This is the accepted manuscript made available via CHORUS. The article has been published as:

Optimal control with accelerated convergence: Combining the Krotov and quasi-Newton methods

Reuven Eitan, Michael Mundt, and David J. Tannor

Phys. Rev. A **83**, 053426 — Published 24 May 2011

DOI: 10.1103/PhysRevA.83.053426

Optimal control with accelerated convergence: Combining the Krotov and quasi-Newton methods

Reuven Eitan, Michael Mundt and David J. Tannor*

Department of Chemical Physics,

The Weizmann Institute of Science,

Rehovot 76100, Israel

REVIEW COPY
NOT FOR DISTRIBUTION

Abstract

One of the most popular methods for solving numerical Optimal Control problems is the Krotov method, adapted for quantum control by Tannor and coworkers. The Krotov method has the following three appealing properties: 1) monotonic increase of the objective with iteration number; 2) no requirement for a line search, leading to a significant savings over gradient (first-order) methods; 3) macro-steps at each iteration, resulting in significantly faster growth of the objective at early iterations than in gradient methods where small steps are required. The principal drawback of the Krotov method is slow convergence at later iterations, which is particularly problematic when high fidelity is desired. We show here that near convergence the Krotov method degenerates to a first-order gradient method. We then present a variation on the Krotov method that has all the advantages of the original Krotov method but with significantly enhanced convergence (second-order or quasi-Newton) as the optimal solution is approached. We illustrate the method by controlling the three-dimensional dynamics of the valence electron in the Na atom.

*Electronic address: reuven.eitan@weizmann.ac.il,michael.g.mundt@web.de,David.Tannor@weizmann.ac.il

I. INTRODUCTION

The field of quantum control has developed rapidly since the 1980's. The key aspects of the theory were formulated by Tannor, Rice, Kosloff, Shapiro, Brumer and Rabitz [1–5], and the first experimental demonstrations were performed a decade later [6–8]. Today, with the advances in technology one can shape the amplitude, phase and polarization of femtosecond laser pulses. Recently, the production of attosecond laser pulses has also been demonstrated [9], which opens the door to the possibility of controlling fast electronic processes.

One of the main theoretical tools in quantum control is Optimal Control Theory (OCT). OCT provides a variational framework for calculating the optimal shaped laser pulse to maximize a desired physical objective. Its application to quantum mechanics was first formulated in the late 1980's [5, 10]. In rare instances, the equations for the optimal control field that emerge from OCT can be solved analytically. However in the vast majority of cases a numerical treatment is required. Invariably, these numerical treatments are iterative — one starts from an initial guess for the control and improves on it by repeating some procedure over and over.

The standard optimization approaches are either gradient methods [11], which use first derivative information, or Newton and quasi-Newton methods, which use second derivative information. A somewhat unconventional approach is the Krotov method [12], adapted for quantum control by Tannor *et al.* [13, 14]. A decade later Sklarz and Tannor [15] adapted the nonlinear Krotov method [16] for quantum control, where for example the system equation of motion is nonlinear. The Krotov algorithm has been used with great success in many works for applications ranging from cooling [17] to quantum communication [18] and quantum computation [19, 20]. A related algorithm was introduced by Zhu and Rabitz [21], and an illuminating comparison of the algorithms was carried out by Maday and Turinici [22]. The Krotov method has several distinct adventages over gradient and quasi-Newton methods: 1) monotonic increase of the objective with iteration number; 2) no requirement for a line search, leading to a significant savings over gradient (first-order) methods; 3) macro-steps at each iteration, resulting in significantly faster growth of the objective at early iterations than in gradient methods where small steps are required. On the other hand, at high iteration number the method tends to become inefficient and it is hard to achieve high fidelities with this method. With the Newton and quasi-Newton

methods one can achieve high fidelities; however because of its line-search procedure it is nonmonotonic and is restricted to micro-steps, i.e. a significant number of iterations are usually needed to get a reasonable outcome. (The simple gradient search suffers from both shortcomings: low fidelity together with nonmonotonicity and micro-steps. However it is a relatively simple procedure that can be useful in certain cases.)

In this article we analyze the source of the inefficiency of the Krotov method at later iterations, and show that it is due to the degeneration of the method to a first-order gradient method as convergence is approached. We then present a variation on the Krotov algorithm that combines the benefits of the original Krotov method with that of Newton's method — monotonicity and macro-steps, along with significantly improved convergence at high iteration numbers.

The remainder of this article is organized as follows. Section II is devoted to methodology. We first formulate OCT using continuous functions (Section II A); from now on we will call this the "function language". We use this formalism to show the degeneration of the Krotov algorithm to a gradient method as convergence is approached (Section II B). We then proceed to describe the Newton and quasi-Newton methods, focusing on a particular algorithm known as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm after its inventorsapplied [23–26]. To explain this algorithm we first reformulate OCT in a vector space (the "vector language", Section II C). We then continue by explaining Newton's method in general and its approximation, the BFGS method in particular (Section II D). Finally we present our algorithm, which we call Krotov–BFGS (or K–BFGS for short, Section II E). For each method, after presenting the working equations of the method we provide a step-by-step description of the algorithm. Section III is devoted to numerical examples. Section 4 is a Summary and Conclusion.

II. METHODOLOGY

A. General Formalism for OCT – Necessary Condition for Optimal Control

Consider a system defined by the complex state function, $\psi(\mathbf{r}, t)$. The controllable parameters of the system are represented by the function $\xi(t)$. The state function, ψ, evolves

under an equation of motion of the general form:

$$\dot{\psi}(\mathbf{r},t) = f[\psi,\xi](\mathbf{r},t). \tag{1}$$

In Dirac notation Eq. (1) is written as: $|\dot{\psi}(t)\rangle = |f[\psi,\xi](t)\rangle$. We will frequently use Dirac notation in the rest of the article and we will generally omit writing the dependence on t except in integrals over t.

Given the equation of motion one would like to maximize the value of some target at time T [27]. The target could be a specific state ("state-to-state transfer"), an observable or something else. One can generally write the objective in the form:

$$J = \langle\psi(T)|P|\psi(T)\rangle, \tag{2}$$

where P is some projection operator. For generality we will simply write $J[\psi(T)]$.

Once the mathematical form of the objective functional is defined, maximizing the target corresponds to maximizing the objective under the constraint of Eq. (1). This is done by adding to the objective the equation of motion using a Lagrange multiplier function χ (sometimes referred to as the "dual" or "adjoint" function), and defining a new functional $\bar{J}$:

$$\bar{J} = J - 2\mathrm{Re}\int_0^T \langle\chi(t)|\dot{\psi}(t) - f[\psi,\xi](t)\rangle dt \tag{3}$$

It is sometimes useful to add other costs on the controls [28]:

$$J_{\mathrm{c}} = \int_0^T \lambda(t) g[\xi](t) dt. \tag{4}$$

There are two purposes J_{c} can serve; these two purposes correspond to two different meanings for λ. (a) If the purpose of J_{c} is to be an additional constraint, i.e., ξ fulfills some equation, then λ should be treated as a Lagrange multiplier. In this case the maxima of $\bar{J}$ still equal the maxima of J since adding J_{c} is to add zero. (b) However, if the purpose of J_{c} is to be a penalty function, which means the system has some freedom to deviate from an equation for ξ but is penalized for doing so, then λ is a fixed predetermined function that determines the penalty for deviation. For this option, J_{c} has to be non-positive for maximization (and non-negative for minimization). However, in this case the maxima of $\bar{J}$ will differ in general

from the maxima of J. Nevertheless one can still keep J_c small enough relative to J so the maxima of $\bar{J}$ will be approximately the maxima of J [29]. For the remainder of this article we assume that:

$$g[\xi](t) = \big(\xi(t) - \zeta(t)\big)^2, \tag{5}$$

and therefore:

$$J_c = -\int_0^T \lambda(t)\big(\xi(t) - \zeta(t)\big)^2 dt, \tag{6}$$

where ζ is the "reference control", and $\lambda(t) > 0$.

Combining everything together we obtain:

$$\bar{J} \;=\; J + J_1 + J_2 + J_3, \tag{7a}$$

where

$$J_1 \;=\; -2\mathrm{Re}\int_0^T \langle\chi(t)|\dot{\psi}(t)\rangle dt, \tag{7b}$$

$$J_2 \;=\; 2\mathrm{Re}\int_0^T \langle\chi(t)|f[\psi,\xi](t)\rangle dt, \tag{7c}$$

$$J_3 \;=\; \int_0^T \lambda(t) g[\xi](t) dt = -\int_0^T \lambda(t)\big(\xi(t) - \zeta(t)\big)^2 dt. \tag{7d}$$

The condition for an optimal control is that $\bar{J}$ be maximized, i.e. that $\nabla\bar{J} = 0$, or more specifically $\frac{\delta\bar{J}}{\delta\xi} = \frac{\delta\bar{J}}{\delta\xi^*} = \frac{\delta\bar{J}}{\delta|\psi\rangle} = \frac{\delta\bar{J}}{\delta\langle\psi|} = 0$. Using the maximization condition $\frac{\delta\bar{J}}{\delta\langle\psi|} = 0$ [30], one gets an equation of motion for the Lagrange multiplier, χ, and its value at the final time, T (see Appendix A):

$$|\dot{\chi}\rangle \;=\; -\frac{\delta J_2}{\delta\langle\psi|} \tag{8a}$$

$$|\chi(T)\rangle \;=\; \frac{\delta J}{\delta\langle\psi(T)|}. \tag{8b}$$

By taking χ to fulfill Eqs. (8a, 8b) one can address the $\frac{\delta\bar{J}}{\delta\xi}$ part of the gradient of $\bar{J}$ alone (since all the other parts are zero), and one obtains:

$$\frac{\delta\bar{J}}{\delta\xi} \;=\; 2\mathrm{Re}\left\langle\chi\middle|\frac{\delta f[\psi,\xi]}{\delta\xi}\right\rangle + \lambda\frac{\delta g[\xi]}{\delta\xi} = 2\mathrm{Re}\left\langle\chi\middle|\frac{\delta f[\psi,\xi]}{\delta\xi}\right\rangle - 2\lambda\left(\xi - \zeta\right), \tag{9}$$

where we emphasize that both the LHS and the RHS are functions of time. $\bar{J}$ is then maximized by a $\xi = \xi_0$ that gives $\frac{\delta \bar{J}}{\delta \xi} = 0$ [31].

There are a few methods for finding (or at least approaching) the ξ_0 that fulfills $\frac{\delta \bar{J}}{\delta \xi} = 0$. Except for a few specific cases where a full analytical solution is possible, all methods use Eqs. (7a – 9) in an iterative self-consistent manner. In this article we present three such methods. On one hand there are methods that do not use a line-search approach. The premier example is the Krotov method, which as we will see immediately below has the advantage of monotonic convergence of $\bar{J}$ as a function of the iterations. On the other hand there are line-search methods — Newton's method and in particular the BFGS (and its limited memory version, LBFGS). These methods use second order information to approach ξ_0 but due to their line-search lack monotonic convergence with iteration number. Finally, we present our combined version of the two, the Krotov–BFGS (K–BFGS) method, which preserves both advantages, second order information with monotonicity in the iterations.

B. The Krotov Method

The main feature of the Krotov method (in fact Krotov originally derived his method to fulfill this feature) is the monotonicity of $\bar{J}$ as a function of the iterations (i.e. $\bar{J}^{(k)} \geq \bar{J}^{(k-1)}$, for the kth iteration). We consider systems that evolve under linear equations of motion, (i.e. f in Eq. (1) depends linearly on ψ and ξ). The Krotov method achieves monotonicity by updating the control in the kth iteration as (see Appendix B):

$$\xi^{(k)} \;=\; \zeta^{(k)} + \frac{1}{\lambda}\mathrm{Re}\left\langle \chi^{(k-1)} \middle| \frac{\delta f[\psi^{(k)}, \xi]}{\delta \xi} \right\rangle \equiv y[\psi^{(k)}, \chi^{(k-1)}]. \tag{10}$$

Notice that in the Krotov method the ψ of the kth iteration, $\psi^{(k)}$, is used to update the kth guess of the control, $\xi^{(k)}$, while also depending on it by Eq. (1). This differs from gradient search approaches that use only information from the previous iteration(s). The function, ζ, is some reference control that in principle can be chosen differently from iteration to iteration. Choosing $\zeta^{(k)}$ as fixed for all iterations (i.e, $\zeta^{(k)} = \zeta^{(k-1)} = \zeta$), requires that J_3 be chosen as in Eq. (6) if monotonicity is to be preserved. However, if one allows ζ to change from one iteration to the next, and takes $\zeta^{(k)} = \xi^{(k-1)}$, then monotonicity is achieved even if $J_3 = 0$ (a proof is given in Appendix C). Nevertheless, even in this case, the function λ in Eq. (10) is still needed as a calibration for the steps of the iterations (see in [15] Ref. [15]) .

Experience shows that the Krotov method is inefficient as convergence is approached, i.e., one cannot reach high fidelities with the Krotov method in a reasonable number of iterations, while with second order methods like Newton's method one can reach high fidelities. In this section we give a theoretical analysis of the Krotov method that sheds light on this phenomenon.

Taylor expanding the $\psi^{(k)}$ equation of motion (RHS of Eq. (1)) around $\psi^{(k-1)}$, we obtain:

$$|f[\psi^{(k)},\xi]\rangle \;=\; |f[\psi^{(k-1)},\xi]\rangle + \left|\frac{\delta f[\psi,\xi]}{\delta\psi}\Delta\psi\right\rangle, \tag{11}$$

where $\Delta\psi = \psi^{(k)} - \psi^{(k-1)}$. (Since we assume linear dependence of f on ψ, higher order elements vanish.) Varying Eq. (11) with respect to the control gives:

$$\left|\frac{\delta f[\psi^{(k)},\xi]}{\delta\xi}\right\rangle = \left|\frac{\delta f[\psi^{(k-1)},\xi]}{\delta\xi}\right\rangle + \left|\frac{\delta^2 f[\psi,\xi]}{\delta\xi\delta\psi}\Delta\psi\right\rangle. \tag{12}$$

Inserting Eq. (12) into Eq. (10) gives:

$$\xi^{(k)} = \zeta^{(k)} + \frac{1}{\lambda}\left(\xi_1^{(k)} + \xi_2^{(k)}\right), \tag{13a}$$

where:

$$\xi_1^{(k)} \;=\; \mathrm{Re}\left\langle \chi^{(k-1)}\middle|\frac{\delta f[\psi^{(k-1)},\xi]}{\delta\xi}\right\rangle \tag{13b}$$

$$\xi_2^{(k)} \;=\; \mathrm{Re}\left\langle \chi^{(k-1)}\middle|\frac{\delta^2 f[\psi,\xi]}{\delta\xi\delta\psi}\Delta\psi\right\rangle. \tag{13c}$$

According to Eqs. (9) and (5) $\frac{\delta\bar{J}^{(k-1)}}{\delta\xi}$ is given by:

$$\begin{aligned}\frac{\delta\bar{J}^{(k-1)}}{\delta\xi} \;&=\; 2\mathrm{Re}\left\langle \chi^{(k-1)}\middle|\frac{\delta f[\psi^{(k-1)},\xi]}{\delta\xi}\right\rangle - 2\lambda\big(\xi^{(k-1)} - \zeta^{(k-1)}\big)\\ &=\; 2\left\{\xi_1^{(k)} - \lambda\left(\xi^{(k-1)} - \zeta^{(k-1)}\right)\right\}.\end{aligned} \tag{14}$$

Combining Eq. (14) and Eq. (13a) gives:

$$\begin{aligned}\xi^{(k)} \;&=\; \zeta^{(k)} + \frac{1}{2\lambda}\frac{\delta\bar{J}^{(k-1)}}{\delta\xi} + \xi^{(k-1)} - \zeta^{(k-1)} + \frac{1}{\lambda}\xi_2^{(k)}\\ &=\; \tilde{\zeta}^{(k)} + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi} + 2\alpha\xi_2^{(k)},\end{aligned} \tag{15}$$

where $\tilde{\zeta}^{(k)} = \zeta^{(k)} + \xi^{(k-1)} - \zeta^{(k-1)}$, and $\alpha = \frac{1}{2\lambda}$. We find that the update rule is composed of two elements: the 'simple' first order gradient element (the first two terms on the RHS

of Eq. (15)) and a correction term that depends on the change in ψ between iterations (the third term on the RHS of Eq. (15)). We show in Appendix D that near convergence this third term goes to zero and therefore the Krotov method degenerates to a first-order gradient method. The main disadvantage of first-order gradient methods is that in many cases the first order information is not enough to reach high fidelities in a reasonable number of iterations, since the neighborhood of a maximum is almost flat. (A consequence of this is that one gets a strong dependence on the value of λ, which can cause numerical instabilities.)

We conclude this section by giving the algorithmic steps of the Krotov method. At iteration k:

(i) Using $\chi^{(k-1)}$ from the previous iteration and $\psi^{(k)}$ of the present iteration, an improved control $\xi^{(k)}$ is calculated by Eq. (10). This improved control is used to propagate $\psi^{(k)}$ according to Eq. (1). (Since $\xi^{(k)}$ depends on $\psi^{(k)}$ we solve the equations self-consistently using what is known as an implicit scheme. One can also use an explicit scheme, however it is known to be less stable.)

(ii) At time T use Eqs. (7a – 7d) to calculate $\bar{J}^{(k)}$.

(iii) By Eq. (8b), one gets the final condition of the Lagrange multiplier, $\chi^{(k)}(T)$.

(iv) Using Eq. (8a), $\chi^{(k)}$ is propagated and calculated for all times.

The procedure described above is repeated until convergence. At iteration 0 in step (i) one guesses a control $\xi^{(0)}$ and uses it to propagate $\psi^{(0)}$ up to time T with Eq. (1). A schematic diagram of the algorithm is given in Fig. 1.

C. Translation of the Optimal Control Formalism to Vector Language

The formalism of Newton's method (and its approximation, BFGS) is most easily presented using discrete vectors rather then continuous functions. Therefore, we start by reformulating OCT in this "vector language". For generality we write the formalism for a general vector space. However, at the end we "translate" the results to Dirac notation.

Following section 2.1, we consider a complex state vector, $\vec{v}$:

$$\vec{v} = (v_1, v_2, \cdots, v_n)\,, \tag{16}$$

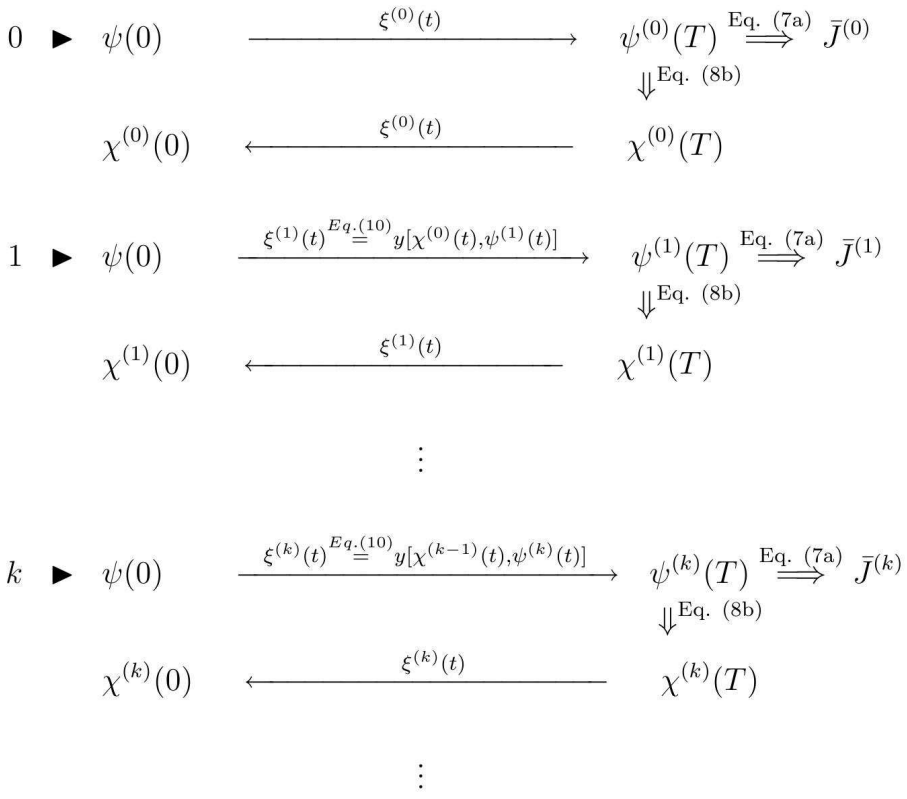

Figure 1: Schematic diagram of the Krotov algorithm.

where each of the components here are functions of the Hilbert space. The jth vector component obeys the following update rule:

$$v_j = f_j[\vec{v}, \vec{\xi}], \tag{17}$$

where $\vec{\xi}$ is the control vector:

$$\vec{\xi} = (\xi_1, \xi_2, \cdots, \xi_n)\,. \tag{18}$$

Although this formalism is general, for our purposes the components of the vectors play the role of a discrete version of the continuous time dependence of the functions in section 2.1 (one could look at them as a numeric time grid).

For any two vectors in the space, $\vec{w}$ and $\vec{\widetilde{w}}$, we define a notation for the inner product of

their kth and lth components, w_k and $\widetilde{w}_l$, respectively:

$$w_k \widetilde{w}_l = \int_{-\infty}^{-\infty} w_k(\mathbf{r}) \widetilde{w}_l(\mathbf{r}) d\mathbf{r}. \tag{19}$$

We now can reformulate Eqs. (7) with the help of a Lagrange multiplier vector, $\vec{u} = (u_1, u_2, \cdots, u_n)$:

$$\bar{J} \;=\; J + J_1 + J_2 + J_3, \tag{20}$$

where

$$J = J[v_n], \tag{21a}$$

i.e. the objective functional J depends only on v_n, and

$$J_1 \;=\; -2\mathrm{Re}\left(u_n^* v_n + \sum_{j=1}^{n-1} u_j^* v_j \right), \tag{21b}$$

$$J_2 \;=\; 2\mathrm{Re} \sum_{j=1}^{n-1} u_j^* f_j[\vec{v}, \vec{\xi}], \tag{21c}$$

$$J_3 \;=\; \sum_{j=1}^{n-1} \lambda_j g_j[\vec{\xi}] = -\sum_{j=1}^{n-1} \lambda_j \left(\xi_j - \zeta_j\right)^2, \tag{21d}$$

where $\vec{\zeta}$ is a reference control vector, and the discussion of J_3 in section 2.1 is also relevant here. Using the maximization condition $\frac{\delta \bar{J}}{\delta v_j^*} = 0$ one gets the analog of Eqs. (8):

$$u_j \;=\; -\frac{\delta J_2}{\delta v_j^*}; \quad j \neq n \tag{22a}$$

$$u_n \;=\; \frac{\delta J}{\delta v_n^*}. \tag{22b}$$

Equations (22a, 22b) lead to an equation for the gradient of $\bar{J}$ (the analog of Eq. (9)):

$$\begin{aligned} \frac{\delta \bar{J}}{\delta \xi_s} \;&=\; 2\mathrm{Re} \sum_{j=1}^{n-1} u_j^* \frac{\delta f_j[\vec{v}, \vec{\xi}]}{\delta \xi_s} + \sum_{j=1}^{n-1} \lambda_j \frac{\delta g_j[\vec{\xi}]}{\delta \xi_s} \\ &=\; 2\mathrm{Re} \sum_{j=1}^{n-1} u_j^* \frac{\delta f_j[\vec{v}, \vec{\xi}]}{\delta \xi_s} - 2\lambda_s \left(\xi_s - \zeta_s\right). \end{aligned} \tag{23}$$

With this "vector formalism of OCT" in hand we can now explain Newton's method and its BFGS approximation.

D. Newton's method and the BFGS Method

Newton's method uses a line-search approach to minimize a function $\mathcal{F}$ (i.e., a search along the gradient of the function to find a minimum). The method minimizes a function, $\mathcal{F}:\mathbb{C}^n\rightarrow\mathbb{R}$ (for maximization one can minimize $-\mathcal{F}$), if its gradient, $\vec{\nabla}\mathcal{F}$, and its Hessian matrix $\mathbf{A}$ (or its inverse $\mathbf{A}^{-1}$) are known (where $A_{ij}=\frac{\delta^2\mathcal{F}}{\delta x_i\delta x_j}$, $\vec{x}=(x_1,x_1,\cdots,x_n)\in\mathbb{C}^n$). The actual implementation of the method is iterative, moving along a path until minimization has been achieved. The kth iteration step for $\vec{x}$ ($\vec{x}^{(k)}$) is given by:

$$\vec{x}^{(k+1)}=\vec{x}^{(k)}-\mathbf{A}^{-1}(\vec{x}^{(k)})\vec{\nabla}\mathcal{F}(\vec{x}^{(k)}). \tag{24}$$

Unfortunately, in many cases the Hessian $\mathbf{A}$ is unknown or unfeasible to calculate on a computer. The BFGS method [23–26] approximates the Hessian matrix of the kth iteration, $\mathbf{A}\approx\mathbf{B}^{(k)}$, using the known quantities $\vec{x}^{(k-1)}$, $\vec{x}^{(k)}$, $\vec{\nabla}\mathcal{F}(\vec{x}^{(k-1)})$, $\vec{\nabla}\mathcal{F}(\vec{x}^{(k)})$, and $\mathbf{B}^{(k-1)}$ (usually $\mathbf{B}^{(0)}$ is taken as the unit matrix). These kinds of methods are referred to as "quasi-Newton" methods. For the kth iteration we can therefore rewrite Eq. (24) as:

$$\mathbf{B}^{(k)}\left(\vec{x}^{(k+1)}-\vec{x}^{(k)}\right)=-\vec{\nabla}\mathcal{F}(\vec{x}^{(k)}), \tag{25}$$

where $\vec{x}^{(k+1)}$ is yet unknown. Solving the equation:

$$\mathbf{B}^{(k)}\vec{p}^{(k)}=-\vec{\nabla}\mathcal{F}(\vec{x}^{(k)}), \tag{26}$$

gives a vector, $\vec{p}^{(k)}$. If $\mathcal{F}$ were truly quadratic then $\vec{x}^{(k+1)}$ would be completely determined by Eq. (26); however because in general the function is not quadratic one performs a line-search

$$\vec{x}^{(k+1)}=\vec{x}^{(k)}+\alpha^{(k)}\vec{p}^{(k)} \tag{27}$$

to minimize $\mathcal{F}$ (sometimes another condition, e.g. on the gradient, is added). $\alpha^{(k)}$ is a scalar that one should change until the minimum is found. After finding the minimum, one updates the Hessian approximation for the next iteration according to the formula [23–26]:

$$\mathbf{B}^{(k+1)}=\mathbf{B}^{(k)}+\frac{\vec{y}^{(k)}(\vec{y}^{(k)})^T}{(\vec{y}^{(k)})^T\vec{s}^{(k)}}-\frac{\mathbf{B}^{(k)}\vec{s}^{(k)}(\mathbf{B}^{(k)}\vec{s}^{(k)})^T}{(\vec{s}^{(k)})^T\mathbf{B}^{(k)}\vec{s}^{(k)}}, \tag{28}$$

where $\vec{y}^{(k)}=\vec{\nabla}\mathcal{F}(\vec{x}^{(k+1)})-\vec{\nabla}\mathcal{F}(\vec{x}^{(k)})$, and $\vec{s}^{(k)}=\alpha^{(k)}\vec{p}^{(k)}$. The process is continued until convergence.

The disadvantage of the method is apparent from Eq. (27) — the need to perform a line-search — causing the method to lack monotonicity with respect to iteration number.

Although the dimension of the vector space is n, since $\mathbf{B}$ is symmetric only $\frac{1}{2}n(n+1)$ data points need to be stored. Nevertheless, is some systems even this is too large. The Limited-memory BFGS method [32, 33] (known as the LBFGS method) uses a smaller amount of space to store the BFGS approximation to the Hessian.

Translation of this notation to Dirac notation is straightforward: $\vec{x}$ corresponds to $\vec{v}$ which corresponds to $|\psi\rangle = (|\psi_1\rangle, |\psi_2\rangle, \cdots, |\psi_n\rangle)$, and similarly $\vec{u}$ corresponds to $|\chi\rangle = (|\chi_1\rangle, |\chi_2\rangle, \cdots, |\chi_n\rangle)$. Therefore, in iteration k:

(i) Propagate $|\psi^{(k)}\rangle$ using the $\vec{\xi}^{(k)}$ calculated (by Eq. (17)) at the previous iteration.

(ii) At time T use Eqs. (20 – 21d) to calculate $\bar{J}^{(k)}$.

(iii) Use Eq. (22b) to get the final condition on the Lagrange multiplier, $|\chi_n^{(k)}\rangle$.

(iv) Use Eq. (22a) to calculate $|\chi_j^{(k)}\rangle$ for all j's, and use Eq. (23) to calculate $\nabla\bar{J}^{(k)}$.

(v) Calculate $\vec{p}^{(k)}$ by Eq. (26).

(vii) Guess (smartly) a value for $\alpha^{(k)}$.

(viii) Update $\vec{\xi}^{(k)}$ by Eq. (27) to $\tilde{\vec{\xi}}^{(k+1)}$.

(iix) Propagate $|\tilde{\psi}^{(k+1)}\rangle$ with $\tilde{\vec{\xi}}^{(k+1)}$ by Eq. (17).

(ix) At time T use Eqs. (20 – 21d) to calculate $\tilde{J}^{(k+1)}$.

(x) Check: *if* $\tilde{J}^{(k+1)}$ is not at the minima, guess (smartly) a new $\alpha^{(k)}$ and go back to step (viii); *else* continue to step (xi).

(xi) Accept $\vec{\xi}^{(k+1)} = \tilde{\vec{\xi}}^{(k+1)}$, and update the Hessian approximation, $\mathbf{B}^{(k+1)}$, by Eq. (28).

As in the Krotov algorithm, at iteration 0 in step (i) one guesses a control, $\vec{\xi}^{(0)}$, and uses it to propagate $|\psi^{(0)}\rangle$ up to time T via Eq. (17). When performing the algorithm in practice, note that every time one reaches step (xi) one can skip steps (i) and (ii) by setting $|\psi^{(k+1)}\rangle = |\tilde{\psi}^{(k+1)}\rangle$ and $\bar{J}^{(k+1)} = \tilde{J}^{(k+1)}$. The procedure described above is repeated until convergence. Figure 2 presents a schematic diagram of the algorithm.

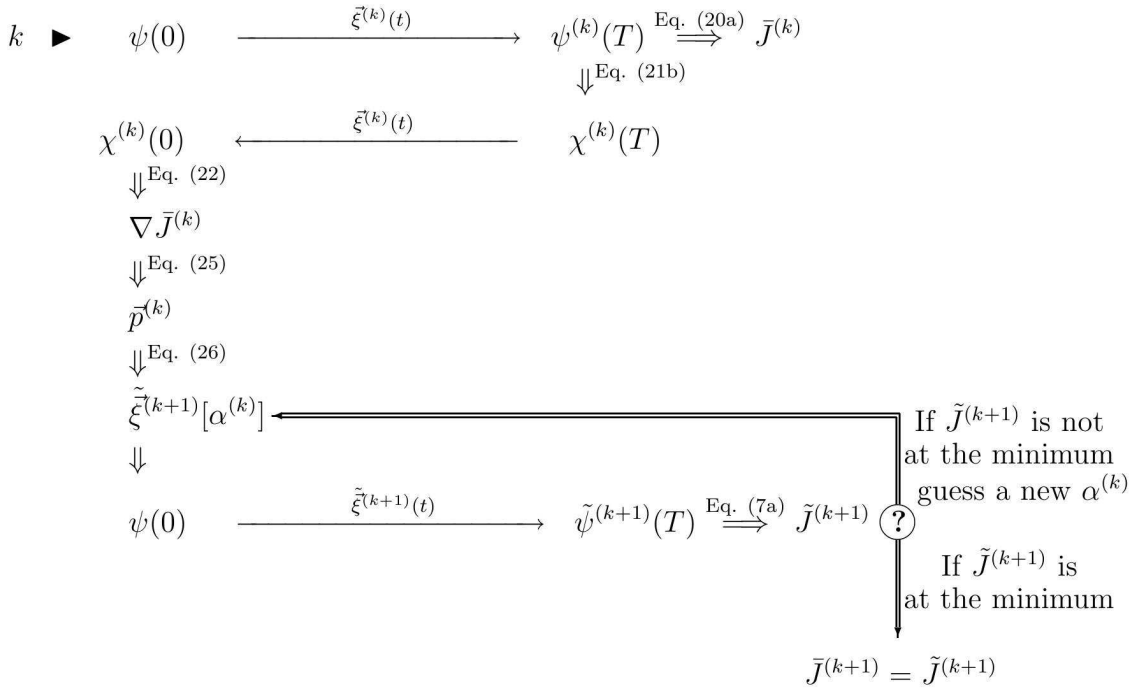

Figure 2: A schematic diagram of the kth iteration in the BFGS algorithm.

E. Combining the Krotov and the BFGS Methods (K–BFGS)

1. The Form of Krotov in Vector Language

Before combining the Krotov and BFGS methods, we need to reformulate the Krotov update rule (Eq. (10)) in vector language:

$$\xi_s^{(k)} = \zeta_s^{(k)} - \frac{1}{\lambda_s}\mathrm{Re}\sum_{j=1}^{n-1} u_j^{*(k-1)}\frac{\delta f_j[\vec{v}^{(k)},\vec{\xi}]}{\delta\xi_s} \tag{29}$$

$$= \zeta_s^{(k)} - \frac{1}{\lambda_s}\mathrm{Re}\left(u_s^{*(k-1)}\frac{\delta f_s[\vec{v}^{(k)},\vec{\xi}]}{\delta\xi_s}\right), \tag{30}$$

where Eq. (30) follows from the fact that $\frac{\delta f_j}{\delta\xi_s} = \frac{\delta f_s}{\delta\xi_s}\delta_{js}$.

2. Combining Krotov and BFGS

As mentioned above (Section II A) the *only* restriction for J_3 (if taken as a penalty) is that it has to be non-positive (for maximization) [34]. The Hessian matrix, $\mathbf{A}$, of a function near its maxima is non-negative, and if one uses the BFGS approximation of the Hessian, $\mathbf{B}$, it will be non-negative everywhere. Hence we do not have to restrict ourselves to the usual form of the penalty function — we can also write Eq. (21d) as:

$$J_3 = -\lambda \sum_{k,j=1}^{n} B_{kj}(\xi_k - \zeta_k)(\xi_j - \zeta_j), \tag{31}$$

where $\lambda > 0$. In the function language of section 2.1 this is similar to claiming that J_3 (Eq. (7d)) can take the form of a quadratic integral, i.e., $J_3 = \int_0^T \int_0^T \lambda(t,t')g[\xi](t,t')dtdt'$. This generalized form for J_3 allows us to incorporate second order information such as $\frac{\delta^2 \bar{J}}{\delta\xi_k \delta\xi_j}$ into the Krotov algorithm.

Using Eqs. (21a - 21c) for J, J_1 and J_2 and Eq. (31) for J_3, Eq. (20) becomes:

$$\begin{aligned} \bar{J} \;=\; & J[v_n] - 2\mathrm{Re}\left\{ \sum_{j=1}^{n-1} \left(u_j^* v_j - u_j^* f_j[\vec{v}, \vec{\xi}] \right) - u_n^* v_n \right\} \\ & -\lambda \sum_{k,j=1}^{n} B_{kj}(\xi_k - \zeta_k)(\xi_j - \zeta_j). \end{aligned} \tag{32}$$

The gradient of $\bar{J}$ (Eq. (23)) is given by:

$$\frac{\delta \bar{J}}{\delta \xi_s} \;=\; 2\mathrm{Re}\left(u_s^* \frac{f_s[\vec{v}, \vec{\xi}]}{\delta \xi_s} \right) - 2\lambda \sum_{j=1}^{n} B_{sj}(\xi_j - \zeta_j). \tag{33}$$

From the maximization condition $\frac{\delta \bar{J}}{\delta \xi_s} = 0$ we get:

$$\sum_{j=1}^{n} B_{sj}(\xi_j - \zeta_j) \;=\; \frac{1}{\lambda}\mathrm{Re}\left(u_s^* \frac{f_s[\vec{v}, \vec{\xi}]}{\delta \xi_s} \right). \tag{34}$$

Our goal now is to get an expression for $\xi_s^{(k)}$, $s = 1, \cdots, n$, i.e. the value of the control of each time point s at the kth iteration. To do this we rewrite Eq. (34) as:

$$\xi_s \;=\; \zeta_s + \frac{1}{B_{ss}} \left\{ \frac{1}{\lambda}\mathrm{Re}\left(u_s^* \frac{f_s[\vec{v}, \vec{\xi}]}{\delta \xi_s} \right) - \sum_{j=1, j\neq s}^{n} B_{sj}(\xi_j - \zeta_j) \right\}, \tag{35}$$

where we have isolated the sth term from the sum. By analogy with the usual Krotov procedure we replace $\vec{v} \to \vec{v}^{(k)}$, $\vec{u} \to \vec{u}^{(k-1)}$ and $\vec{\xi} \to \vec{\xi}^{(k)}$ (see Eq. (30)). But in addition we

replace $\mathbf{B} \to \mathbf{B}^{(k-1)}$, exactly as in the usual BFGS method, since $\mathbf{B}$ is an estimation of the Hessian based on the previous iteration. This leads to the following update rule for the kth iteration:

$$\xi_s^{(k)} \;=\; \zeta_s^{(k)} + \frac{1}{B_{ss}^{(k-1)}} \left\{ \frac{1}{\lambda} \mathrm{Re} \left(u_s^{*(k-1)} \frac{f_s[\vec{v}^{(k)}, \vec{\xi}]}{\delta \xi_s} \right) - \sum_{j=1, j\neq s}^{n} B_{sj}^{(k-1)} (\xi_j^{(k)} - \zeta_j^{(k)}) \right\}. \tag{36}$$

However this update rule raises a problem: the sth component of the updated control depends on later time elements that are as yet unknown (i.e. $\xi_s^{(k)}$ depends on $\xi_j^{(k)}$'s for $j > s$ which are unknown) through the second term in the parentheses of the RHS of Eq. (36).

The solution is based on the idea that, to paraphrase George Orwell's words, "he who knows the past controls the future" [35]. We set $\xi_j = 0$ for all $j > s$, which reduces Eq. (36) to:

$$\xi_s^{(k)} \;=\; \zeta_s^{(k)} + \frac{1}{B_{ss}^{(k-1)}} \left\{ \frac{1}{\lambda} \mathrm{Re} \left(u_s^{*(k-1)} \frac{f_s[\vec{v}^{(k)}, \vec{\xi}]}{\delta \xi_s} \right) - \sum_{j=1}^{s-1} B_{sj}^{(k-1)} (\xi_j^{(k)} - \zeta_j^{(k)}) \right\}. \tag{37}$$

Now all of the elements on the RHS of the equation are known. As s increases we use the past information (all $\xi_j^{(k)}$ where $j < s$, according to Eq. (37)) to determine the present ($\xi_s^{(k)}$) in such a way that the maximization condition of Eq. (34) is obtained. Equation (37) is the central result of this paper. It gives an update rule for the control that combines the monotonicity of the Krotov method with the second order information of the BFGS method. We call the combined method Krotov–BFGS or K–BFGS. In Dirac notation the K–BFGS update rule takes the form:

$$\xi_s^{(k)} \;=\; \zeta_s^{(k)} + \frac{1}{B_{ss}^{(k-1)}} \left(\frac{1}{\lambda} \mathrm{Re} \left\langle \chi_s^{(k-1)} \middle| \frac{f_s[\psi^{(k)}, \vec{\xi}]}{\delta \xi_s} \right\rangle - \sum_{j=1}^{s-1} B_{sj}^{(k-1)} (\xi_j^{(k)} - \zeta_j^{(k)}) \right). \tag{38}$$

Finally, we want to avoid the line-search of Eq. (27) while preserving monotonicity. Therefore, we fix $\vec{s}^{(k)} = \vec{p}^{(k)}$, skipping the line-search of Eq. (27). However, the BFGS update equation for the approximate Hessian (Eq. (28)) is retained.

As before we conclude by giving the algorithm steps. At iteration k:

(i) Using $|\chi^{(k-1)}\rangle$ and the Hessian $\mathbf{B}^{(k-1)}$ from the previous iteration and $|\psi^{(k)}\rangle$ of the present iteration, the improved control $\xi^{(k)}$ is calculated using Eq. (36). The new control is then used to propagate $|\psi^{(k)}\rangle$ according to Eq. (17).

(ii) At time T use Eqs. (20 – 21d) to calculate $\bar{J}^{(k)}$.

(iii) By Eq. (22b) one gets the final condition of the Lagrange multiplier, $|\chi_n^{(k)}\rangle$.

(iv) Use Eq. (22a) to calculate $|\chi_j^{(k)}\rangle$ for all j's, and use Eq. (23) to calculate $\nabla\bar{J}^{(k)}$.

(v) Calculate $\vec{p}^{(k)}$ by Eq. (26) and set $\vec{s}^{(k)} = \vec{p}^{(k)}$.

(vi) Update the Hessian approximation, $\mathbf{B}^{(k+1)}$, by Eq. (28).

The procedure described above is repeated until convergence. (As in the Krotov and BFGS algorithms above, at iteration 0 in step (i) one guesses a control $\xi^{(0)}$ and uses it to propagate $\psi^{(0)}$ up to time T with Eq. (17).) The algorithm is given in a schematic diagram in Fig. 3. Note that the K–BFGS algorithm, like the original Krotov algorithm avoids the need for a line search. As such it has a significantly simpler structure than the BFGS algorithm shown is Section II D.

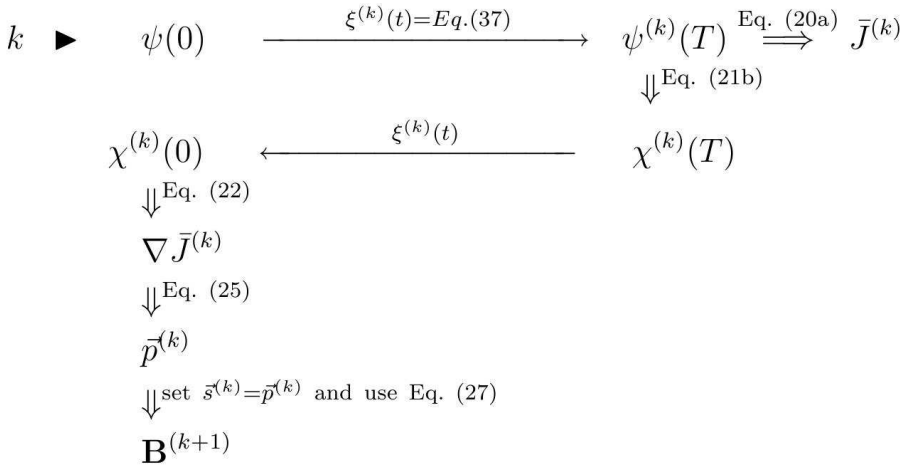

Figure 3: A schematic diagram of the kth iteration in the K–BFGS algorithm.

III. RESULTS

We consider the Na atom, treating its single valence electron fully quantum mechanically, and its core electrons as an external pseudo-potential. This results in a linear Schrödinger equation of motion. We take the control to be an x-polarized electric field $\varepsilon_x(t)$ (we are

using the dipole approximation, i.e. $\varepsilon(x,t) = -\hat{\mu}_x \varepsilon_x(t)$, where $\hat{\mu}_x$ is the x element of the dipole operator). The initial state, $|\psi(0)\rangle$, is taken as the ground-state of sodium.

We choose as the target a state $|\phi\rangle$ that is the sum of two Gaussians, i.e.:

$$\phi(x,y,z) = A(e^{-\alpha_x(x-x_\alpha)^2-\alpha_y(y-y_\alpha)^2-\alpha_z(z-z_\alpha)^2} + e^{-\beta_x(x-x_\beta)^2-\beta_y(y-y_\beta)^2-\beta_z(z-z_\beta)^2}), \tag{39}$$

where $\{\alpha_j\}$, $\{\beta_j\}$, x_α, x_β, etc. are constants and A is a normalization factor. We would like to achieve this target up to a global phase. Then, the objective functional J can be rewritten with the help of a projection operator, P_ϕ:

$$J = \langle\psi(T)|P_\phi|\psi(T)\rangle = \langle\psi(T)|\phi\rangle\langle\phi|\psi(T)\rangle. \tag{40}$$

J_3 was chosen to be zero. Figure 4 shows that J is a monotonic function of the iteration number for the Krotov and K–BFGS algorithms (remember that J and not just $\bar{J}$ is monotonic, see Appendix C). On the other hand, high fidelities are achieved only with the LBFGS and K–BFGS algorithms. Only the K–BFGS algorithm is monotonic and has high fidelity.

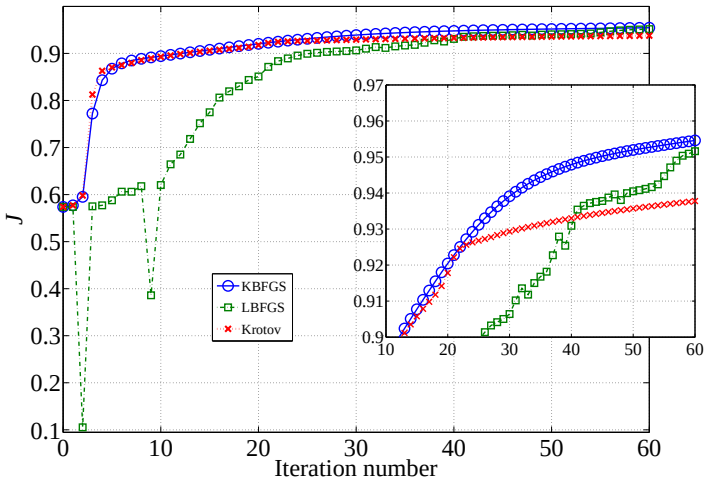

Figure 4: The objective J vs. iteration number for the double-Gaussian target in the sodium atom. $\alpha_x = \alpha_y = \alpha_z = \frac{1}{18}$, $x_\alpha = -x_\beta = 5$, $y_\alpha = y_\beta = z_\alpha = z_\beta = 0$ and $T = 10$fs.

IV. CONCLUSIONS

We started by showing that the low fidelity in the Krotov algorithm arises from its degeneration to a simple gradient search as the algorithm reaches convergence. By rewriting the Krotov update equation for the control such that it includes the Hessian information Eq. (32) (actually the BFGS approximation to the Hessian) we established a variation on the Krotov algorithm that is both monotonic in the iteration number and achieves high fidelities. Although the inclusion of the Hessian raises a difficulty (the dependence of a control component on the future unknown control components, see Eq. (36)), we established a procedure that fulfills the necessary condition for monotonic convergence, Eq. (37). We also provided a detailed description of the K–BFGS algorithm steps.

Regarding the numerical costs, the CPU time for K–BFGS is essentially identical to that of the usual Krotov method, and a factor of m more efficient then the usual BFGS method, where m is the average number of line searches per iteration. The K–BFGS method, as in the usual BFGS method, requires storing the matrix $\mathbf{B}$, the approximate Hessian. The dimension of $\mathbf{B}$ is $\sim N^2$, where N is the number of steps. In the usual BFGS approach, this memory requirement can be avoided using the Limited-memory BFGS (LBFGS) method. A natural extension of the present work would be to constract a Limited-memory K–BFGS (LK–BFGS) method, but this is somewhat specialized task.

For processes such as quantum computing where extremely high fidelities are necessary, as well as for other processes where one seeks high fidelities, the K–BFGS method appears to be a promising alternative to the existing algorithms.

Acknowledgment

This work was supported by the Minerva Foundation and by the EU Marie–Curie–Network EMALI. The authors would like to thank KITP for the generous hospitality during the 2009 program on Quantum Control of Light and Matter during which a collective effort was initiated to compare the efficiency of quantum control algorithms. Particular thanks are due to Dr. Ilya Kuprov who drew our attention to the BFGS method. This research was supported in part by the National Science Foundation under Grant No. PHY05-51164, KITP preprint number: NSF-KITP-10-157. This research is made possible in part by the

historic generosity of the Harold Perlman family.

Appendix A: Deriving the Euler–Lagrange Equations for Quantum Optimal Control

We give here the derivation of Eqs. (8a) and (8b). The variation of $\bar{J}$ (Eq. (7)) with respect to ψ^* is given by:

$$\frac{\delta \bar{J}}{\delta\langle\psi|} \;=\; \frac{\delta J}{\delta\langle\psi|} + \frac{\delta J_1}{\delta\langle\psi|} + \frac{\delta J_2}{\delta\langle\psi|} + \frac{\delta J_3}{\delta\langle\psi|}. \tag{A1}$$

Note that $\frac{\delta J_3}{\delta\langle\psi|} = 0$ since J_3 is independent of $\langle\psi|$. Furthermore J depends only on $\langle\psi(T)|$ so:

$$\frac{\delta J}{\delta\langle\psi|} \;=\; \frac{\delta J}{\delta\langle\psi(T)|}. \tag{A2}$$

Integrating J_1 of Eq. (7b) by parts gives:

$$J_1 \;=\; -2\mathrm{Re}\left(\langle\chi(T)|\psi(T)\rangle - \langle\chi(0)|\psi(0)\rangle - \int_0^T \langle\dot{\chi}(t)|\psi(t)\rangle dt\right). \tag{A3}$$

Therefore (since the initial condition $\psi(0)$ is fixed):

$$\frac{\delta J_1}{\delta\langle\psi|} \;=\; |\dot{\chi}\rangle - |\chi(T)\rangle. \tag{A4}$$

Inserting Eqs. (A2) and (A4) into Eq. (A5) gives:

$$\frac{\delta \bar{J}}{\delta\langle\psi|} \;=\; \frac{\delta J}{\delta\langle\psi(T)|} - |\chi(T)\rangle + |\dot{\chi}\rangle + \frac{\delta J_2}{\delta\langle\psi|}. \tag{A5}$$

Using the necessary condition $\frac{\delta \bar{J}}{\langle\psi|} = 0$ we get Eqs. (8a) and (8b):

$$\begin{aligned} |\dot{\chi}\rangle \;&=\; -\frac{\delta J_2}{\delta\langle\psi|} \\ |\chi(T)\rangle \;&=\; \frac{\delta J}{\delta\langle\psi(T)|}. \end{aligned}$$

In a similar way one can derive Eqs. (22a) and (22b).

Appendix B: Deriving the Krotov Update Rule for The Control Field

We give here a short derivation for the Krotov update rule for linear equation of motion (Eq. (10)). For a more complete treatment, including nonlinear equation of motion, one should look in Ref. [15].

In the Krotov method one needs to defines an auxiliary function:

$$L[\psi,\xi;\phi] = G[\psi(T)] - \int_0^T R[\psi,\xi](t)dt - \phi[\psi(0),0] \tag{B1a}$$

where

$$G[\psi(T)] \;=\; J[\psi(T)] + \phi[\psi(T),T] \tag{B1b}$$

$$R[\psi,\xi] \;=\; \mathrm{Re}\left\{\frac{\partial\phi}{\partial\psi}f[\psi,\xi] - \lambda g[\xi] + \frac{\partial\phi}{\partial t}\right\}, \tag{B1c}$$

and ϕ is a scalar function that depends on ψ that one is free to choose as desired. In reference [15] it is shown that $L = J$, which means that one should maximize G and minimize R. The Krotov algorithm achieves this by an iterative method. At the kth iteration one maximizes R with respect to the control ξ while *at the same time* maximizing G and minimizing R with respect to ψ, i.e.:

$$\xi^{(k)} \;=\; \arg\max_{\xi} R[\psi^{(k)},\xi] \tag{B2a}$$

$$R[\psi^{(k)},\xi^{(k)}] \;=\; \arg\min_{\psi} R[\psi,\xi^{(k)}] \tag{B2b}$$

$$G[\psi^{(k)}] \;=\; \arg\max_{\psi} G[\psi]. \tag{B2c}$$

For linear equations of motion $\phi = \langle\chi|\psi\rangle$ and Eq. (B1c) becomes:

$$R[\psi,\xi] = \mathrm{Re}\left\{\langle\chi|f[\psi,\xi]\rangle - \lambda g[\xi] + \langle\dot{\chi}|\psi\rangle\right\} \tag{B3}$$

Using the necessary conditions $\frac{\delta R}{\delta\psi} = 0$ and $\frac{\delta G}{\delta\psi} = 0$ one obtains Eqs. (8a) and (8b) for $\chi^{(k)}$, to be used in the next iteration. Using the minimization condition $\frac{\delta R}{\delta\xi} = 0$ for R with g given by Eq. (5) one gets Eq. (10):

$$\xi^{(k)} \;=\; \zeta^{(k)} - \frac{1}{\lambda}\left\{\mathrm{Re}\left\langle\chi^{(k-1)}\middle|\frac{\delta f[\psi^{(k)},\xi]}{\delta\xi}\right\rangle\right\}. \tag{B4}$$

Appendix C: Sufficient Conditions for J as well as $\bar{J}$ to Increase Monotonically

We prove here the claim: for the Krotov method (as presented in Section II B), if in the update rule for the control (Eq. (10)) one takes $\zeta^{(k)} = \xi^{(k-1)}$ then J (not only $\bar{J}$) is monotonic with iteration number.

Proof: We can take J_3 to be as in Eq. (6), and rewrite Eq. (7a) as:

$$\bar{J} = J + J_1 + J_2 + J_3 \equiv J' - \int_0^T \lambda(t)\big(\xi(t) - \zeta(t)\big)^2 dt, \tag{C1}$$

where $\lambda(t) > 0$. However, according to Eq. (1) $J_1 + J_2 = 0$; therefore we can rewrite Eq.(C1) as:

$$\bar{J} = J - \int_0^T \lambda(t)\big(\xi(t) - \zeta(t)\big)^2 dt. \tag{C2}$$

If ξ is updated according to the Krotov update rule (Eq. (10)), then for the kth iteration we know that:

$$\bar{J}^{(k)} \geq \bar{J}^{(k-1)} \tag{C3}$$

By inserting Eq.(C2) into Eq. (C3) we get:

$$J^{(k)} - \int_0^T \lambda(t)\big(\xi^{(k)}(t) - \zeta(t)\big)^2 dt \geq J^{(k-1)} - \int_0^T \lambda(t)\big(\xi^{(k-1)}(t) - \zeta(t)\big)^2 dt. \tag{C4}$$

However ζ can be chosen arbitrarily, so we can choose $\zeta = \xi^{(k-1)}$, and we get:

$$J^{(k)} - J^{(k-1)} \geq \int_0^T \lambda(t)\big(\xi^{(k)}(t) - \xi^{(k-1)}(t)\big)^2 dt \geq 0. \tag{C5}$$

Therefore we get:

$$J^{(k)} \geq J^{(k-1)}, \tag{C6}$$

which concludes the proof.

Following the same steps one can prove the above claim also for the K–BFGS update rule (Eq. (37)), where J_3 is written as in Eq. (31).

Appendix D: The Degeneration of the Krotov Method to a First-Order Gradient Method Near Convergence

We prove here that the Krotov method degenerates to a first-order gradient method near convergence. Our starting point is Eq. (15). If we assume that the reference field is the same at all iterations, $\zeta^{(k)} = \zeta^{(k-1)} = \zeta$, we obtain:

$$\xi^{(k)} \;=\; \xi^{(k-1)} + \alpha \frac{\delta \bar{J}^{(k-1)}}{\delta \xi} + 2\alpha \mathrm{Re}\left\langle \chi^{(k-1)} | \mu | \psi^{(k)} - \psi^{(k-1)} \right\rangle. \tag{D1}$$

Moreover, $\mu = \frac{\delta^2 f}{\delta\xi\delta\psi}$ is independent of ξ and ψ since f is linear in both.

The strategy of our proof will be to start from $t = 0$ and show that as $\frac{\delta\bar{J}}{\delta\xi} \to 0$ the Krotov and first-order gradient methods give the same field at all future values of t. According to Eq. (D1) the updated control, $\xi^{(k)}$, at time $t = 0$ is given by:

$$\begin{aligned}\xi^{(k)}(0) &= \xi^{(k-1)}(0) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(0)} + 2\alpha\mathrm{Re}\left\langle\chi^{(k-1)}(0)|\mu|\psi(0) - \psi(0)\right\rangle \\ &= \xi^{(k-1)}(0) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(0)},\end{aligned} \tag{D2}$$

where $\psi^{(k)}(0) = \psi^{(k-1)}(0) = \psi(0)$ is the initial condition of ψ. For small enough Δt Eq. (1) can rewritten as:

$$|\psi(t + \Delta t)\rangle \approx |\psi(t)\rangle + |f[\psi(t), \xi(t)]\rangle\Delta t. \tag{D3}$$

Therefore near convergence $\psi^{(k)}$ at Δt is given by:

$$\begin{aligned}|\psi^{(k)}(\Delta t)\rangle &= |\psi(0)\rangle + |f[\psi(0), \xi^{(k)}(0)]\rangle\Delta t \\ &\overset{\text{Eq. (D2)}}{=} |\psi(0)\rangle + |f[\psi(0), \xi^{(k-1)}(0) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(0)}]\rangle\Delta t \xrightarrow{\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(0)}\to 0} |\psi^{(k-1)}(\Delta t)\rangle.\end{aligned} \tag{D4}$$

Hence the updated control, $\xi^{(k)}$, at Δt is given by (Eq. (D1)):

$$\begin{aligned}\xi^{(k)}(\Delta t) &= \xi^{(k-1)}(\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(\Delta t)} + 2\alpha\mathrm{Re}\left\langle\chi^{(k-1)}(\Delta t)|\mu|\psi^{(k)}(\Delta t) - \psi^{(k-1)}(\Delta t)\right\rangle \\ &\overset{\text{Eq. (D4)}}{=} \xi^{(k-1)}(\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(\Delta t)} + 2\alpha\mathrm{Re}\left\langle\chi^{(k-1)}(\Delta t)|\mu|\psi^{(k-1)}(\Delta t) - \psi^{(k-1)}(\Delta t)\right\rangle \\ &= \xi^{(k-1)}(\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(\Delta t)}.\end{aligned} \tag{D5}$$

Using the above, near convergence $\psi^{(k)}$ at $2\Delta t$ is given by:

$$\begin{aligned}|\psi^{(k)}(2\Delta t)\rangle &= |\psi^{(k)}(\Delta t)\rangle + |f[\psi^{(k)}(\Delta t), \xi^{(k)}(\Delta t)]\rangle\Delta t \\ &\overset{\text{Eqs. (D4) \& (D5)}}{=} |\psi^{(k-1)}(\Delta t)\rangle + |f[\psi^{(k-1)}(\Delta t), \xi^{(k-1)}(\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(\Delta t)}]\rangle\Delta t \\ &\xrightarrow{\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(\Delta t)}\to 0} |\psi^{(k-1)}(2\Delta t)\rangle,\end{aligned} \tag{D6}$$

and therefore $\xi^{(k)}$ at $2\Delta t$ is given by:

$$\begin{aligned}\xi^{(k)}(2\Delta t) &= \xi^{(k-1)}(2\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(2\Delta t)} + 2\alpha\mathrm{Re}\left\langle\chi^{(k-1)}(2\Delta t)|\mu|\psi^{(k)}(2\Delta t) - \psi^{(k-1)}(2\Delta t)\right\rangle \\ &\overset{\text{Eq. (D6)}}{=} \xi^{(k-1)}(2\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(2\Delta t)} + 2\alpha\mathrm{Re}\left\langle\chi^{(k-1)}(2\Delta t)|\mu|\psi^{(k-1)}(2\Delta t) - \psi^{(k-1)}(2\Delta t)\right\rangle \\ &= \xi^{(k-1)}(2\Delta t) + \alpha\frac{\delta\bar{J}^{(k-1)}}{\delta\xi(2\Delta t)},\end{aligned} \tag{D7}$$

and so on until final time T.

From Eqs. (D2), (D5) and (D7) we see that near convergence, as $\frac{\delta \bar{J}}{\delta \xi} \to 0$, the Krotov methods degenerates to a first-order gradient method.

[1] D. J. Tannor and S. A. Rice, *J. Chem. Phys.* **83**, 5013 (1985).

[2] D. J. Tannor, R. Kosloff, and S. A. Rice, *J. Chem. Phys.* **85**, 5805 (1986).

[3] M. Shapiro and P. Brumer, *J. Chem. Phys.* **84**, 4103 (1986).

[4] P. Brumer and M. Shapiro, *Chem. Phys. Lett.* **126**, 541 (1986).

[5] A. P. Peirce, M. A. Dahleh, and H. Rabitz, *Phys. Rev. A* **37**, 4950 (1988).

[6] A. Assion et al., *Science* **282**, 919 (1998).

[7] T. C. Weinacht, J. Ahn, and P. H. Bucksbaum, *Nature* **397**, 233 (1999).

[8] R. Bartels et al., *Nature* **406**, 164 (2000).

[9] P. Corkum, *Nature* **403**, 845 (2000).

[10] R. Kosloff, S. A. Rice, P. Gaspard, S. Tersigni, and D. J. Tannor, *Chem. Phys.* **139**, 201 (1989).

[11] D. J. Tannor and Y. Jin, Design of femtosecond pulse sequences to control photochemical products, in *Mode Selective Chemistry*, edited by B.Pullman, J. Jortner, and R. D. Levine, pages 333–345, Kluwer, Dordrecht, 1991.

[12] V. F. Krotov and I. N. Fel'dman, *Engrg. Cybren* **17**, 123 (1983).

[13] D. J. Tannor, V. Kazakov, and V. Orlov, Control of photochemical branching : Novel procedures for finding optimal pulses and global upper bounds, in *Time-Dependent Quantum Molecular Dynamics*, edited by J. Broeckhove and L. Lathouwers, pages 347–360, Plenum Press, New York, 1992.

[14] J. Somlói, V. A. Kazakov, and D. J. Tannor, *Chem. Phys.* **172**, 85 (1993).

[15] S. E. Sklarz and D. J. Tannor, *Phys. Rev. A* **66**, 053619 (2002).

[16] A. I. Konnov and V. F. Krotov, *Automation Remote Control* **60**, 1427 (1999).

[17] C. P. Koch, J. P. P. R. Kosloff, and F. Masnou-Seeuws, *Phys. Rev. A* **70**, 013402 (2004).

[18] M. Murphy, S. Montanegro, V. Giovannetti, and T. Calarco, *Phys. Rev. A* **82**, 022318 (2010).

[19] J. P. Palao and R. Kosloff, *Phys. Rev. Lett.* **89**, 188301 (2002).

[20] J. P. Palao and R. Kosloff, *Phys. Rev. A* **68**, 062308 (2003).

[21] W. Zhu and H. Rabitz, *J. Chem. Phys.* **109**, 385 (1998).

[22] Y. Maday and G. Turinici, *J. Chem. Phys.* **118**, 8191 (2003).

[23] C. G. Broyden, *J. Inst. Math. App.* **6**, 76 (1970).

[24] R. A. Fletcher, *Comput. Journal* **13**, 317 (1970).

[25] D. A. Goldfarb, *Math. Comput.* **24**, 23 (1970).

[26] D. F. Shanno, *Math. Comput.* **24**, 647 (1970).

[27] Targets that depend on a time interval are also possible in general but we will not include them in the formalism.

[28] In general, one can add more terms to the cost function and each of them can also depend on ψ, but for simplicity we will not include those cases in our discussion.

[29] In certain cases, the penalty function may be part of one's goal, in which case $\bar{J}$ is the real objective and maximization of $\bar{J}$ and not of J is desired. However, most times one aims to maximize J and uses the penalty function only as a mathematical tool to make the objective a convex function.

[30] In principle, a second condition $\frac{\delta J}{\delta|\psi\rangle} = 0$ should also be taken into account but due to the symmetry of J this simply gives the c.c. of Eqs. (8a – 8b).

[31] If ξ is taken to be complex the maximization condition $\frac{\delta \bar{J}}{\delta \xi^*} = 0$ should also be fulfilled. However this condition is often redundant since $\bar{J}$ is symmetric with respect to ξ and ξ^*.

[32] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, *SIAM J. Sci. Comput.* **16**, 1190 (1995).

[33] C. Zhu, R. Byrd, and J. N. P. Lu, *L-BFGS-B: a limited memory FORTRAN code for solving bound constrained optimization problems*, Tech. Report, NAM-11, EECS Department Northwestern University, 1994.

[34] I. Degani, A. Zanna, L. Sælen, and R. Nepstad, *SIAM J. Sci. Comput.* **31**, 3566 (2009).

[35] G. Orwell, *Nineteen Eighty-Four*, Secker and Warburg, London, 1949.