriminatory step can always find a control such that the vector $\mathbf{v}_{\beta^k}$ is not in the kernel of $[W(\mathbf{u}, t)]_{[1:k+1,1:k+1]}$. In fact, it is sufficient to show that for any $k \in \{1, \ldots, K\}$ there exists a control $\mathbf{u} \in \mathcal{U}$ and a $t \in [0, t_f]$ such that $\langle \mathbf{v}_{\beta^k}|[W(\mathbf{u}, t)]_{[1:k+1,1:k+1]}|\mathbf{v}_{\beta^k}\rangle > 0$. We show in Theorem 2 that this is valid in the context of this paper.

Theorem 2. *Let $k \in \{1, \ldots, K-1\}$, $W^k_{[1:k,1:k]}$ be positive definite, β^k the solution to the fitting-step problem* (13)*, and $\boldsymbol{v}_{\beta^k} = [(\beta^k)^\top, -1]^\top$. Then any* *solution* $(\boldsymbol{u}, t)$ *to the discriminatory-step problem* (14) *satisfies*

$$\langle \boldsymbol{v}_{\beta^k} | W_{[1:k+1,1:k+1]}(\boldsymbol{u}, t) | \boldsymbol{v}_{\beta^k} \rangle = \|\boldsymbol{h}^{(K)}(\boldsymbol{e}^{k+1}, \boldsymbol{u}; t) - \boldsymbol{h}^{(k)}(\beta^k, \boldsymbol{u}; t)\|^2 > 0.$$

Proof. For brevity, we identify α with $1+\alpha$ for the remainder of this proof. We start by writing

$$\mathbf{h}^{(K)}(\boldsymbol{e}^{k+1}, \mathbf{u}, t) - \mathbf{h}^{(k)}(\beta^k, \mathbf{u}, t) = \sum_{\ell=1}^{K} \left(\phi_{k+1}(\ell) - \sum_{j=1}^{k} \beta_j^k \phi_j(\ell) \right) \mathbf{Y}_{\mathbf{u},(\Delta,\alpha)_\ell}(t).$$

Since the functions $\{\phi_1, \ldots, \phi_K\}$ are linearly independent, it holds that

$$\exists \widetilde{\ell} \in \{1, \ldots, K\} : \quad \phi_{diff}(\widetilde{\ell}) := \phi_{k+1}(\widetilde{\ell}) - \sum_{j=1}^{k} \beta_j^k \phi_j(\widetilde{\ell}) \neq 0. \tag{15}$$

According to (1), we have for any $(\alpha, \Delta)_\ell$

$$\frac{d}{dt}\mathbf{X}(t) = \Big[\Delta_\ell A + \alpha_\ell (u_x B_x + u_y B_y)\Big]\mathbf{X}(t), \quad \mathbf{X}(0) = \mathbf{X}_0, \tag{16}$$

where

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{X}_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

Now, consider the control $\widetilde{\mathbf{u}} := [0, b]^\top$ and a corresponding control time $\widetilde{t} \in [0, t_f]$, where both $b \in \mathbb{R} \setminus \{0\}$ and $\widetilde{t}$ are to be chosen later. We have $\mathbf{Y}_{\mathbf{u},(\Delta,\alpha)_\ell}(\widetilde{t}) = C\mathbf{X}(\mathbf{u}, (\alpha, \Delta)_\ell; \widetilde{t})$, where $\mathbf{X}(\mathbf{u}, (\alpha, \Delta)_\ell; \widetilde{t})$ is the solution to (16) and $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. Thus, we obtain

$$\mathbf{Y}_{\widetilde{\mathbf{u}},(\Delta,\alpha)_\ell}(\widetilde{t}) = C e^{\widetilde{t}(\Delta_\ell A + \alpha_\ell b B_y)} \mathbf{X}_0.$$

Since $\Delta_\ell A + \alpha_\ell b B_y$ is skew-symmetric, we can compute its exponential matrix explicitly. By setting $\widetilde{A} := \widetilde{t}(\Delta_\ell A + \alpha_\ell b B_y)$ and $x_\ell := \sqrt{\Delta_\ell^2 + \alpha_\ell^2 b^2}$, we have $e^{\widetilde{A}} = I_3 + \frac{\sin(\widetilde{t}x)}{\widetilde{t}x}\widetilde{A} + \frac{1-\cos(\widetilde{t}x_\ell)}{\widetilde{t}^2 x_\ell^2}\widetilde{A}^2$ (see, e.g., [37]). Since $CI_3\mathbf{X}_0 = 0$ and

$$\widetilde{A}^2 = \widetilde{t}^2 \begin{bmatrix} -\Delta_\ell^2 - \alpha_\ell b^2 & 0 & 0 \\ 0 & -\Delta_\ell^2 & \Delta_\ell \alpha_\ell b \\ 0 & \Delta_\ell \alpha_\ell b & -\alpha_\ell^2 b^2 \end{bmatrix},$$

we obtain

$$C\mathbf{X}(\widetilde{\mathbf{u}},\alpha_\ell;\widetilde{t}) = C\left(\tfrac{\sin(\widetilde{t}x_\ell)}{x_\ell}\begin{bmatrix}\alpha_\ell b\\ 0\\ 0\end{bmatrix} + \tfrac{1-\cos(\widetilde{t}x_\ell)}{x_\ell^2}\begin{bmatrix}0\\ \Delta_\ell\alpha_\ell b\\ -\alpha_\ell^2 b^2\end{bmatrix}\right) = \begin{bmatrix}\frac{\sin(\widetilde{t}x_\ell)}{x_\ell}\alpha_\ell b\\ \frac{1-\cos(\widetilde{t}x_\ell)}{x_\ell^2}\Delta_\ell\alpha_\ell b\end{bmatrix}.$$

Thus, we have

$$\mathbf{h}^{(K)}(\boldsymbol{e}^{k+1},\widetilde{\mathbf{u}},\widetilde{t}) - \mathbf{h}^{(k)}(\beta^k,\widetilde{\mathbf{u}},\widetilde{t}) = b\sum_{\ell=1}^{K}\phi_{diff}(\ell)\alpha_\ell\begin{bmatrix}\frac{\sin(\widetilde{t}x_\ell)}{x_\ell}\\ \Delta_\ell\frac{1-\cos(\widetilde{t}x_\ell)}{x_\ell^2}\end{bmatrix} =: F(\widetilde{t}).$$

Now, seeking a contradiction, assume that $\mathbf{h}^{(K)}(\boldsymbol{e}^{k+1},\widetilde{\mathbf{u}},\widetilde{t}) - \mathbf{h}^{(k)}(\beta^k,\widetilde{\mathbf{u}},\widetilde{t}) = 0$ for all $\widetilde{t}\in[0,t_f]$ and all $b\in\mathbb{R}\setminus\{0\}$. Since F is analytic in $\widetilde{t}$, we obtain $F^{(k)}(\widetilde{t}) = 0$ for all $k\in\mathbb{N}$ and all $\widetilde{t}\in[0,t_f]$. For k odd, we have

$$F^{(k)}(\widetilde{t}) = b\sum_{\ell=1}^{K}\phi_{diff}(\ell)\alpha_\ell(-1^{\frac{k-1}{2}})\begin{bmatrix}x_\ell^{k-1}\cos(Tx_\ell)\\ \Delta_\ell x_\ell^{k-2}\sin(Tx_\ell))\end{bmatrix}. \tag{17}$$

Since $F^{(k)}(T) = 0$ for all k odd, the first component of $F^{(k)}(T)$ in (17), for different k odd, implies that

$$\underbrace{\begin{bmatrix}1 & 1 & \cdots & 1\\ x_1^2 & x_2^2 & \cdots & x_K^2\\ x_1^4 & x_2^4 & \cdots & x_K^4\\ \vdots & \vdots & \vdots & \vdots\\ x_1^{\widetilde{K}} & x_2^{\widetilde{K}} & \cdots & x_K^{\widetilde{K}}\end{bmatrix}}_{=:D}\underbrace{\begin{bmatrix}\phi_{diff}(1)\alpha_1\cos(Tx_1)\\ \phi_{diff}(2)\alpha_2\cos(Tx_2)\\ \vdots\\ \phi_{diff}(K)\alpha_K\cos(Tx_K)\end{bmatrix}}_{=:\boldsymbol{\phi}_{\widetilde{t}}} = 0.$$

Notice that D is a Vandermonde matrix (see, e.g., [38]). Now, let $\widetilde{K} = 2(K-1)$, meaning that $D\in\mathbb{R}^{(\frac{\widetilde{K}}{2}+1)\times K}$ is a square matrix. Then, the determinant of D is given exactly by

$$\det(D) = \prod_{1\le i<j\le K}(x_j^2 - x_i^2).$$

This implies that two rows of D are linearly independent if and only if $|x_i|\neq|x_j|$. Hence, $\det(D_x) = \det(D_y)\neq 0$ (and therefore $\boldsymbol{\phi}_{\widetilde{t}} = 0$) if and only if $|x_i|\neq|x_j|$ for $i\neq j$. Recalling that $x_\ell = \sqrt{\Delta_\ell^2+\alpha_\ell^2 b^2}$, $|x_i|\neq|x_j|$ is equivalent to $\Delta_i^2 + \alpha_i^2 b^2 \neq \Delta_j^2 + \alpha_j^2 b^2$. For $i \neq j$ we also have $\alpha_i \neq \alpha_j$ and/or $\Delta_i \neq \Delta_j$ by definition. Since $\alpha_\ell \in [0.8, 1.2]$ and $\Delta_\ell \in \Delta_0 + 2\pi[-0.2, 0.2]$ with $\Delta_0 \geq 0.4\pi$, we obtain $\alpha_i^2 \neq \alpha_j^2$ and/or $\Delta_i^2 \neq \Delta_j^2$ for $i \neq j$. Thus, there exists $b \in \mathbb{R} \setminus 0$ such that $\Delta_i^2 + \alpha_i^2 b^2 \neq \Delta_j^2 + \alpha_j^2 b^2$ for all $i, j \in \{1, \dots, K\}$ with $i \neq j$. In conclusion, we have $|x_i| \neq |x_j|$ for $i \neq j$, which implies that $\boldsymbol{\phi}_{\widetilde{t}} = 0$ and therefore $\phi_{diff}(\ell)\alpha_\ell \cos(\widetilde{t}x_\ell) = 0$ for all $\ell \in \{0, \dots, K\}$ and all $\widetilde{t} \in [0, t_f]$. However, we also have $\phi_{diff}(\widetilde{\ell}) \neq 0$ by (15), $\alpha_{\widetilde{\ell}} > 0$ and $x_{\widetilde{\ell}} > 0$. Thus, there exists $\widetilde{t} \in [0, t_f]$ such that $\phi_{diff}(\widetilde{\ell})\alpha_{\widetilde{\ell}}\cos(\widetilde{t}x_{\widetilde{\ell}}) \neq 0$, which is a contradiction. □

Analogously to the proof of Theorem 2, one can show that any solution $(\mathbf{u}_1, t_1)$ to the initialization problem (12) satisfies $[W(\mathbf{u}_1, t_1)]_{1,1} > 0$. We conclude our analysis by the following theorem.

Theorem 3. *Let $(\boldsymbol{u}_k, t_k)$, $k = 1, \dots, K$, be a set of controls and corresponding control times generated by GRA. Then problem* (4) *is uniquely solvable by $\beta = \beta_\star$.*

Proof. Let β^k be the solution to the fitting step problem (13) for $k = 1, \dots, K-1$. By Theorem 2, the vector $\mathbf{v}_{\beta^k} = [(\beta^k)^\top, -1]^\top$ is not in the kernel of $[W(\mathbf{u}_{k+1}, t_{k+1})]_{[1:k+1,1:k+1]}$ for all $k \in \{1, \dots, K-1\}$. Thus, we obtain by Theorem 1 that the matrix $W = \sum_k W(\mathbf{u}_k, t_k)$ is positive definite. Hence, problem (9) is uniquely solvable by $\beta = \beta_\star$. By equivalency of problems (9) and (4), we obtain the result. □

Notice that, in the notation above, OGRA simply reorders rows and columns of the matrix W^k while attempting to find and correct its kernel. In fact, the second improvement in lines 8-9 in OGRA skips the discriminatory step only if there exists a row and column of W^k with index ℓ_{k+1} such that, by swapping ϕ_{k+1} and $\phi_{\ell_{k+1}}$, the matrix $W^k_{[1:k+1,1:k+1]}$ is positive definite. Thus, if tol_1 is sufficiently small, one can also prove convergence of OGRA analogously to GRA.

6. Numerical Results

We test GRA and OGRA on the setting described in Sec. 2. We choose a maximum control time of 160 ms, which corresponds to a normalized time $t_f = 16$. The shift of the parameter Δ is set to $\Delta_0 = 4\pi$ and the width of its interval to $4\pi\Delta_1$, with $\Delta_1 = 0.2$. We consider two different probability distributions $P_\star$, a simple Gaussian one (see panel on the left in Fig. 1) and a step distribution with three peaks (see panel on the left in Fig. 3). They are discretized by a uniform mesh of 100 points (10 points in each direction). Similarly, we discretize the set of linearly independent functions $\{\phi_j\}_{j=1}^K$ by setting $K = 100$ and $\phi_j = \boldsymbol{e}_j \in \mathbb{R}^{100}$ the j-th canonical vector in $\mathbb{R}^{100}$. Finally, we fix the tolerances for OGRA to be $\mathrm{tol}_1 = 10^{-14}$ and $\mathrm{tol}_2 = 10^{-4}$.

Now, let us briefly discuss how we solve the sub-steps of the algorithms numerically. The initialization and discriminatory step problems are solved by a second-order trust-region method. For the fitting step, we use the equivalent compact form (13). The corresponding first-order optimality system is given by

$$[W^k]_{[1:k,1:k]}\beta = [W^k]_{[1:k,k+1]}. \tag{18}$$

Since the matrix $[W^k]_{[1:k,1:k]}$ is symmetric and positive definite, any solution to Eq. (18) is a global solution to Eq. (13). Hence, we solve the fitting step problem by solving the linear system (18) using a solver based on the SVD. This solver first computes the SVD of $[W^k]_{[1:k,1:k]}$, i.e. two orthogonal matrices $U, V \in \mathbb{R}^{k\times k}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{k\times k}$ such that $U\Sigma V^\top = [W^k]_{[1:k,1:k]}$. To make the method more robust against numerical instabilities, it then removes all singular values that are smaller than a given tolerance, and the corresponding columns of U and V. Finally, it computes β by setting $\widetilde{\beta} = V^\top\beta$ and solving $\Sigma\widetilde{\beta} = U^\top[W^k]_{[1:k,k+1]}$.

After running the algorithms, we reconstruct $P_\star$ by solving problem (4). Notice that, using the notation (10)-(11), the gradient of the cost function in (4) is given by $W\beta - \sum_k \Gamma(\mathbf{u}_k, t_k)^\top \mathbf{Y}_{\mathbf{u}_k}^{\mathrm{exp}}(t_k)$, where the columns of Γ are given by the $\gamma_j(\mathbf{u}_k, t_k)$ defined in Eq. (11). We can also immediately see that the Hessian of the cost function in Eq. (4) is exactly W, which is guaranteed to be positive definite by our analysis in Sec. 5. Hence, the global solution to Eq. (4) is given by the (unique) solution to

$$W\beta = \sum_k \Gamma(\mathbf{u}_k, t_k)^\top \mathbf{Y}_{\mathbf{u}_k}^{\mathrm{exp}}(t_k). \tag{19}$$

However, in order to ensure that the coefficients of the computed solution correspond to a probability distribution (i.e. belong to $\widehat{\mathbb{R}}^K$), we add the necessary constraints and solve Eq. (4) with the second-order interior point algorithm of MATLAB's *fmincon*-solver. Nonetheless, the code includes an option to solve directly Eq. (19) using a SVD solver (see Section 4).

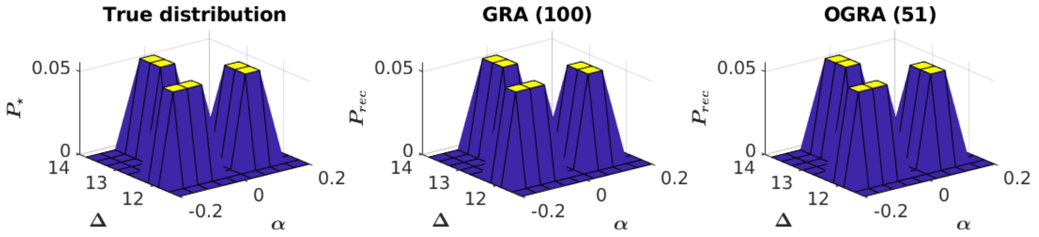

Fig. 3: Same as Fig. 1 but for a step distribution with three peaks. In brackets are the number of control fields for each set.

Now, we run both GRA and OGRA on the canonical set $\{\phi_j\}_{j=1}^{100}$ of hat functions. In contrast to [30], we do not include any additional random vectors in the canonical set for OGRA and also do not remove any elements from the set during OGRA (but still reorder them). The reason for this is that we experienced for the problem of this paper that additional random elements do not improve the results and removing elements from the canonical set does not reduce the number of controls, but is more likely to make the final identification problem numerically unstable. While GRA computes 100 controls, OGRA only designs 51 by skipping 48 discriminatory steps. We then choose $P_\star$ as the Gaussian distribution in Fig. 1 (left) and compute the corresponding experimental realizations $\{\mathbf{Y}^{\exp}_{\mathbf{u}_k}(t_k)\}_{k=1}^{\widetilde{K}}$ for the two resulting sets of control fields, with $\widetilde{K} = 100$ for GRA and $\widetilde{K} = 51$ for OGRA. Reconstructing $P_\star$ as described above, we obtain the coefficient vectors β_{rec} and thereby the distributions $P_{rec} = \sum_{j=1}^{100} \beta_{rec,j}\phi_j$ corresponding to GRA and OGRA, shown in Fig. 1. Looking at the errors with respect to the true distribution $P_\star$ shown in Fig. 2, we observe that GRA outperforms OGRA by one order of magnitude. However, the difference is so small that it is not visible in the reconstructed distributions. Similar results are obtained for a step distribution with three peaks in Fig. 3.

To investigate the dependence of the results on the choice of parameters, we repeat the experiment for different maximum control times, widths of the Δ-interval and $K = 400$ mesh points. First, we take a look at the number of control fields generated by OGRA in Tab. 5. We observe that the number of generated control fields is increasing with decreasing maximum control time and decreasing width of the Δ-interval. We also observe that the ratio between the number of GRA controls (which is equal to the number of mesh points K) and the number of OGRA controls is decreasing with an increasing

	$K = 100$				$K = 400$			
Δ_1 \ t_f	8	16	24	32	8	16	24	32
0.1	70	**52**	**50**	**50**	259	299	276	240
0.2	**58**	**51**	**50**	**50**	305	288	**223**	**220**
0.4	**56**	**50**	**50**	**50**	326	256	**211**	**200**
0.8	**51**	**50**	**50**	**50**	292	**210**	**200**	**200**
1.6	**50**	**50**	**51**	**50**	275	**205**	**200**	**200**

Tab. 5: Number of controls computed by OGRA for a control bound $u_m = 10$, and different numbers of discretization points K, maximum control times t_f and widths $4\pi\Delta_1$ of the Δ-interval. Bold numbers indicate that the number of OGRA controls is less than 60% of the number of GRA controls. Notice that GRA always generates K controls.

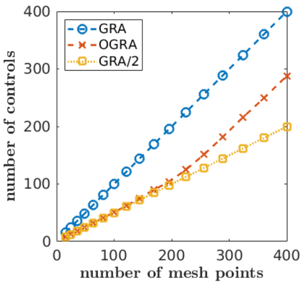

Fig. 4: Number of controls for GRA (dashed circles) and OGRA (solid crosses) for different total numbers of mesh points. To highlight the ratio between the amount of controls, we also plot half the amount of GRA controls (dotted squares).

number of mesh points. To validate this point, we plot the number of controls for both algorithms, for different total numbers of mesh points in Fig. 4.

An explanation of this behaviour is given by the condition number of the corresponding matrices W, defined in Eq. (10), representing the compact form (9) of the final identification problem. The condition numbers corresponding to GRA and OGRA for the settings in Tab. 5 are shown in Tabs. 6 and 7. Based on our theoretical results for GRA and OGRA proving that $\widetilde{K} = K$ controls are sufficient, we also add a set of fully random controls (randomized within the given bounds u_m and t_f) that has the same number of controls as GRA (i.e. $\widetilde{K} = 100$ and $\widetilde{K} = 400$, respectively). We observe that the condition number shows the same correlation with respect to the maximum control time, width of the Δ-interval and number of mesh points, as the number of OGRA controls. In particular, the condition number of

	GRA				OGRA				random control set			
t_f Δ_1	8	16	24	32	8	16	24	32	8	16	24	32
0.1	$3e16$	**5e07**	**7e03**	**6e03**	$3e15$	**3e08**	**7e05**	**7e06**	$1e18$	**1e09**	**1e06**	**3e06**
0.2	**7e09**	**4e06**	**3e03**	**1e03**	**5e09**	**1e08**	**1e08**	**1e06**	**7e12**	**1e09**	**4e04**	**9e03**
0.4	**1e11**	**5e03**	**1e03**	**1e03**	**2e11**	**3e06**	**1e07**	**1e07**	$1e16$	**7e04**	**9e03**	**1e03**
0.8	**1e06**	**2e03**	**1e03**	**1e03**	**3e07**	**8e05**	**2e05**	**2e05**	**7e10**	**3e03**	**2e03**	**1e03**
1.6	**2e04**	**9e02**	**1e03**	**8e02**	**2e06**	**4e07**	**1e05**	**7e07**	**5e09**	**1e04**	**2e03**	**1e03**

Tab. 6: Condition number of W for different control sets, maximum control times t_f and widths $4\pi\Delta_1$ of the Δ-interval. The total number of mesh points is $K = 100$ and the bound on the control is $u_m = 10$. Bold numbers indicate that the condition number is smaller than $1e15$.

	GRA				OGRA				random control set			
t_f Δ_1	8	16	24	32	8	16	24	32	8	16	24	32
0.1	$2e20$	$1e19$	$1e20$	$2e15$	$3e19$	$2e19$	$9e19$	$3e15$	$4e19$	$2e19$	$2e19$	$3e18$
0.2	$1e19$	$1e19$	**1e14**	**4e13**	$6e19$	$5e19$	**2e14**	**1e14**	$2e19$	$2e19$	$6e19$	$3e18$
0.4	$8e19$	$2e19$	**1e14**	**1e04**	$5e19$	$8e18$	**1e14**	**4e07**	$6e19$	$4e19$	$8e15$	**2e06**
0.8	$3e19$	**2e13**	**1e04**	**9e03**	$3e19$	**2e14**	**1e07**	**5e07**	$5e19$	$4e18$	**1e08**	**1e05**
1.6	$6e20$	**1e10**	**2e04**	**6e03**	$1e20$	**4e11**	**8e07**	**2e08**	$3e19$	$6e18$	**1e08**	**2e04**

Tab. 7: Same as Tab. 6 but for a total number of mesh points $K = 400$.

OGRA is below $1e15$ for all settings where OGRA computed less than 60% of the number of GRA controls.

Regarding the condition numbers, GRA and random controls show the same behaviour as OGRA. The reason can be found by taking a closer look at the entries of the matrix W. It can be shown that the difference between two adjacent rows or columns of W is bounded in norm by u_m, t_f and the mesh size for the probability distribution, i.e., $\alpha_{\ell+1} - \alpha_\ell$ and $\Delta_{\ell+1} - \Delta_\ell$. The interested reader can find more details about this result in Appendix A. We conclude that, if the control bound, the maximum control time, or the mesh size (or equivalently the width of the Δ-interval) is too small, the difference between two adjacent rows/columns of W can become numerically equal to zero, implying that W has a nontrivial kernel.

In order to investigate the impact of this numerical instability on the reconstructed results, we consider again the setting of the beginning of this

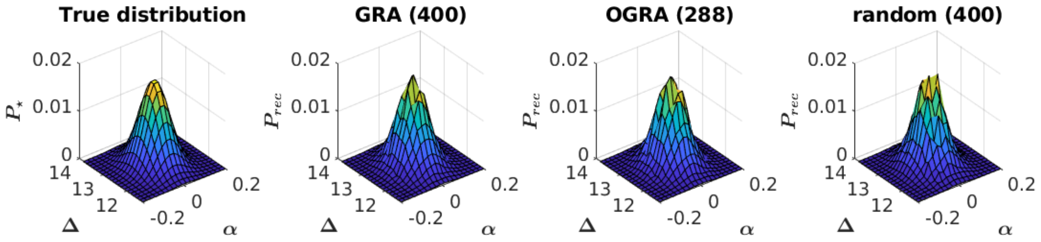

Fig. 5: Same as Fig. 1 but for $K = 400$ and including the reconstructed distribution for 400 random control fields. In brackets are the number of control fields for each set.

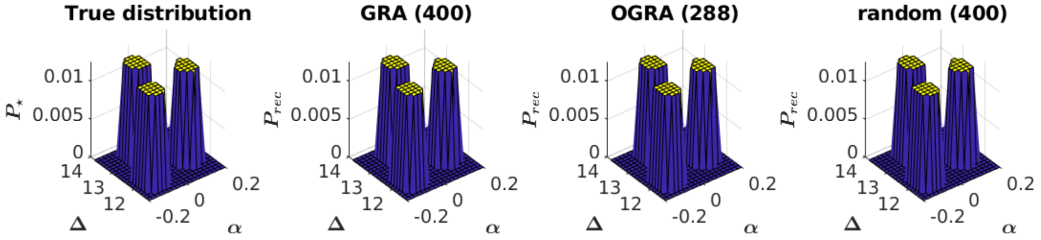

Fig. 6: Same as Fig. 5 but for a step distribution with three peaks. In brackets are the number of control fields for each set.

section (i.e. $\Delta_1 = 0.2$ and $t_f = 16$), but for $K = 400$ mesh points. The results for a Gaussian and a step distribution with three peaks are plotted in Figs. 5 and 6, respectively. We observe that all three control field sets are able to fully reconstruct the step distribution and, at least partially, the Gaussian distribution. This is because the admissible set of solutions for the final identification problem (4) is restricted to $\widehat{\mathbb{R}}^K$. Thus, a bad condition number does not necessarily imply that it is impossible to (at least partially) reconstruct the true probability distribution. However, a good condition number guarantees stability of the numerical solver and improves the accuracy of the results. In this context, notice that if we would sufficiently increase either the control bound u_m, or the maximum control time t_f, both GRA and OGRA would show better condition numbers and be able to perfectly reconstruct also the Gaussian distribution in Fig. 5.

We observe also that, if one knows the number of sufficient control functions $\widetilde{K} = K$, then even completely random control fields can be able to perform similarly to GRA and OGRA controls. However, while OGRA finds automatically $\widetilde{K}$ (reduces the number of control fields to a sufficient amount), there is no indicator for a sufficient amount of random controls in general.

Additionally, the corresponding condition numbers are in many cases worse than for GRA and OGRA, as seen in Tabs. 6 and 7, meaning that they are more likely to show numerical instabilities. Thus, the recommended strategy is clearly OGRA, since it is able to reduce the number of control fields by up to 50%, while accurately reconstructing the probability distributions.

Lastly, we remark that making the tolerance tol_2 smaller can generally lead to even fewer controls being computed by OGRA. However, this in turn can lead to less accurate results in the reconstructed solution, meaning the user has to decide for themselves if such a trade-off is desirable.

7. Conclusion

In conclusion, we introduce `SPIRED`, a Greedy reconstruction algorithm to estimate spin distribution in NMR. We show that this approach can be used to jointly find the distribution of two Hamiltonian parameters, namely the offset term and the magnetic field inhomogeneity. We discuss the accuracy and limitations of this method through experimentally relevant numerical simulations. We provide and describe the codes allowing to reproduce the results of this paper. A proof of the algorithm convergence is also given.

This paper opens the way to a series of interesting questions in quantum control. A first step is to apply this algorithm to other areas in which an ensemble of quantum systems is used. Among others, we mention Bose Einstein Condensates in an optical lattice [39, 11] or molecular rotational dynamics in gas phase [10, 40]. The greedy reconstruction algorithm can in principle be applied to these examples, but specific constraints related to the corresponding experimental setups would be to take into account and would require adaptations of the `SPIRED` code. A final stage concerns the experimental implementation of this approach which seems realistic in the near future in view of the current state of the art.

Acknowledgements

Simon Buchwald is funded by the DFG via the collaborative research center SFB1432, Project-ID 425217212. Gabriele Ciaramella is member of the INDAM GNCS. The research of D. Sugny has been supported by the ANR project “QuCoBEC” ANR-22-CE47-0008-02.

Appendix A. Numerical stability of the matrix W

We study in this section the numerical stability of W. Notice that for constant controls the solution to the dynamical equation (1) can be written as

$$\mathbf{X}_{\mathbf{u},(\Delta,\alpha)}(t) = e^{t(\Delta A+\alpha(\mathbf{u}_x B_x+\mathbf{u}_y B_y))}\mathbf{X}_0.$$

Recall that for two matrices X and Y, we have

$$\|e^Y - e^X\| \leq \|Y - X\| e^{\|Y\|} e^{\|X\|}.$$

Now, consider two parameter pairs $(\alpha_\ell, \Delta_\ell)$ and $(\alpha_{\ell+1}, \Delta_{\ell+1})$, and define $D_\ell := t(\Delta_\ell A + \alpha_\ell(\mathbf{u}_x B_x + \mathbf{u}_y B_y))$ and $D_{\ell+1} := t(\Delta_{\ell+1} A + \alpha_{\ell+1}(\mathbf{u}_x B_x + \mathbf{u}_y B_y))$. Since $\|\mathbf{X}_0\| = 1$, $|\mathbf{u}_x| \leq u_m$ and $|\mathbf{u}_y| \leq u_m$, we obtain

$$\begin{aligned}\|\mathbf{X}_{\mathbf{u},(\alpha_\ell,\Delta_\ell)}(t) - \mathbf{X}_{\mathbf{u},(\alpha_{\ell+1},\Delta_{\ell+1})}(t)\| &\leq e^{\|D_\ell\|} e^{\|D_{\ell+1}\|} \|t(\Delta_\ell A + \alpha_\ell(\mathbf{u}_x B_x + \mathbf{u}_y B_y)) \\ &\qquad - t(\Delta_{\ell+1} A + \alpha_{\ell+1}(\mathbf{u}_x B_x + \mathbf{u}_y B_y))\| \\ &\leq e^{\|D_\ell\|} e^{\|D_{\ell+1}\|} t_f \Big(|(\Delta_\ell - \Delta_{\ell+1})| \|A\| \\ &\qquad + |\alpha_\ell - \alpha_{\ell+1}| u_m (\|B_x\| + \|B_y\|) \Big).\end{aligned}$$

Since the exponential mapping is continuous, we have $e^{\|D_{\ell+1}\|} \to e^{\|D_\ell\|}$ for $\Delta_{\ell+1} \to \Delta_\ell$ and $\alpha_{\ell+1} \to \alpha_\ell$. Thus, the norm of the difference between the two solutions $\mathbf{X}_{\mathbf{u},(\alpha_\ell,\Delta_\ell)}(t_f)$ and $\mathbf{X}_{\mathbf{u},(\alpha_{\ell+1},\Delta_{\ell+1})}(t)$ is bounded by the differences $|\Delta_\ell - \Delta_{\ell+1}|$, $|\alpha_\ell - \alpha_{\ell+1}|$, the bound to the control u_m and the maximum control time t_f. Recalling (11) and that $\phi_j = \boldsymbol{e}_j$ in our example, the matrix entries of W are given by

$$W_{\ell,j} = \sum_k \langle \mathbf{Y}_{\mathbf{u}_k,(\alpha_\ell,\Delta_\ell)}(t_k) | \mathbf{Y}_{\mathbf{u}_k,(\alpha_j,\Delta_j)}(t_k) \rangle.$$

References

[1] D. D'Alessandro, Introduction to Quantum Control and Dynamics, Chapman & Hall/CRC, Boca Raton, 2007.

[2] U. Boscain, M. Sigalotti, D. Sugny, Introduction to the pontryagin maximum principle for quantum optimal control, PRX Quantum 2 (2021) 030203. doi:10.1103/PRXQuantum.2.030203.

[3] C. Altafini, F. Ticozzi, Modeling and control of quantum systems: An introduction, IEEE Trans. Automat. Control 57 (2012) 1898.

[4] D. Dong, I. A. Petersen, Quantum control theory and applications: A survey, IET Control Theory A 4 (2010) 2651.

[5] A. Borzì, G. Ciaramella, M. Sprengel, Formulation and Numerical Solution of Quantum Control Problems, SIAM, Philadelphia, PA, 2017.

[6] M. Levitt, Spin Dynamics: Basics of Nuclear Magnetic Resonance, Wiley, 2013.

[7] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, D. Sugny, F. K. Wilhelm, Training schrödinger's cat: quantum optimal control, Eur. Phys. J. D 69 (12) (2015) 279. doi:10.1140/epjd/e2015-60464-1.

[8] M. Lapert, Y. Zhang, M. A. Janich, S. J. Glaser, D. Sugny, Exploring the physical limits of saturation contrast in magnetic resonance imaging, Sci. Rep. 2 (2012) 589. doi:10.1038/srep00589.

[9] C. Brif, R. Chakrabarti, R. Rabitz, Control of quantum phenomena: past, present and future, New J. Phys. 12 (2010) 075008.

[10] C. P. Koch, M. Lemeshko, D. Sugny, Quantum control of molecular rotation, Rev. Mod. Phys. 91 (2019) 035005. doi:10.1103/RevModPhys.91.035005.

[11] N. Dupont, G. Chatelain, L. Gabardos, M. Arnal, J. Billy, B. Peaudecerf, D. Sugny, D. Guéry-Odelin, Quantum state control of a bose-einstein condensate in an optical lattice, PRX Quantum 2 (2021) 040303. doi:10.1103/PRXQuantum.2.040303.
URL `https://link.aps.org/doi/10.1103/PRXQuantum.2.040303`

[12] M. Lapert, G. Ferrini, D. Sugny, Optimal control of quantum superpositions in a bosonic josephson junction, Phys. Rev. A 85 (2012) 023611. doi:10.1103/PhysRevA.85.023611.

[13] A. Acín, I. Bloch, H. Buhrman, T. Calarco, C. Eichler, J. Eisert, D. Esteve, N. Gisin, S. J. Glaser, F. Jelezko, S. Kuhr, M. Lewenstein, M. F.

Riedel, P. O. Schmidt, R. Thew, A. Wallraff, I. Walmsley, F. K. Wilhelm, The quantum technologies roadmap: a european community view, New J. Phys. 20 (8) (2018) 080201. doi:10.1088/1367-2630/aad1ea.

[14] C. P. Koch, U. Boscain, T. Calarco, G. Dirr, S. Filipp, S. J. Glaser, R. Kosloff, S. Montangero, T. Schulte-Herbrüggen, D. Sugny, F. K. Wilhelm, Quantum optimal control in quantum technologies. strategic report on current status, visions and goals for research in europe, EPJ Quantum Technology 9 (2022) 19. doi:10.1140/epjqt/s40507-022-00138-x.

[15] C. W. Helstrom, Quantum detection and estimation theory, Journal of Statistical Physics 1 (1969) 231–252. doi:10.1007/BF01007479.

[16] V. Giovannetti, S. Lloyd, L. Maccone, Quantum-enhanced measurements: Beating the standard quantum limit, Science 306 (5700) (2004) 1330–1336. doi:10.1126/science.1104149.

[17] C. L. Degen, F. Reinhard, P. Cappellaro, Quantum sensing, Rev. Mod. Phys. 89 (2017) 035002. doi:10.1103/RevModPhys.89.035002.

[18] Y. Maday, J. Salomon, A greedy algorithm for the identification of quantum systems, in: Proceedings of the 48th IEEE Conference on Decision and Control, 2009, Held jointly whit the 28th Chinese Control Conference (CDC/CCC 2009), IEEE Conference on Decision and Control, 2009, pp. 375–379.

[19] S. Conolly, D. Nishimura, A. Macovski, Optimal control solutions to the magnetic resonance selective excitation problem, IEEE Transactions on Medical Imaging 5 (2) (1986) 106–115. doi:10.1109/TMI.1986.4307754.

[20] T.-M. Zhang, R.-B. Wu, F.-H. Zhang, T.-J. Tarn, G.-L. Long, Minimum-time selective control of homonuclear spins, IEEE Transactions on Control Systems Technology 23 (5) (2015) 2018–2025. doi:10.1109/TCST.2015.2390191.

[21] Q. Ansel, S. J. Glaser, D. Sugny, Selective and robust time-optimal rotations of spin systems, Journal of Physics A: Mathematical and Theoretical 54 (8) (2021) 085204. doi:10.1088/1751-8121/abdba1.
URL `https://dx.doi.org/10.1088/1751-8121/abdba1`

[22] L. Van Damme, Q. Ansel, S. J. Glaser, D. Sugny, Time-optimal selective pulses of two uncoupled spin-1/2 particles, Phys. Rev. A 98 (2018) 043421. doi:10.1103/PhysRevA.98.043421.

[23] J. Liu, H. Yuan, Quantum parameter estimation with optimal control, Phys. Rev. A 96 (2017) 012117. doi:10.1103/PhysRevA.96.012117.

[24] H. Yuan, C. F. Fung, Quantum parameter estimation with general dynamics, npj Quantum Inf. 3 (2017) 14.

[25] C. Lin, Y. Ma, D. Sels, Optimal control for quantum metrology via pontryagin's principle, Phys. Rev. A 103 (2021) 052607. doi:10.1103/PhysRevA.103.052607.

[26] J. Liu, M. Zhang, H. Chen, L. Wang, H. Yuan, Optimal scheme for quantum metrology, Advanced Quantum Technologies 5 (1) (2022) 2100080. doi:https://doi.org/10.1002/qute.202100080.

[27] C. Lin, Y. Ma, D. Sels, Application of pontryagin's maximum principle to quantum metrology in dissipative systems, Phys. Rev. A 105 (2022) 042621. doi:10.1103/PhysRevA.105.042621.

[28] D. Ma, V. Gulani, N. Seiberlich, Magnetic resonance fingerprinting, Nature 495 (2013) 187.

[29] Q. Ansel, M. Tesch, S. J. Glaser, D. Sugny, Optimizing fingerprinting experiments for parameter identification: Application to spin systems, Phys. Rev. A 96 (2017) 053419. doi:10.1103/PhysRevA.96.053419.

[30] S. Buchwald, G. Ciaramella, J. Salomon, D. Sugny, Greedy reconstruction algorithm for the identification of spin distribution, Phys. Rev. A 104 (2021) 063112.

[31] S. Buchwald, G. Ciaramella, J. Salomon, Analysis of a greedy reconstruction algorithm, SIAM Journal on Control and Optimization 59 (6) (2021) 4511–4537. doi:10.1137/20M1373384.

[32] K. Kobzar, T. E. Skinner, N. Khaneja, S. J. Glaser, B. Luy, Exploring the limits of broadband excitation and inversion: Ii. rf-power optimized pulses, J. Magn. Reson. 194 (1) (2008) 58–66. doi:https://doi.org/10.1016/j.jmr.2008.05.023.

[33] M. Lapert, Y. Zhang, M. Braun, S. J. Glaser, D. Sugny, Singular extremals for the time-optimal control of dissipative spin $\frac{1}{2}$ particles, Phys. Rev. Lett. 104 (2010) 083001. doi:10.1103/PhysRevLett.104.083001.

[34] N. Khaneja, R. Brockett, S. J. Glaser, Phys. Rev. A 63 (2001) 032308.

[35] B. Bonnard, O. Cots, S. J. Glaser, M. Lapert, D. Sugny, Y. Zhang, Geometric optimal control of the contrast imaging problem in nuclear magnetic resonance, IEEE Trans. Autom. Control 57 (8) (2012) 1957–1969. doi:10.1109/TAC.2012.2195859.

[36] T. E. Skinner, T. O. Reiss, B. Luy, N. Khaneja, S. J. Glaser, Tailoring the optimal control cost function to a desired output: application to minimizing phase errors in short broadband excitation pulses, J. Magn. Reson. 172 (1) (2005) 17–23.

[37] Rodrigues, Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire., Journal de Mathématiques Pures et Appliquées (1840) 380–440.

[38] K. Lundengård, Generalized vandermonde matrices and determinants in electromagnetic compatibility, 2017.

[39] C. A. Weidner, D. Z. Anderson, Experimental demonstration of shaken-lattice interferometry, Phys. Rev. Lett. 120 (2018) 263201. doi:10.1103/PhysRevLett.120.263201.

[40] D. Sugny, A. Keller, O. Atabek, D. Daems, C. M. Dion, S. Guérin, H. R. Jauslin, Laser control for the optimal evolution of pure quantum states, Phys. Rev. A 71 (2005) 063402. doi:10.1103/PhysRevA.71.063402.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

80/2023 Buchwald, S.; Ciaramella, G; Salomon, J.; Sugny, D.
A SPIRED code for the reconstruction of spin distribution

79/2023 Agosti, A.; Bardin, R.; Ciarletta, P.; Grasselli, M.
A diffuse interface model of tumour evolution under a finite elastic confinement

78/2023 Antonietti, P.F.; Bonizzoni, F.; Corti, M.; Dall'Olio, A.
Discontinuous Galerkin for the heterodimer model of prion dynamics in Parkinson's disease

77/2023 Fumagalli, I.; Corti, M.; Parolini, N.; Antonietti, P. F.
Polytopal discontinuous Galerkin discretization of brain multiphysics flow dynamics

76/2023 Ieva, F.; Galliani, G.; Secchi, P.
The impact of public transport on the diffusion of COVID-19 pandemie in Lombardy during 2020

74/2023 Pidò, S.; Pinoli, P.; Crovari, P.; Ieva, F.; Garzotto, F.; Ceri, S.
Ask Your Data—Supporting Data Science Processes by Combining AutoML and Conversational Interfaces

75/2023 Archetti, A.; Ieva, F.; Matteucci, M.
Scaling survival analysis in healthcare with federated survival forests: A comparative study on heart failure and breast cancer genomics

71/2023 Conni, G.; Piccardo, S.; Perotto, S.; Porta, G.M.; Icardi, M.
HiPhome: HIgh order Projection-based HOMogEnisation for advection diffusion reaction problems

70/2023 Ragni, A.; Ippolito, D.; Masci, C.
Assessing the Impact of Hybrid Teaching on Students' Academic Performance via Multilevel Propensity Score-based techniques

69/2023 Ferro, N.; Micheletti, S.; Parolini, N.; Perotto, S.; Verani, M.; Antonietti, P. F.
Level set-fitted polytopal meshes with application to structural topology optimization