Closed Loop Quantum Control Using High Dimensional Model Representation

Julie Biteen
Princeton University Department of Chemistry
Professor Herschel Rabitz

September 15, 2000

Abstract

In my research this summer, I used closed loop learning in order to create non linear input–output maps for the control of quantum dynamics phenomena. The maps were generated by sampling the output based on a series of amplitude manipulations in the time domain, and the optimal field for each problem was determined based on these maps by using a genetic algorithm. My work showed that an optimal field can be found by generating a series of maps, in an iterative approach.

Introduction

The control of quantum dynamics phenomena is a field which has attracted much attention over the years. [1, 2, 3, 4] Many applications in chemistry, physics and engineering call for controlled microscale chemical and physical processes, yet these quantum processes cannot be observed (and therefore cannot be controlled) in the same manner as larger scale events. Possible applications of quantum control include nonlinear optical processes, atom lasers, quantum electron motion in semiconductors, and quantum computers. The number of processes that depend on quantum control has expanded from the historical goal of developing the means by which to manipulate chemical reactions. However, the possibilities do not end at the present — as the scale of traditional control problems becomes smaller, it will become increasingly important to consider quantum effects on the system.

Most of the work done on quantum control involves manipulating the system's Hamiltonian in order to direct the quantum nature of the subsequent dynamics. Laser fields are generally accepted to be the best form for these applications since they can have complex frequency and time

components that can be adjusted in a way that enables them to be discriminate in their interactions with the system. The electrical fields are designed to work through interference — by constructively enhancing the desired product channels and destructively interfering with undesirable pathways. Physically, this is done by finding an external potential that drives the system to the desired target state.

A quantum system that is independent of any imposed controls will have a wavefunction, $\Psi(t)$, that will evolve in time according to the system's Hamiltonian, H_0. In order for us to dictate the state of $\Psi(t)$ throughout a process or at a target time, T, an external control must be added to H_0. For electric field control, this control term is $-\mu \cdot E(t)$, where μ is the system dipole moment and $E(t)$ is the imposed laser field. The Hamiltonian now becomes $H = H_0 - \mu \cdot E(t)$. The controlled system will evolve according to the time–dependent Schrödinger equation,

$$i\hbar \frac{\partial \Psi}{\partial t} = [H_0 - \mu\varepsilon(t)]\Psi(t) \tag{1}$$

The ability to solve this equation and find the desired $E(t)$ assumes that H_0 and μ are precisely known for the given system. However, this is only true for simple, gas–phase systems containing two or three atoms, and as a result, there is a need for ways to develop accurate maps that can link the input control field to its outcome without the need for a precise determination of H_0 and μ.

Most quantum control applications currently undertaken use optimization techniques such as genetic algorithms or gradient methods.[5, 6, 7, 8, 9] While these techniques can be used to yield the optimal field for a given problem, they have a major drawback. These methods do not serve to teach experimenters about the system at hand. They return only limited information about how the transitions were effected, and each time the target goal is changed, the entire algorithm must be repeated. As a result of this problem, efforts have been undertaken to learn about the system by generating input–output maps as the experiments are performed. Linear mapping procedures that generate a series of maps which approximate the input–output relationship as linear over their domain have been performed, but these maps are only accurate over a small region and therefore cannot generate a single global map.[10] It seems, therefore, that a non linear mapping technique is needed for learning control and for understanding the system being studied.

Some previous efforts to map systems in a non linear manner have chosen molecule–dictated resonance frequencies as constants, and then considered the amplitudes and phases of these frequency components as variables. [11] Sampling the space using this method has allowed for the construction of maps of reasonable accuracy, and laboratory experiments must ultimately be con-

ducted in this way (by varying amplitudes and phases in the frequency domain). However, it was hypothesized that the mapping would be more effective if conducted in the time domain since time can be thought of as the "natural" variable of electric fields.

Main body

My research involves an attempt to produce non linear maps in the time domain. In order to select variables, the field, in the time domain, was discretized into a set of N slices. The amplitude of each slice, x_i, could then be varied within a given domain in a way that allowed for the creation of a series of different electric fields. Schrödinger's equation was propagated stepwise from time $t = 0$ to time $t = T$ over such fields, and a map linking the input field, $E(t)$, to the output populations was constructed from the results.

The creation of non linear maps often runs into the problem of exponential scaling — the number of experiments that one must perform is of the order S^N, where N is the number of variables and S is the number of samples per variable. Since the number of variables tends to increase as a problem becomes more complex, the amount of required experiments can quickly become unmanageable. In my research, I used High Dimensional Model Representation (HDMR) in order to manage one such problem.[12, 13, 14] HDMR is a technique based on the following exact reformulation of the function, $f(\mathbf{x})$:

$$f(\mathbf{x}) \equiv F_0 + \sum_{i}^{N} F_i(x_i) + \sum_{i<j}^{N} F_{ij}(x_i, x_j) + \cdots + F_{12\ldots N}(x_1, x_2, \ldots, x_N) \tag{2}$$

In these examples, I used a formulation of HDMR called cut–center. These maps are constructed over a domain that is centered around a reference field, $\bar{\mathbf{x}}$, called the cut–center. The constant F_0 term of this equation is equal to $f(\bar{\mathbf{x}})$. The first–order (F_i) terms describe the individual contributions of the various x_i components as they are varied while holding all other components at their reference values, and the second–order (F_{ij}) terms show the pairwise contributions of two components, x_i and x_j, as they are varied simultaneously. For most physical systems, equation 2 converges to a very reasonable approximation after only the first– or second–order terms. This greatly reduces the scaling problem since a first–order HDMR map (one using only the F_0 and F_i terms) requires only $(S \cdot N + 1)$ experiments to be performed, and even a second–order map will require only a number of experiments on the order of $O(N^2)$.

The reference Hamiltonian, H_0, has eigenfunctions, $\phi_i(t)$, such that $\Psi(t) = \sum a_i \phi_i(t)$. $\Psi(t)$ is a complex valued function which cannot be directly measured. However, $\mathbf{p}$, the populations at a target time, T, are readily observable quantities that are directly related to $\Psi(T)$: the population in state i, p_i is equal to $\mid a_i \mid^2$. In my research, I performed simulations of population transfer experiments on a system based on the lower Morse Oscillator vibrational energy levels of a hydrogen fluoride molecule.[15] My simulations considered a finite level approximation to the molecule — only the ground state and first three excited states of HF were considered. The transition frequencies ω_{ij} and dipole moments μ_{ij} for this 4–level system are listed in reference [16].

The HDMR maps were constructed beginning from a reference field, $\bar{\mathbf{x}}$, that was designed according to the equation

$$E(t) = A(e^{-\frac{(t-T/2)^2}{2\sigma^2}}) \sum_i b_i \sin(\omega_i t) \tag{3}$$

The frequencies, ω_i, in this Gaussian pulse, were chosen to be the $v \rightarrow v+1$ and $v \rightarrow v+2$ transition frequencies of the HF molecule. The amplitude, A, was equal to $2.0 \cdot 10^{10}$V/m, the final time, T, was equal to 0.5ps, the pulse width, σ, was equal to $T/8.0$. The individual amplitudes, b_i for each frequency component, were varied subject to the constraint $\sum b_i = 1.0$. This reference field was a starting point about which each discrete step of the electric field, x_i, could be varied within a domain that corresponded to a relative difference, as seen in Figure 1.

Once an accurate input–output map is found, a genetic algorithm (GA)[17, 18, 19] is used to search the map space to find an optimal field for any specified target population without the need for further experimentation. In my research, the goal of the simulated population transfer experiments was to move the population from an initial population, $\mathbf{p}_0 = [1, 0, 0, 0]$ to a target population, $\mathbf{p}_{target} = [0, 0, 1, 0]$. The goal of the GA was therefore to find a field, $\mathbf{x}$, that minimized the expression

$$J = \parallel \mathbf{p}_{target} - f(\mathbf{x}) \parallel^2 \tag{4}$$

where $f(\mathbf{x})$ is the output population for the input field, $\mathbf{x}$, as indicated by the HDMR map. This algorithm works by sampling a specified number of fields (here 50) within the domain and assessing their fitness, as measured by equation 4. Fields giving low results to equation 4 are more fit and are therefore used to create new "offspring" fields through recombination, crossing over and mutation, in a way that simulates natural genetics. These offspring fields are tested for their fitness and used to produce another generation. After a specified number of generations (here 100), the field that best minimizes equation 4 is returned as the optimal field. Afterward, the corresponding frequency

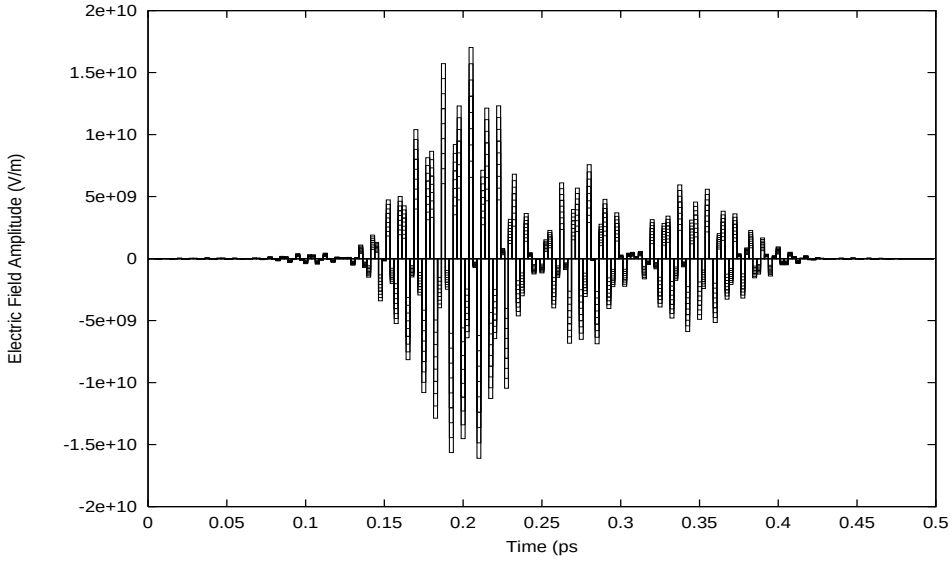

Figure 1: Domain for an HDMR map with reference field $E(t) = A(e^{-\frac{(t-T/2)^2}{2\sigma^2}})(\sin(7.46 \cdot 10^{14}t)/3 + \sin(7.12 \cdot 10^{14}t)/3 + \sin(6.78 \cdot 10^{14}t)/3)$ and relative windows of 50%. Allowable fields are those that can be drawn through the striped region.

domain field, required for the actual controls in a laboratory, is found through a Fourier transform.

In cases where a map is not global (*i.e.* its domain does not include all the fields needed to reach all desirable target populations), a multiple iteration approach to the algorithms described above can be used. Each iteration of such an approach includes creating an HDMR map using a domain with which accurate results can be found, and then running the GA with target, $\mathbf{p}_{target}$. If the GA cannot find a good enough field in the domain for a given iteration, the HDMR domain is moved such that its center corresponds to the best field found by the GA. The next iteration of HDMR mapping and GA optimization will generally find a field that moves the population from $\mathbf{p}_0$ closer to $\mathbf{p}_{target}$. With multiple iterations, the suitable field should eventually be found.

Discussion

Due to the large number of variables involved in the mapping in the experiments conducted on this system, we decided to use only first–order HDMR maps (ones using only the F_0 and F_i terms). Within this limitation, several simulated experiments were performed on the system. The first task

Table 1: Comparing the absolute errors of maps generated using different numbers of time slices

Number of Slices	Test Points with Absolute Error $\leq$ 0.02 (%)
100	75.8
150	61.6
200	62.2
250	68.8
300	71.2
350	70.0

Table 2: Comparing the accuracy of optimal fields found from maps generated using different numbers of time slices

Number of Slices	Actual Population	HDMR–Indicated Population	Percent Error (%)
100	0.77	0.56	37.9
150	0.83	0.73	13.9
200	0.95	0.81	17.1
250	0.96	0.84	13.2
300	0.99	0.77	27.2
350	0.94	0.85	10.1

was to determine a sufficient number of time slices for the mapping. Unlike previously conducted frequency domain mapping attempts, the number of variables needed to map in the time domain could not be determined by the system itself. By repeating the HDMR mapping for different numbers of time slices using a fixed relative window size (50%), it was determined that there was no correlation between a map's absolute error and the number of variables used to create the map (see Table 1). As well, by repeating the optimization for this set of HDMR maps, it was determined that there exists no correlation between the number of time slices used and the accuracy of the optimal field found by the GA (see Table 2).

The number of frequencies used in the control field and the starting amplitudes of each frequency were also varied. Maps constructed about a control field with only the $v \rightarrow v+1$ transition frequencies (ω_{01}, ω_{12}, and ω_{23}) were found to be less accurate than those which also contained the $v \rightarrow v+2$ frequencies (ω_{02} and ω_{13}). However, adding in an ω_{03} component proved unnecessary as it did not add to the map quality. The experiments were then repeated using a zero

reference field and sampling within a Gaussian window, $E(t) = A \cdot e^{-(t-T/2)^2/2\sigma^2}$. Regardless of how many (or how few) time slices were used, a map of adequate quality could not be generated starting with this zero reference field.

Further experiments therefore used the non–zero reference field as in equation 3. Only the $v \rightarrow v+1$ and $v \rightarrow v+2$ frequencies were used to construct this reference field, and each frequency component of the reference field had an amplitude of $a_1 = a_2 = a_3 = a_4 = a_5 = 0.2$. 200 time slices were used for generating HDMR maps in windows around the slices. Maps of reasonable accuracy could be made by using a relative window of 50%. These maps were however not global maps — they spanned a limited number of attainable populations.

If the map quality could be improved, the window size could have been increased which would allow us to use a map that spans a wider dynamic range of attainable populations, and perhaps even a single global map. Several methods of transforming the independent variables ($\mathbf{x}_i$) and dependent variables ($\mathbf{p}_T$) were used in efforts to improve the map quality. The first method was a non–linear transformation of the dependent variables — generating a map that would relate the inputs to the natural logarithm of the populations rather than the populations themselves. This was suggested because in propagating Schrödinger's Equation, the time–dependent wave function $\Psi(t)$ depends on terms like $e^{-iE_it/\hbar}$, and so it was supposed that taking the logarithm of the output values would reduce cooperativity in the mapping relationship.

The second method used was to transform the independent variables into a basis set of orthogonal vectors that were created by taking linear combinations of the original basis set. This was done at first using the moment of inertia matrix, but when this did not lead to improved accuracy, different rotation matrices were chosen using a conjugate gradient minimization algorithm. These methods were attempted since they have been shown to lead to better results in other systems analyzed with HDMR by reducing cooperativity among variables and therefore causing the HDMR terms to converge faster. Unfortunately, none of these methods of change of variable improved the map quality.

Since the map quality could not be improved enough to expand the domain, a global map could not be constructed. We therefore used a series of smaller, non–global maps in an iterative approach. Three iterations of the algorithm were found to be sufficient to find an accurate electric field in order to produce any specified output (see Table 3).

Although results were obtained, the research that I conducted this summer was by no means

Table 3: Percent errors in the second excited state populations indicated by the HDMR maps for a series of HDMR map/GA optimization iterations

Iteration	Actual Population	HDMR–Indicated Population	Percent Error (%)
1	0.715	0.910	27.3
2	0.892	1.000	12.1
3	0.973	0.996	2.4

exhaustive. Other things that can be done in order to improve the results of this algorithm include using a multicut approach to HDMR in which the map is constructed for several cut centers such that any point that is sought can be found using a weighted combination of the results given by each cut center. Random sample HDMR could also be tried; this would involve sampling the domain randomly such that the number of samples needed does not scale at all with the number of time slices used. If an iterative approach was still found to be necessary, the quality of the maps generated after each successive iteration might be improved by continuing to construct the HDMR maps in the time domain but running the GA in the frequency domain. It is also important for us to compare time domain and frequency domain HDMR studies in order to determine which one is more suitable to the HDMR mapping technique and why.

In summary, my research this summer has shown that the iterative method of optimization through a series of maps can be used in the time domain to yield accurate results after only a small number of iterations. Time domain mapping is a valid alternative for producing maps, and should therefore be considered alongside its frequency domain counterpart.

Acknowledgments

The author would like to recognize the support of the Princeton Plasma Physics Laboratory, as well as Professor Herschel Rabitz and JM Geremia.

References

[1] H. Rabitz, R. de Vivie-Riedle, M. Motzjus, and K. Kompa, Science **288**, 824 (2000).

[2] R. Judson and H. Rabitz, Phys. Rev. Lett. **68**, 1500 (1992).

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

[4] M. Shapiro and P. Brumer, J. Chem. Phys. **84**, 4013 (1986).

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

[6] A. Assion, M. Baumbert, T. Bergt, B. Brixner, V. Kiefer, M. Seyfried, M. Strehle, and G. Gerber, Science **282**, 919 (1998).

[7] C. Bardeen, V. Yakovlev, K. Wilson, S. Carpenter, P. Weber, and W. Warren, Chem. Phys. Lett. **280**, 151 (1997).

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

[9] T. Weinacht, J. White, and P. Bucksbaum, J. Phys. Chem. A **103**, 10166 (1999).

[10] M. Phan and H. Rabitz, J. Chem. Phys **217**, 389 (1997).

[11] E. Weiss, J. Geremia, and H. Rabitz **in preparation**.

[12] H. Rabitz, O. Alis, J. Shorter, and K. Shim, Comp. Phys. Comm. **11**, 117 (1999).

[13] O. Alis and H. Rabitz, J. Math Chem. **25**, 197 (1999).

[14] H. Rabitz and K. Shim, J. Chem. Phys. **111**, 10640 (1998).

[15] A. Matsumoto and K. Iwamoto, J. Quant. Spectrosc. Radiat. Transfer **55**, 457 (1996).

[16] $\omega_{01} = 0.746$, $\omega_{02} = 1.457$, $\omega_{03} = 2.135$, $\omega_{12} = 0.712$, $\omega_{13} = 1.390$, and $\omega_{23} = 0.678$. $\mu_{01} = 0.518$, $\mu_{02} = -0.108$, $\mu_{03} = 0.028$, $\mu_{12} = 0.708$, $\mu_{13} = -0.182$, and $\mu_{23} = 0.835$. Frequencies are in rad$\cdot$fs^{-1} and transition dipole moments in 10^{-30}C$\cdot$m.

[17] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, Mass., 1989).

[18] T. Bäck, U. Hammel, and H.-P. Schwefel, IEEE Transactions on Evolutionary Compuation **1**, 3 (1997).

[19] S. Forrest, ed., *Proceedings of the Fifth International Conference on Genetic Algorithms* (Morgan Kaufmann Pub., San Mateo, 1993).