



## Applying artificial neural networks to coherent control experiments: A theoretical proof of concept

Thomas, Esben F.; Henriksen, Niels E.

*Published in:*  
Physical Review A

*Link to article, DOI:*  
[10.1103/PhysRevA.99.023422](https://doi.org/10.1103/PhysRevA.99.023422)

*Publication date:*  
2019

*Document Version*  
Peer reviewed version

[Link back to DTU Orbit](#)

*Citation (APA):*  
Thomas, E. F., & Henriksen, N. E. (2019). Applying artificial neural networks to coherent control experiments: A theoretical proof of concept. *Physical Review A*, 99(2), Article 023422. <https://doi.org/10.1103/PhysRevA.99.023422>

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# 1 Applying Artificial Neural Networks to Coherent Control Experiments: A 2 Theoretical Proof of Concept

3 Esben F. Thomas and Niels E. Henriksen

4 *Department of Chemistry, Technical University of Denmark, Building 206, DK-2800 Kongens Lyngby,*  
5 *Denmark*

We propose a method of experimental coherent control that exploits partial/prior knowledge of a molecular system to efficiently arrive at a solution by using an artificial neural network (ANN) to generate a control field in consecutive temporal steps based on dynamic experimental feedback. Using a 1D double well potential model corresponding to the torsional motion of 3,5-difluoro-3',5'-dibromobiphenyl ( $F_2H_3C_6 - C_6H_3Br_2$ ) to outline and verify our approach, we theoretically demonstrate that an optimized ANN can achieve robust quantum control of nuclear wave packet transfer between wells despite the addition of random perturbations to the simulated molecular potential energy and polarizability surfaces. We suggest that under certain conditions this will also allow the ANN to achieve the stated control objective in an experimental situation. We show that the number of measurements our method requires to generate an optimized field is equal to the dimensionality of the optimization problem, which is significantly less than a naive closed-loop approach would generally need to achieve the same results.

## 6 I. INTRODUCTION

7 The concept of applying ultrashort laser pulses to con-  
8 trol the dynamics of molecular systems has been a topic  
9 of interest for some time. A large body of theoretical (see,  
10 e.g., refs. 1–6) and experimental (see, e.g., refs. 7–13)  
11 work has been produced in which the feasibility of apply-  
12 ing custom-tailored laser pulses to drive various molec-  
13 ular systems into specific target states has been demon-  
14 strated.

15 For simple molecules, where it is possible to make ac-  
16 curate theoretical predictions, a so called “open-loop”  
17 scheme may be employed, where the driving pulse shapes  
18 are designed based on knowledge of the system Hamil-  
19 tonian. However, molecular systems are generally too  
20 complicated for this approach to be of much use. Al-  
21 ternatively, a so called “closed-loop” scheme may be  
22 employed<sup>1,14</sup>. Essentially, the closed-loop approach is  
23 based on the application of a gradient-free optimization  
24 algorithm<sup>15</sup> to the inputs of a pulse shaper<sup>14,16</sup> in a feed-  
25 back loop where the pulse shaper inputs are updated and  
26 optimized “on the fly” based on experimental data gen-  
27 erated by the interaction of the molecular system with  
28 preceding pulses.

29 The efficacy of the closed-loop optimization scheme has  
30 been proven in a number of experiments, including ion-  
31 ization of gas phase diatomic sodium<sup>10</sup>, photoisomeriza-  
32 tion of organic molecules<sup>11</sup>, probing of chemical mecha-  
33 nisms via optimized pulse analysis<sup>12</sup>, and manipulation  
34 of biological proteins<sup>13</sup>. One reason the closed-loop ap-  
35 proach works well in experiments is that it requires little  
36 or no prior knowledge of the system it operates on; it  
37 is essentially a “black-box” approach to coherent control  
38 since the relationship between the problem inputs and  
39 outputs is unknown or does not need to be known.

40 The reality is, of course, that we generally have partial  
41 (but not complete) prior knowledge of any given molecu-  
42 lar system and/or process that we wish to control. This  
43 naturally leads to the question of whether or not it is

44 possible to devise a control scheme that makes use of  
45 this partial information in some way to arrive at a so-  
46 lution more quickly and/or efficiently. Such a scheme  
47 would be useful in situations where, e.g., it takes a long  
48 time to gather the experimental feedback data, since it  
49 would allow us to reduce the total number of measure-  
50 ments required to achieve a desired result.

51 There are numerous ways to implement this idea. One  
52 popular approach is to use a closed-loop scheme where  
53 the inputs to the optimization problem are parameter-  
54 ized based on prior knowledge of the system. For ex-  
55 ample, in ref. 12 it is demonstrated that optimizing the  
56 phase of a transform-limited laser pulse with an evolu-  
57 tionary algorithm (EA) can lead to selective control over  
58 the branching ratio between two competing energy flow  
59 pathways in a bioinspired dyad molecule. The authors  
60 use two different strategies to achieve this; initially they  
61 perform a “blind” or “naive” optimization, i.e. they al-  
62 low the EA to search for a solution without placing any  
63 restrictions on the way the phase function is constructed,  
64 leading to a search space containing 208 parameters. In  
65 the second strategy, they parameterize the phase function  
66 based on a qualitative analysis of the optimized pulse fea-  
67 tures from the initial (unconstrained) approach, resulting  
68 in a reduced search space containing 40 parameters. It  
69 is shown that parameterizing the phase function in this  
70 manner leads to significantly faster convergence, however  
71 the optimized pulse doesn’t perform as well as the pulse  
72 found using the unrestricted approach. This indicates  
73 that properly parameterizing the search space of the op-  
74 timization algorithm can be a challenge, in particular for  
75 complicated systems or processes where it may be diffi-  
76 cult to gain an intuitive understanding of the underlying  
77 control mechanism(s).

78 Our approach to implementing a control scheme that  
79 makes use of previously known information about a given  
80 system is based on the primary ansatz that any discrep-  
81 ancies between the theoretical description of an experi-  
82 ment and what happens in reality can, in principle, be

83 rectified by adding some kind of perturbative term(s) to  
 84 the theoretical model (in section III we will show that this  
 85 is loosely analogous to assuming that the quantum molec-  
 86 ular dynamics can be described using the time-dependent  
 87 self-consistent field (TDSCF) approximation<sup>17</sup>).

88 Based on this premise, we demonstrate theoretically  
 89 how an artificial neural network<sup>18</sup> (ANN) can be trained  
 90 to achieve a desired control objective when applied  
 91 to a theoretical molecular model with randomly per-  
 92 turbed potential and laser-molecule interaction func-  
 93 tions. Furthermore, we suggest that an ANN that has  
 94 been trained in this manner may be able to achieve  
 95 the same control objective in a real experimental situ-  
 96 ation. Note that while ANNs have been used in the past  
 97 to generate predictive models of ultrafast laser-molecule  
 98 interactions<sup>19,20</sup>, to our knowledge they have not been  
 99 applied to coherent control experiments before.

100 While the results in this paper are theoretical, we sub-  
 101 stantiate the general experimental feasibility of our ap-  
 102 proach by demonstrating that the ANN only requires the  
 103 measurements of experimentally observable quantities to  
 104 be able to generate an optimized field that achieves the  
 105 desired control objective. This is accomplished by allow-  
 106 ing the ANN to construct the field directly in the tempo-  
 107 ral domain in consecutive steps, where the amplitude at  
 108 each time step is based on measurements of the system  
 109 at previous time steps. We also demonstrate that the  
 110 number of required measurements is equal to the dimen-  
 111 sionality of the optimization problem (i.e. the number  
 112 of discrete temporal components that characterize the  
 113 shape of the field), which is far fewer measurements than  
 114 would typically be needed if we naively applied a closed-  
 115 loop optimization scheme to the same problem.

116 At this point it is relevant to mention local control  
 117 theory (LCT)<sup>21,22</sup>, a qualitatively similar approach that  
 118 allows for on-the-fly calculation of an electric field based  
 119 on the dynamics of a theoretical model system at each  
 120 time step in a way that leads to a monotonic increase  
 121 (or decrease) in some predefined expectation value. A  
 122 key difference between our method and LCT is that ours  
 123 is intended for use on experimental, real-world systems  
 124 as an alternative to the standard closed-loop approach,  
 125 whereas LCT is generally used as a more efficient alter-  
 126 native to optimal control theory<sup>23</sup> (OCT) when working,  
 127 e.g., with theoretical models that are very computa-  
 128 tionally expensive to simulate.

129 We would like to underline that the work presented in  
 130 this paper is intended as a preliminary proof of concept  
 131 for a general idea: that ANNs or other machine learning  
 132 techniques can be used to increase the speed and effi-  
 133 ciency of determining optimal pulse shapes in coherent  
 134 control experiments. Consequently, we are not suggest-  
 135 ing that the procedure presented here is the best, or only,  
 136 way to implement this concept. Similarly, the model that  
 137 we use to outline and theoretically verify our approach  
 138 is only intended to provide a reasonably plausible exam-  
 139 ple of how our idea might work in a real experiment.  
 140 For this reason, although we attempt to demonstrate

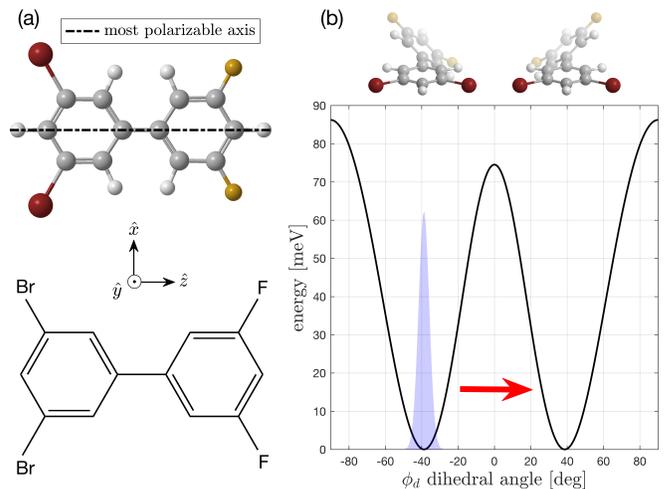


FIG. 1. (a) Molecular structure of the  $F_2H_3C_6 - C_6H_3Br_2$  molecule. The most polarizable axis is also shown (dashed black line). In the simulations performed throughout this article, the MPA is always oriented along the lab frame  $\hat{z}$  axis (b) potential energy surface as a function of the torsional angle  $\phi_d$  between the Br and F substituted rings. The minimum energy nuclear wave packet localized in the left well is shown in light blue, and the right-facing red arrow illustrates the desired control objective; wave packet transfer from the left to the right well using a shaped laser pulse.

141 experimental feasibility when possible by citing relevant  
 142 work, we do not exhaustively consider all of the technical  
 143 challenges that might be associated with this particular  
 144 setup.

## 145 II. MODEL SYSTEM

146 The model we use is based on the torsional poten-  
 147 tial energy surface of 3,5-difluoro-3',5'-dibromobiphenyl  
 148 (which we will henceforth refer to as  $F_2H_3C_6 - C_6H_3Br_2$ )  
 149 in the electronic ground state (see figure 1). In ref.  
 150 24 it is shown that the lowest vibrational mode of this  
 151 molecule corresponds primarily to the torsional motion  
 152 of the phenyl rings. Furthermore, it is shown that by  
 153 aligning the most polarizable axis (MPA) of the molecule  
 154 along the lab-frame  $\hat{z}$  axis (see figure 1) and neglecting  
 155 all higher frequency modes, the Hamiltonian of the sys-  
 156 tem interacting with a non-resonant laser pulse polarized  
 157 in the  $\hat{x}\hat{y}$  plane can be approximated by (using atomic  
 158 units):

$$\hat{H}_{\Phi, \phi_d} = -\frac{1}{2I} \frac{\partial^2}{\partial \Phi^2} - \frac{1}{2I_{\text{rel}}} \frac{\partial^2}{\partial \phi_d^2} + V_{\text{tor}}(\phi_d) - \frac{1}{4} \varepsilon^2(t) \alpha(\Phi, \phi_d), \quad (1)$$

159 where  $\Phi = (\phi_{\text{Br}} I_{\text{Br}} + \phi_{\text{F}} I_{\text{F}}) / (I_{\text{Br}} + I_{\text{F}})$  is the weighted  
 160 azimuthal angle,  $\phi_{\text{Br}}$  ( $\phi_{\text{F}}$ ) and  $I_{\text{Br}}$  ( $I_{\text{F}}$ ) are, respectively,  
 161 the rotational angle and inertial moment of the Br (F)  
 162 substituted ring,  $I = I_{\text{Br}} + I_{\text{F}}$  is the total moment of  
 163 inertia for rotation around the stereogenic axis,  $I_{\text{rel}} =$

164  $I_{\text{Br}}I_{\text{F}}/(I_{\text{Br}} + I_{\text{F}})$  is the relative moment of inertia,  $\phi_d =$   
 165  $\phi_{\text{Br}} - \phi_{\text{F}}$  is the relative torsional angle between the rings,  
 166  $V_{\text{tor}}(\phi_d)$  is the torsional potential energy,  $\varepsilon(t)$  is the time  
 167 dependent electric field of the laser, and  $\alpha(\Phi, \phi_d)$  is the  
 168 molecular polarizability function (the exact forms of the  
 169  $V_{\text{tor}}(\phi_d)$  and  $\alpha(\Phi, \phi_d)$  functions we use, as well as other  
 170 model details, can be found in ref. 25).

171 Note that the first term on the right side of equation  
 172 1 describes the overall rotational kinetic energy of the  
 173 molecule, and the second term describes the ‘‘internal’’  
 174 energy of the torsional oscillations. In addition to induc-  
 175 ing torsional vibrations in the  $\phi_d$  coordinate, driving the  
 176 system with a time-dependent field will generally lead to  
 177 rotation in the  $\Phi$  coordinate as the second most polariz-  
 178 able axis (SMPA, see, e.g., fig. 2 in ref. 24) rotates to  
 179 align with the field polarization axis. However, if we as-  
 180 sume that the SMPA of the molecule is pre-aligned with  
 181 the polarization direction of the driving field, there will  
 182 be very little induced rotation in the  $\Phi$  coordinate. This  
 183 type of 3D orientation/alignment of the MPA and SMPA  
 184 molecular axes can be achieved using an elliptically po-  
 185 larized adiabatic alignment pulse<sup>26,27</sup>. In such a case,  
 186 the Hamiltonian can be reduced to 1D by considering  
 187 the dihedral motion at a fixed  $\Phi$  coordinate, i.e.

$$\hat{H}_{\phi_d} = -\frac{1}{2I_{\text{rel}}}\frac{\partial^2}{\partial\phi_d^2} + V_{\text{tor}}(\phi_d) - \frac{1}{4}\varepsilon^2(t)\alpha(\Phi; \phi_d), \quad (2)$$

188 where the  $\Phi; \phi_d$  notation in  $\alpha(\Phi; \phi_d)$  indicates that  $\Phi$  is  
 189 held fixed over the duration of the pulse. Note that we  
 190 will henceforth always assume that  $\Phi = -4.25^\circ$  (which  
 191 corresponds to alignment of the SMPA with the field po-  
 192 larization direction), and for clarity of notation  $\Phi$  will  
 193 therefore be dropped from subsequent equations.

194 Having defined our model, the control task will be to  
 195 generate a field that can transfer the minimum energy  
 196 wave packet localized in the left well of the system over  
 197 the energy barrier located at  $\phi_d = 0$ , and into the right  
 198 well (see the right side of figure 1).

199 In ref. 24 it is demonstrated that the description of the  
 200  $\text{F}_2\text{H}_3\text{C}_6 - \text{C}_6\text{H}_3\text{Br}_2$  system given by equation 2 will yield  
 201 provisionally accurate results based on comparisons with  
 202 experimental data. However, it is clear that such a dras-  
 203 tic simplification will generally not be able to accurately  
 204 reproduce the real behaviour of the system, particularly  
 205 if the torsional oscillations become very large.

### 206 III. CORRECTING FOR DISCREPANCIES

207 The potential energy  $V_{\text{tor}}(\phi_d)$  and polarizability  $\alpha(\phi_d)$   
 208 functions used in equation 2 are derived from a series  
 209 of quantum-chemical calculations performed in ref. 24  
 210 where the torsional angle of the central C – C bond was  
 211 held fixed at various angles and the remaining structure  
 212 was allowed to relax into the minimum energy configura-  
 213 tion before calculating the energy and polarizability.

214 We can identify at least two sources of error that are  
 215 likely to cause discrepancies in the dynamic behaviour of

216 the experimental system compared to the simulated sys-  
 217 tem; the first is simply due to unavoidable inaccuracies  
 218 associated with any chosen method of quantum-chemical  
 219 calculation, and the second is due to the fact that other  
 220 modes will undoubtedly become activated as the ampli-  
 221 tude of the torsional oscillations become large, which will  
 222 in turn lead to time-dependent distortions in the poten-  
 223 tial energy and polarizability surfaces that the simplified  
 224 Hamiltonian in equation 2 does not account for. We will  
 225 now briefly outline how the TDSCF approximation pro-  
 226 vides a framework allowing us to formally represent the  
 227 influence of these other activated modes in the 1D Hamil-  
 228 tonian from equation 2.

229 Assuming a generalized molecular system is evolving  
 230 on a single electronic potential surface within the Born-  
 231 Oppenheimer approximation (in our case, this is the  
 232 ground state), the TDSCF approximation assumes that  
 233 the total system wave function containing  $N$  nuclear de-  
 234 grees of freedom can be written as a single Hartree prod-  
 235 uct:

$$\Psi(x_1, x_2, \dots, x_N, t) = \prod_{i=1}^N \psi_i(x_i, t). \quad (3)$$

236 In ref. 17, it is shown that this allows us to express the  
 237 Hamiltonian of the  $i^{\text{th}}$  mode as follows:

$$\hat{H}_i^{\text{TDSCF}} = \hat{T}_i + V_i(x_i) + \bar{V}_i(x_i, t), \quad (4)$$

where  $\hat{T}_i$  and  $V_i(x_i)$  represent the kinetic and potential  
 energy, respectively, and where  $\bar{V}_i(x_i, t)$  represents the  
 time-dependent influence (i.e. energy exchange) from all  
 other activated modes. Inspired by this formal treat-  
 ment, we now modify the Hamiltonian from equation 2 by  
 respectively adding time-dependent perturbing functions  
 $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  to the torsional energy function  
 $V_{\text{tor}}(\phi_d)$  and the polarizability function  $\alpha(\phi_d)$ :

$$\hat{H}_{\phi_d}^{\text{TDSCF}} = -\frac{1}{2I_{\text{rel}}}\frac{\partial^2}{\partial\phi_d^2} + [V_{\text{tor}}(\phi_d) + \eta_V(\phi_d, t)] - \frac{1}{4}\varepsilon^2(t)[\alpha(\phi_d) + \eta_\alpha(\phi_d, t)]. \quad (5)$$

238 Now, the  $\eta_V(\phi_d, t)$  term in equation 5 is equivalent to  
 239  $\bar{V}_i(x_i, t)$  in equation 4, however the addition of  $\eta_\alpha(\phi_d, t)$   
 240 to the polarizability function requires further justifica-  
 241 tion. First of all, the form of equation 5 implicitly as-  
 242 sumes that the experimental field/molecule interaction  
 243 is still dominated by the molecular polarizability term,  
 244 which is reasonable provided the experimental laser pulse  
 245 remains in the non-resonant regime and does not be-  
 246 come too intense. Furthermore, it can be shown that  
 247 the energy shifts caused by the polarizability interac-  
 248 tion can be expressed as an expansion of the molecu-  
 249 lar dipole moment onto the unperturbed (i.e. field-free)  
 250 electronic eigenfunctions<sup>28,29</sup>. It is reasonable to assume  
 251 that these electronic eigenfunctions will be modified when  
 252 other molecular modes become activated, and this will in

turn lead to time-dependent discrepancies between the calculated polarizability term  $\alpha(\phi_d)$  and the “real” polarizability of the system that evolve in a way that is qualitatively similar to the perturbations described by  $\eta_V(\phi_d, t)$ . While this somewhat ad-hoc rationalization for adding the  $\eta_\alpha(\phi_d, t)$  term to equation 5 may require further analysis, we feel that it is sufficient for our immediate purposes.

Note that in a sense  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  represent the “difference” between the simulated and experimental systems. The implicit assumption is, therefore, that for a given field  $\varepsilon(t)$ , some set of  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  functions exist that will reproduce the behaviour of the experimental wave packet with perfect accuracy if we insert them into the Hamiltonian in equation 5 and simulate the dynamics.

The problem is still, of course, that we do not know the “correct” form of  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$ . However, provided that all our aforementioned assumptions are valid, there is a non-zero probability that any randomly generated set of  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  functions will reproduce the behaviour of the experimental wave packet. Furthermore, if we can teach a computer to generate a field that accomplishes the simulated control task on the Hamiltonian in equation 5 given *any* set of  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  functions, then it should in principle be able to accomplish the same control task in an experimental situation with no further optimization required. While this may seem like a tall order, if we assume that the method(s) used to calculate  $V_{\text{tor}}(\phi_d)$  and  $\alpha(\phi_d)$  are moderately accurate we can simplify the task by making a few assumptions:

- Structural distortions that occur as the dihedral oscillations become large will primarily be caused by the activation of the other low-frequency modes present in the system. As a result, the temporal variation of the features in  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  will occur on a time-scale that is comparable to the time-scales of these modes.
- As the system interacts with the driving pulse the configuration of the structural distortions will not dramatically fluctuate as the dihedral angle between the rings changes by a small amount. Consequently, the features in  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  will vary relatively smoothly as a function of  $\phi_d$ .
- The amplitudes of the features appearing in  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  are relatively small compared to the characteristic energies (e.g., the potential barrier heights) and polarizabilities of the calculated  $V_{\text{tor}}(\phi_d)$  and  $\alpha(\phi_d)$  surfaces.

Based on these assumptions/simplifications, we will now outline how we generated random  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  perturbing functions. While there are countless ways we can attempt to model these functions depending on how realistic/plausible we want them to be, an

in-depth analysis of this topic is beyond the scope of this paper. For this reason, we have chosen a relatively simple approach based around the application of a Gaussian lowpass filter to a 2D white noise signal, the details of which can be found in appendix A.

Applying this method allows us to generate random  $\eta_V(\phi_d, t)$  perturbing functions consisting of features with an amplitude variance of 11.9 meV, a mean angular coherence length of  $12.2^\circ$  in the  $\phi_d$  dimension, and a mean temporal coherence length of 0.27 ps in the temporal dimension (see appendix A for an explanation of how the coherence lengths are defined). Figure 2 shows 2D plots of  $V(\phi_d)$  combined with an example of a randomly generated  $\eta_V(\phi_d, t)$  function to demonstrate how these perturbations will modify the potential energy surface. The  $\phi_d$  and  $t$  coherence length parameters we use to generate the random  $\eta_\alpha(\phi_d, t)$  perturbing functions are identical to the ones used for  $\eta_V(\phi_d, t)$ , and the amplitude variance parameter for  $\eta_\alpha(\phi_d, t)$  has been chosen such that the amplitude variance of the  $\varepsilon^2(t)\eta_\alpha(\phi_d, t)/4$  term in equation 5 is equal to 11.9 meV when the amplitude of  $\varepsilon(t)$  is at its maximum allowed value.

The choice of mean coherence lengths for the temporal features in  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  is roughly based on a normal mode analysis performed in ref. 24, and the corresponding mean coherence lengths in the angular dimension are roughly based on a potential energy surface calculation performed in ref. 30. Furthermore, we tuned the amplitude variance parameters of  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  to be as large as possible while still allowing our approach to yield good results, and in section V we will demonstrate that this resulted in perturbation amplitudes that are a nontrivial task for the ANN to deal with.

#### IV. IMPLEMENTING AND OPTIMIZING THE ANN

As outlined in section III, the goal is to teach a computer to achieve the control task described in section II on the system represented by equation 5 for any random set of perturbing  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  functions. In this section we will demonstrate how this can be accomplished using an ANN combined with a genetic algorithm (GA)<sup>15</sup> using a technique called neuroevolution<sup>31</sup>.

ANNs and similar machine learning techniques are currently a hot topic in a variety of fields. Because the literature related to this topic is already quite extensive (see, e.g., refs. 18–20, 31–33), we will here only provide a brief general description of ANNs and their operating principles.

An ANN is essentially a mathematical function that can be characterized by a network of directionally linked nodes connected to a set of network input and output vectors. Each node in the ANN consists of a so-called “activation function” that receives a series of node inputs and generates a node output based on their weighted

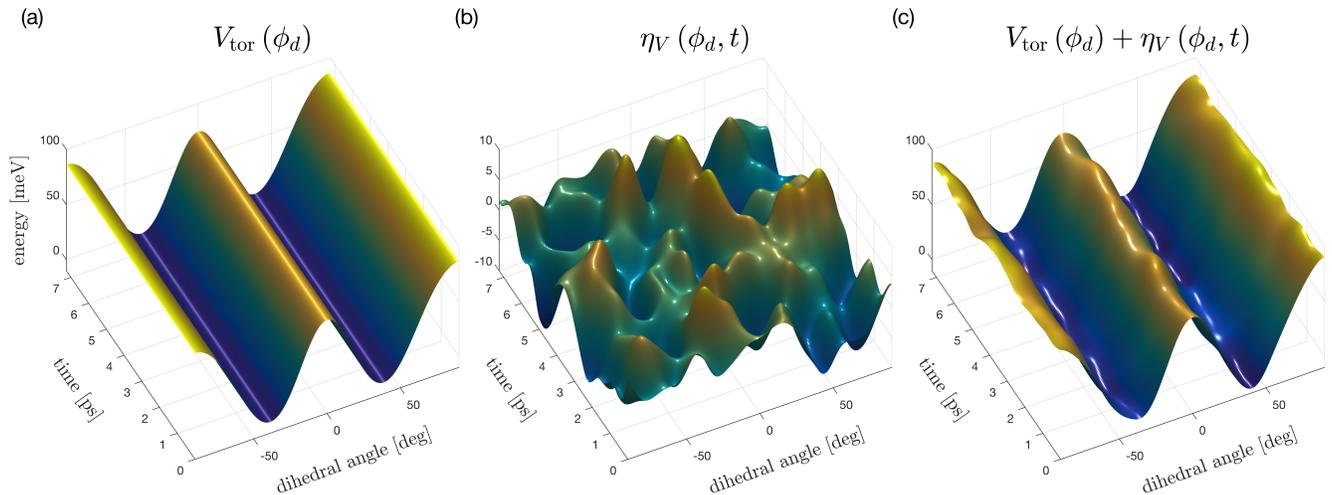


FIG. 2. (a) the theoretical torsional potential energy function  $V_{\text{tor}}(\phi_d)$ . (b) Example of the perturbing  $\eta_V(\phi_d, t)$  function displaying the characteristic size scale of the perturbations in the temporal and spatial domains. (c) When  $\eta_V(\phi_d, t)$  is added to  $V_{\text{tor}}(\phi_d)$ , the torsional potential is perturbed in time and space.  $\eta_\alpha(\phi_d, t)$  perturbs  $\alpha(\phi_d)$  in a similar fashion (not shown).

362 sum, i.e.

$$f = K \left( \sum_i w_i p_i + w_b \right), \quad (6)$$

363 where  $p_i$  represents the “raw” value from the  $i^{\text{th}}$  incoming  
 364 connection,  $w_i$  is the corresponding connection weight,  
 365  $w_b$  represents the contribution from a constant “bias”  
 366 input (see, e.g., figure 3), and  $K$  is the activation func-  
 367 tion that maps  $\sum_i w_i p_i$  to scalar node output  $f$ . Each  
 368 node input  $p_i$  comes either from other nodes within the  
 369 network, or from “outside” the network as part of the  
 370 network input vector. Likewise, each node output can  
 371 connect to other nodes within the network and/or to the  
 372 network output vector. Evaluation of a given network  
 373 input by the ANN is achieved by propagating the “sig-  
 374 nal” from the network input vector through the network  
 375 nodes until it reaches the network output vector.

376 An ANN can “learn” generalized relationships between  
 377 the inputs and outputs associated with a given problem  
 378 or task. This is accomplished by optimizing all the inter-  
 379 nal connection weights until the ANN consistently pro-  
 380 duces the “correct” output for any relevant input. In  
 381 many cases, the connection weights can be optimized by  
 382 gradient descent using backpropagation<sup>33</sup>, however this  
 383 method requires access to a training set of valid input-  
 384 output pairs. Neuroevolution avoids this issue by us-  
 385 ing a GA to optimize the network connection weights  
 386 instead, where each candidate network receives a fitness  
 387 score based on how successful it is at performing a desired  
 388 behaviour or task. This makes neuroevolution particu-  
 389 larly useful for reinforcement learning problems where  
 390 the correct network outputs for any given set of network  
 391 inputs may not be known. Note that more advanced neu-  
 392 roevolution algorithms will evolve both the topology of  
 393 the network and its weights<sup>34</sup>, however we will not be

394 making use of this approach.

395 We will now describe our operational approach to us-  
 396 ing an ANN to generate a field based on dynamic feed-  
 397 back from an arbitrarily perturbed  $\text{F}_2\text{H}_3\text{C}_6 - \text{C}_6\text{H}_3\text{Br}_2$   
 398 system, and we will demonstrate how we used neuroevo-  
 399 lution to optimize the ANN connection weights. The  
 400 temporal pulse envelope is characterized by a series of  $N$   
 401 discrete, equally spaced regions or “bins” with width  $\delta t$   
 402 and total length  $N \times \delta t = T$ . The ANN assigns the am-  
 403 plitude of the bin at time step  $n+1$  with a constant value  
 404 based on information about the system behaviour from  
 405 time steps 0 to  $n$ . In practice, the pulse time window was  
 406 set to  $T = 7.25$  ps, and the number of field components  
 407 was set to  $N = 300$ , i.e. each field component had a  
 408 width of  $\sim 24$  fs, significantly shorter than the  $\sim 1200$  fs  
 409 vibrational period of the system. As we will demonstrate  
 410 in section V, this step size is small enough to allow the  
 411 network to tailor fields that perform well on specific sets  
 412 of perturbations.

413 We now outline the general procedure. First, let  $\varepsilon_n$  and  
 414  $\langle \phi_d \rangle_n$  denote the respective field amplitude and position  
 415 expectation value at the  $n^{\text{th}}$  time step, and let  $\mathbf{M}_n$  denote  
 416 the list of positions between time step 0 and  $n$ , i.e.

$$\mathbf{M}_n = \{ \langle \phi_d \rangle_0, \langle \phi_d \rangle_1, \dots, \langle \phi_d \rangle_{n-1}, \langle \phi_d \rangle_n \}. \quad (7)$$

417 Now, let  $F(\subseteq \mathbf{M}_n)$  denote the evaluation of the ANN  
 418 when it receives a subset of the information in  $\mathbf{M}_n$  as  
 419 input(s). At each general time step  $n$ , the amplitude of  
 420 the subsequent field component ( $\varepsilon_{n+1}$ ) is constructed as  
 421 follows:

$$\varepsilon_{n+1} = F(\subseteq \mathbf{M}_n). \quad (8)$$

422 Next,  $\varepsilon_{n+1}$  is appended to the total field shape, the sys-  
 423 tem wave packet is propagated forward from time step  
 424  $n$  to time step  $n+1$  using split-operator propagation<sup>35</sup>,

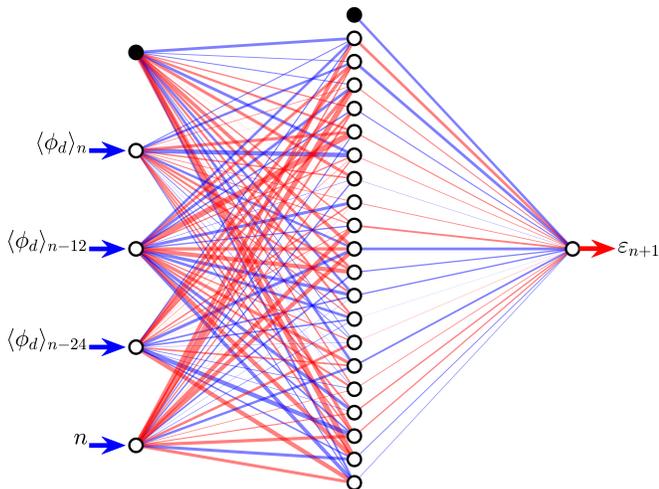


FIG. 3. Sketch of the optimized feedforward network with 4 inputs and a single hidden layer with 20 nodes. The blue (dark gray) and red (light gray) lines respectively indicate positive and negative connection weights, and the line thicknesses correspond to the absolute weight magnitude (for reference, the mean absolute weight magnitude is 3.6, and the maximum absolute weight magnitude is 11.1). All hidden and output nodes contain tanh activation functions, and the black nodes at the top of the structure are bias nodes set to constant output 1. The bottom input, labelled  $n$ , inputs the current time step. All inputs are pre-processed by scaling them to a range between approximately  $-1$  and  $1$ , and the network output is scaled to a value between  $0$  and the (user-defined) laser intensity cutoff limit.

and the expectation value of the new wave packet position  $\langle \phi_d \rangle_{n+1}$  is calculated. To further clarify, an illustration demonstrating how the ANN uses the measurements from previous time steps as inputs to determine the amplitude of the next portion of the field is shown in figure 4. Note that in the corresponding experimental situation the overall pulse shape would be updated to include the appended component, the new pulse would be applied to the molecules, and a new measurement of the system would be performed at the appropriately updated time step, as shown in figure 5.

The new positional information is added to  $\mathbf{M}_n$  (which now becomes  $\mathbf{M}_{n+1}$ ), and the ANN is reapplied to determine the field amplitude at time step  $n+2$ , i.e.

$$\varepsilon_{n+2} = F(\subseteq \mathbf{M}_{n+1}). \quad (9)$$

Iterating this procedure  $N$  times allows the ANN to construct the entire field envelope in consecutive steps based on dynamic feedback from the system. Note that in practice we “seeded” the dynamics by uniformly setting the field amplitude at the first 10 time steps to the maximum value, as it was found that this led to improved performance.

The choice of network topology (i.e. the number of nodes in the network and their connectivity) and the type of activation function(s) used in the network nodes

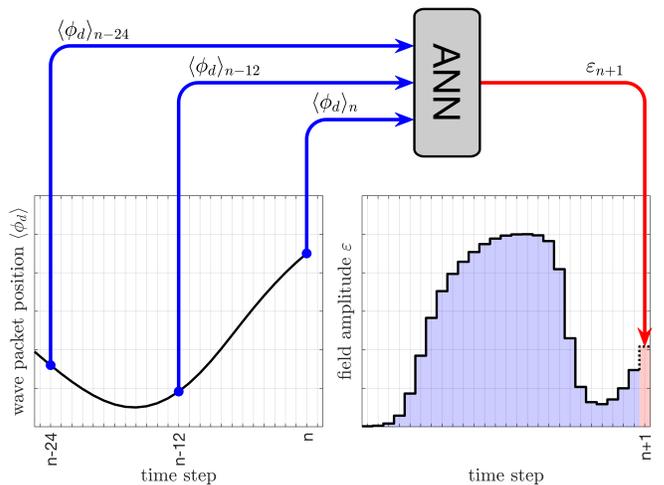


FIG. 4. Schematic showing how information from time steps  $0$  to  $n$  is used by the ANN to determine the field amplitude at time step  $n+1$ . The left and right panels respectively show the wave packet position  $\langle \phi_d \rangle$  and the field amplitude  $\varepsilon$  at an (arbitrary) interval between time steps  $n-15$  and  $n+1$ . As stated in the article text, once the ANN determines the amplitude of  $\varepsilon_{n+1}$ , the new component is added to the total field and the system is propagated forward from time step  $n$  to  $n+1$ . Finally, a new measurement of the wave packet position at time step  $n+1$  is made, and the process is repeated.

(see, e.g., equation 6) can significantly impact the quality of the results. We found that a simple feedforward configuration<sup>32</sup> with a single hidden layer with 20 nodes containing tanh activation functions yielded good results. A sketch of the topology and connection weights of an optimized network is shown in figure 3 (note that since the ANN uses information from previous time steps it might formally be classified as a *recurrent* neural network<sup>36</sup>, however for our purposes this distinction is moot).

In an attempt to minimize the number of network inputs (and thereby limit the number of connection weights that need to be optimized), we assume that only recent information about the position of the wave packet is relevant for informing the ANN what to do next at any given time step. For this reason, inputs from  $\mathbf{M}_n$  were specifically chosen as a series of  $P$  datapoints going “back in time” from the most recent measurement, equally spaced at interval  $K$ , i.e.

$$\subseteq \mathbf{M}_n = \{\langle \phi_d \rangle_n, \langle \phi_d \rangle_{n-K}, \langle \phi_d \rangle_{n-2K}, \dots, \langle \phi_d \rangle_{n-(P-1)K}\}. \quad (10)$$

Based on this general approach, we tuned the input parameters by systematically optimizing the ANN with different types of input configurations (specifically, we tested combinations of  $P = 1, 2, 3, 4$  and  $K = 1, 6, 12, 18$ ). We found that the ANN performed best when  $P = 3$  and  $K = 12$ , i.e. when the network inputs are given by  $\langle \phi_d \rangle_n$ ,  $\langle \phi_d \rangle_{n-12}$  and  $\langle \phi_d \rangle_{n-24}$ , as shown in figure 3.

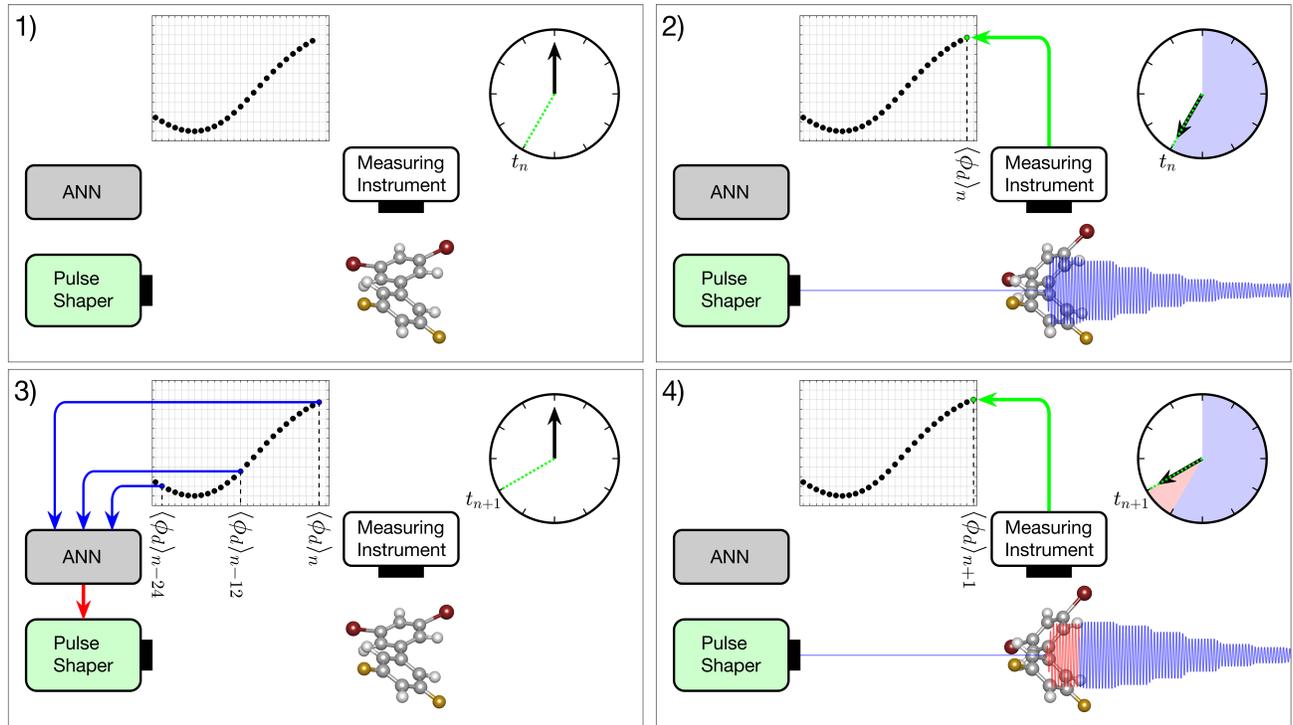


FIG. 5. Illustration of how the trained ANN can be implemented to construct an optimized field in an experimental situation. Note that this figure is meant to show a single intermediate iteration at the  $n^{\text{th}}$  step of a process that has already been repeated  $n - 1$  times beforehand. 1) At time step  $n$  the measurement acquisition time is set to  $t_n$  (where  $t_n = n \times \delta t$ ), as represented by the dotted green line on the clock. 2) The current form of the shaped pulse is applied to the experimental system, a measurement of the system is performed when  $t = t_n$ , and the new measurement data is added to the full set of information about the system from time steps 0 to  $n$ . 3) The measurement acquisition time is updated to  $t_{n+1}$ , and a subset of the list of measurement data is used as inputs to the ANN, which in turn informs the pulse shaper what the field amplitude at time step  $n + 1$  should be. 4) The pulse shaper generates a new pulse identical to the former albeit with the newly appended field component appearing between  $t_n$  and  $t_{n+1}$  (shown in red), and a new measurement is performed when  $t = t_{n+1}$ . At this point the value of the current time step is increased by 1 and steps 3 – 4 are repeated until  $N$  time steps have passed.

475 We will now explain how we used a GA to optimize 496  
 476 the network connection weights in order to achieve the 497  
 477 stated control objective. Note that the feedforward network 498  
 478 we wish to train contains a total of 121 connection 499  
 479 weights (see figure 3). This means that we will essentially 500  
 480 be using the GA to solve a 121 parameter optimization 501  
 481 problem, i.e. each candidate GA solution is represented 502  
 482 by a “genome” consisting of 121 double-precision float- 503  
 483 ing point numbers. Each number in a given genome 504  
 484 defines a unique network connection weight within the 505  
 485 predefined network topology. For this reason, a given 506  
 486 genome can be used to generate its corresponding “phe- 507  
 487 notype” by mapping its values to the weights of a net- 508  
 488 work, and conversely the “genotype” of a given network 509  
 489 can be extracted by mapping its connection weights to 510  
 490 the corresponding genome positions. In the following 511  
 491 outlined steps it should therefore be understood that 512  
 492 the terms “genome” and “network” essentially mean the 513  
 493 same thing, and will be used interchangeably depending 514  
 494 on context. 515

495 1. Generate  $S$  random “training” systems consist-

ing of perturbed potential functions  $V_s(\phi_d, t) = V_{\text{tor}}(\phi_d) + \eta_V(\phi_d, t)$  and the corresponding perturbed polarizabilities  $\alpha_s(\phi_d, t) = \alpha(\phi_d) + \eta_\alpha(\phi_d, t)$ .

2. Define the initial wave packet configuration of each training system,  $\Psi_s(\phi_d, t = 0)$ , by calculating the minimum energy state localized in the left well of  $V_s(\phi_d, t = 0)$  using the Fourier grid Hamiltonian<sup>37</sup> (FGH) method.
3. For each training system, define a set of 10 target states,  $\bar{\chi}_s = [\chi_{s,0}, \chi_{s,1}, \dots, \chi_{s,9}]$ , by using the FGH method to calculate the 10 lowest energy states localized in the right well of  $V_s(\phi_d, t = T)$ .
4. Create a random initial “population” of  $M$  networks where, as stated, the genome of each network is characterized by a list of values that each define a unique connection weight in the network.
5. Apply the  $m^{\text{th}}$  network to all  $S$  training sets in the manner outlined previously in this section, resulting in  $S$  different wave packets propagated to time  $T$  by the network generated fields,  $\Psi_{m,s}(\phi_d, t = T)$ .

- 516 6. Assign the  $m^{\text{th}}$  network a fitness score  $F_m$ , defined  
 517 as the mean overlap of all  $\Psi_{m,s}(\phi_d, t = T)$  from step  
 518 5 with the target states in  $\bar{\chi}_s$ , i.e.

$$F_m = \frac{1}{S} \sum_{s=1}^S \sum_{k=0}^9 |\langle \Psi_{m,s}(\phi_d, t = T) | \chi_{s,k} \rangle|^2 \quad (11)$$

- 519 7. Repeat steps 5 – 6 for all  $M$  networks, and  
 520 use the GA to create a population of new net-  
 521 works/genomes by mutating and cross breeding  
 522 networks/genomes from the current generation  
 523 with higher fitness scores (see appendix B for de-  
 524 tails about our GA implementation).

- 525 8. Repeat steps 5 – 7 until the maximum fitness level  
 526 of the population converges and/or ceases to signifi-  
 527 cantly improve.

528 By evaluating the performance of the networks on the  
 529 same training systems every generation, we ensure that  
 530 the convergence is monotonic (this would not be the case  
 531 if we, e.g., created a new set of training systems for each  
 532 new generation). The caveat of this approach is that we  
 533 must include a set of training systems that is large enough  
 534 to prevent overfitting; i.e. if we use too few training  
 535 systems, then it is unlikely that a network will be able  
 536 to learn the general rules it needs to know to be able to  
 537 successfully tackle a system that isn't part of the training  
 538 set.

539 In practice, the appropriate number of training sys-  
 540 tems was estimated through trial and error by cross-  
 541 validating the performance of the converged network on  
 542 a series of  $10^5$  random new systems generated using the  
 543 same noise parameters as the training set (i.e. the same  
 544 mean temporal/angular coherence lengths and amplitude  
 545 variances as outlined in section III). We found that us-  
 546 ing a training set containing  $S = 100$  different systems  
 547 yielded very similar training and cross-validation scores,  
 548 indicating that this is a reasonable size (another com-  
 549 mon strategy for avoiding overfitting is to monitor the  
 550 cross-validation error at every iteration and halt the op-  
 551 timization once this value begins to increase, however as  
 552 our initial approach seems to work well enough we did  
 553 not find it necessary to try other methods).

554 As a final aside, it is important that the range of the  
 555 initial guesses provided by the GA, as well as the size of  
 556 the GA mutations, are scaled to reflect the range where  
 557 the tanh activation function changes from  $-1$  to  $1$ . In our  
 558 optimization, the range of the initial weights was between  
 559  $-6$  and  $6$ , and the weights were mutated by adding a  
 560 random Gaussian variable with  $0$  mean and a standard  
 561 deviation that did not exceed  $0.6$ .

## 562 V. RESULTS AND DISCUSSION

563 Using the methodology outlined in section IV we op-  
 564 timized a network using a GA population of  $M = 300$ .

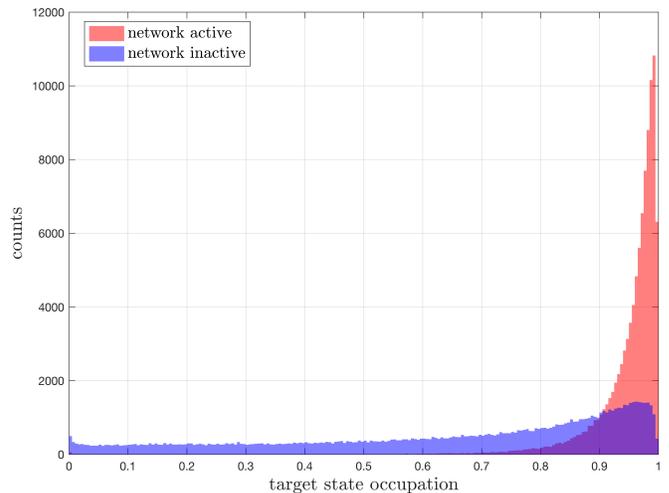


FIG. 6. The strongly peaked red histogram shows the distri-  
 bution of target state occupations when the optimized net-  
 work is applied to  $10^5$  random test systems (note that these  
 systems are not included in the set that was used to train the  
 network). The flatter blue histogram shows the correspond-  
 ing distribution when the  $10^5$  pulses generated by the network  
 in the aforementioned analysis are used to drive different sys-  
 tems than the ones they were intended for. (Note that the  
 red and blue histograms have been visually overlaid, i.e. the  
 darker area in the lower right corner is where the shapes of  
 the two distributions overlap.)

565 The network outputs were scaled to a value between  $0$   
 566 and a peak pulse intensity of  $20 \text{ TW/cm}^2$  (note that ex-  
 567 perimental evidence suggests that this intensity will not  
 568 ionize the molecules<sup>27,38</sup>, nevertheless eventual ionization  
 569 issues may be remedied by increasing the length of the  
 570 pulse and decreasing the allowable peak intensity).

571 Figure 3 shows a sketch of this optimized ANN. As  
 572 stated at the end of section IV, we cross-validated the  
 573 performance of this network by applying it to  $10^5$  new  
 574 random systems generated using the same noise param-  
 575 eters as the training sets, and calculated the resulting  
 576 overlap of the propagated wave packets with the target  
 577 wave functions. The red histogram in figure 6 shows the  
 578 distribution of the target occupation levels for all  $10^5$   
 579 cross-validation measurements. It is clear that the net-  
 580 work is quite effective at achieving the control objective  
 581 when dealing with perturbed systems that it hasn't en-  
 582 countered before, as the distribution is strongly peaked  
 583 with a mean value of  $0.95$ .

584 Now, it is possible that we have created a network  
 585 that simply produces generalized pulse shapes that work  
 586 well across all perturbed systems, i.e. the ANN may not  
 587 actually be making “intelligent” decisions based on the  
 588 immediate behaviour of the wave packet in any given sys-  
 589 tem (in particular, this would be true if the perturbation  
 590 amplitudes were too small to have a significant impact  
 591 on the wave packet dynamics). To test this hypothesis,  
 592 the  $10^5$  pulses that the network generated in the previ-  
 593 ous cross-validation analysis were again applied to the

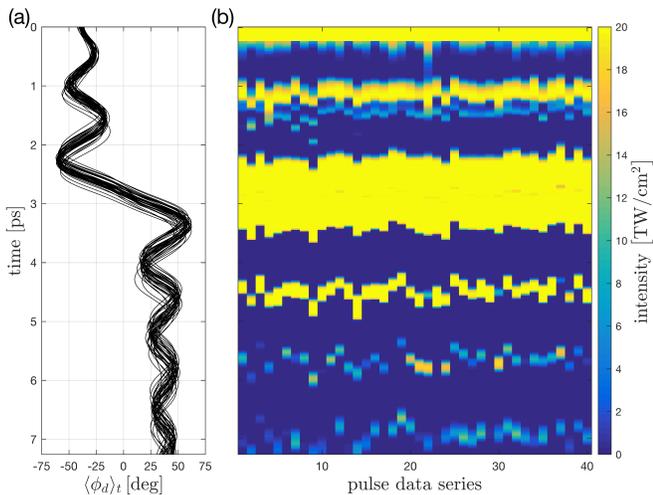


FIG. 7. (a) Overlaid time-dependent position expectation values of wave packet trajectories generated by applying the optimized ANN from figure 3 to 40 different test systems being perturbed by different  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  functions created using the parameters outlined in section III. (b) top-down view of the corresponding 40 optimized pulse envelopes generated by the ANN.

10<sup>5</sup> randomly generated test systems, except this time each pulse was applied to a different system than the one it was originally intended to work on. It is reasonable to assume that if the shapes of the pulses generated by the network are indeed not contingent on the specific perturbed system, then a given pulse should work more or less equally well on any test system that we apply it to. The blue histogram in figure 6 shows the distribution of target occupation levels when we tested the pulses in this manner; this means that in a sense the difference between the red and blue distributions illustrates the degree to which the network is creating pulses that are specifically tailored to the unique set of random perturbing functions associated with any given test system. The fact that this distribution is relatively flat compared to the red distribution shows that there is a significant loss in performance when the network is not allowed to react to system specific feedback, i.e. the pulse shapes are not “trivial”.

To further analyze the behaviour of the optimized ANN, we applied it to a system where the perturbations had been switched off, leading to a target state occupation of 0.995. We also tried applying the original 10<sup>5</sup> pulses to the unperturbed potential, yielding a mean target state occupation of 0.87. Finally, we used the network to generate 10<sup>5</sup> new pulses on a series of systems where the perturbation amplitude had been decreased by 50%. This led to a mean target state occupation of 0.99. These results indicate that the network has learned something about the “general” unperturbed case even though this was not included in the original training set.

Figure 7 shows a comparison of 40 pulse envelopes created by the network when it was applied to a series of sys-

tems perturbed by different  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  functions, again generated using the same noise parameters as before. By comparing the similarities and differences between various pulses, we can gain further insight into the general principles of network operation. The right side of figure 7 shows a “top down” view of the pulses, and the left side shows the 40 overlaid trajectories of the corresponding wave packet expectation values. Here it can be seen that each optimized field broadly consists of a number of pulses appearing at similar times, and the wave packet trajectories all follow similar paths. In general, the initial “seed” pulse at  $t = 0$  and the following pulse are responsible for pumping the amplitude of the dihedral oscillations for two periods in the left well, the large third pulse is responsible for transferring the wave packet over the central energy barrier at  $\phi_d = 0$ , and the last 2–3 pulses are used to dampen the amplitude of the oscillations in the right well. Despite these overarching similarities, figure 7 also illustrates that there are differences between the separate systems. Specifically, there are noticeable variations in the temporal and spatial locations of the turning points of the oscillations, as well as the temporal locations of the rising and falling edges of the pulses.

Note that the average pulse duration consists of multiple discrete field components, suggesting that it might be possible to further reduce the required number of measurements by increasing the time step size. However, doing this is likely to be detrimental to overall performance since the network will lose the ability to precisely control the position of the rising and falling edge of each pulse. Conversely, we could increase the number of time points, e.g. by using 600 time points instead of 300 (provided we appropriately adjust the spacing of the network inputs). However, close inspection of the pulse structures in figure 7(b) (or on the right side of figure 4) indicate that the rising and falling edges of the pulses are generally “smooth”, i.e. each edge consists of multiple intermediate timesteps. This indicates that increasing the “sampling rate” in the aforementioned manner will probably not significantly improve the quality of the results, i.e. the current number of time steps appears to be large enough to capture the variations that allow a generated field to perform well on the perturbed system it is tailored to.

Note also that these results are achieved after performing a total of 300 measurements on each system, which, as stated in section I, is equal to the number of free parameters used to characterize the shape of the field. A naive/blind closed-loop approach to the same problem would be, e.g., to use a GA to individually optimize the temporal components of the field instead, leading to a search space containing 300 parameters. As a rule of thumb, the population of a GA (and therefore the number of measurements performed per iteration/generation) should be proportional to the dimensionality of the search space. Furthermore, the GA will generally require multiple iterations before finding a good solution (e.g. a 400 parameter optimization performed in ref. 29 required

346 iterations of a population containing 2000 individuals, meaning a total of 692000 theoretical “measurements” had to be performed before a converged solution was found). Therefore, optimizing the field shape using a naive closed-loop scheme in the aforementioned manner will likely require a number of measurements that is multiple orders of magnitude larger than our method requires. As a final comment on this topic, the intuitive simplicity of the pulse shapes in figure 7 suggest that it would be relatively easy to reparameterize the search space and significantly reduce the dimensionality of the optimization problem, e.g. as discussed in section I. For this reason, a properly parameterized closed-loop optimization of this particular system would probably require far fewer measurements than the blind approach we have just outlined. However, as also discussed in section I, manually determining a suitable parameterization for any given system is not always trivial. In this respect the methodology outlined in this paper has a distinct advantage, since in a sense training the ANN automatically parameterizes the problem for us.

While the complexity of the connections in figure 3 make it difficult to ascertain exactly how the network uses the inputs to make decisions, we can make a few educated guesses based on the input data characteristics. As stated in section IV, it was discovered through trial and error that the ANN performs best when it receives the wave packet positions at  $\langle\phi_d\rangle_n$ ,  $\langle\phi_d\rangle_{n-12}$  and  $\langle\phi_d\rangle_{n-24}$ . The shape of the wave packet trajectory on the left side of figure 4 shows that the temporal spacing between these three measurements is similar to the time scale of changes in the wave packet trajectory. **This indicates that the chosen input spacing works well because it makes it easier for the network to capture “higher-order” information about the dynamics (i.e. is the wave packet currently decelerating/accelerating, has it reached a turning point, etc.).** The fact that removing  $\langle\phi_d\rangle_{n-12}$  or  $\langle\phi_d\rangle_{n-24}$  from the inputs results in a marked decrease in performance further corroborates this statement by indicating that the wave packet acceleration is a significant factor in the decision making process (conversely, it was found that increasing the number of inputs by including measurements from earlier time steps did not improve the results).

Another aspect worth considering is how well the ANN performs when noise is added to its inputs and outputs. This is important because these types of effects are essentially unavoidable in a laboratory situation where, e.g., experimental measurements of the wave packet position will generally not reflect the actual position with 100% accuracy. To investigate, we modified the process outlined in section IV by adding a random uniformly distributed variable within a range of  $\pm 6^\circ$  to each  $\langle\phi_d\rangle_n$  “measurement”, and a similar random variable within a range of  $\pm 10\%$  of the peak field amplitude to each  $\varepsilon_n$  being output by the ANN. This modified model was then applied to  $10^5$  randomly perturbed test systems exactly as before, which yielded a mean target state occupation

of 0.87. This suggests that the ANN is able to robustly contend with moderate experimental noise.

We also tested the performance of our optimized ANN when encountering systems where the noise features in the perturbing functions had increased amplitudes compared to the original training data. We did this by applying the network to  $10^5$  new test systems, where the amplitude of the random features in  $\eta_V(\phi_d, t)$  and  $\eta_\alpha(\phi_d, t)$  had been increased by an average of 50%. This yielded a mean target state occupation of 0.82. Next, we tried to improve on this result by retraining the network on systems containing the larger amplitude perturbations. Despite repeated attempts using modified network topologies, GA parameters, and pulse time window lengths, we were not able to create a new network that could exceed the performance of the original network when faced with the new test sets.

It is encouraging to see that the original ANN is able to “handle” the larger amplitude perturbations moderately well, as this suggests a degree of flexibility with respect to how realistically the perturbations need to be constructed. However, the fact that an increase in perturbation amplitude leads to a seemingly uncorrectable decrease in performance suggests that there are some fundamental limitations associated with our current approach. Inspection of the systems where the network fails to perform well indicate that the problem arises when the wave packet does not make it over the central barrier in one piece, i.e. part of it is transferred and part remains in the left well. The resulting delocalization means that the position expectation value  $\langle\phi_d\rangle$  is no longer a good indicator of the actual position of the wave packet, which has a deleterious effect on the ANNs ability to move it into the right well.

## VI. FUTURE PERSPECTIVES

As stated, our motivation for suggesting this scheme is the possibility of developing a more efficient approach to coherent control. While the preliminary results outlined here indicate that our approach may be feasible, there are a number of issues and/or limitations that may need to be addressed before an experimental implementation is possible. For example, for our current method to work properly it is a requirement that we have access to some kind of information about the intermediate states of the system before the end of the pulse (whereas in a typical closed-loop approach the algorithm only “cares” about the terminal state). Depending on the experimental setup, these type of intermediate measurements may be difficult or impossible to obtain.

Another concern is how to properly implement the temporal step-by-step construction of the field in an experimental situation; ultrafast pulse shapers generally operate in the Fourier domain by manipulating the frequency components of the pulse spectrum, so constructing a field by precisely controlling the amplitude of the

temporal features as outlined in this paper may pose a challenge. Nevertheless, arbitrary pulse shape generation in the temporal domain has been demonstrated using pulse shapers that combine phase and amplitude manipulation in the spectral domain<sup>39</sup>. Another way around this problem could be to characterize the field as a train of Gaussian pulses generated with a beam splitter<sup>40</sup>, where the ANN could be used to determine the optimal intensity of each pulse.

A linchpin of this work in its present form is the assumption that discrepancies between the simulated quantum model and the real experimental dynamics can be rectified via the addition of one or more perturbing functions to the model Hamiltonian. While it is very unlikely that this assumption is *always* true, it is probably *sometimes* true. Short of a full experimental implementation, one could test when this assumption breaks down by increasing the number of degrees of freedom in the simulated model and checking whether or not the ANN is still able to effectively achieve the control objective. Finally, it would be interesting to see how well the approach outlined in this paper works for more challenging objectives such as, e.g., laser induced deracemization<sup>25</sup>.

Note that there are many other ways an ANN could be used to generate an optimized field based on feedback from a given molecular system. For example, the problem with delocalization might be mitigated by including information about the wave packet variance in the ANN inputs, or by modifying the procedure in a way that allows the ANN to also look “ahead” a few time steps as it constructs the field.

Another interesting possibility is related to the way an ANN might be used to “auto-parameterize” an optimization problem; as outlined in section I, properly parameterizing the search space for a coherent control experiment is not necessarily trivial. The results in this paper indicate that training the ANN allows it to automatically identify which pulse features are important (for example, in our model it appears that the critical parameters are related to the location of the rising and falling edges of the pulses, as exemplified by figure 7). In a sense this can be interpreted as a “hands-free” reduction of the search space dimensionality, which might be a worthwhile concept to further explore.

## VII. CONCLUDING REMARKS

We have proposed a method of experimental coherent control that is designed to make use of partial prior knowledge of a molecular system to arrive at a solution more quickly and/or efficiently than a standard closed-loop approach by reducing the required number of measurements. Our method is based on the application of a trained ANN in a manner that allows it to generate a controlling field in consecutive temporal steps based on dynamic experimental feedback from the molecular system.

Using a 1D model of the torsional motion in  $F_2H_3C_6 - C_6H_3Br_2$ , we have outlined an approach to modelling discrepancies between simulation and experiment by adding perturbing functions to the theoretical model Hamiltonian. We rationalized this treatment using the TDSCF approximation, and discussed the likely sources of error that will cause differences between the simulated and experimental dynamics. We suggested a method of generating random perturbing functions and argue that they will have a finite probability of reproducing the experimental dynamics when included in the model Hamiltonian.

Using neuroevolution, we optimized an ANN in a way that allows it to achieve robust quantum control of a simulated molecular system, despite the addition of the aforementioned random perturbations to the molecular potential energy and polarizability surfaces. We argued that this robustness will potentially allow the optimized ANN to achieve the same control objective in an experimental situation. We also demonstrated that the ANN can achieve the control objective using a number of measurements that is potentially multiple orders of magnitude smaller than a naive closed-loop approach would typically require to produce the same results.

In closing, the purpose of this paper is not to provide a definitive answer regarding the best way to implement a coherent control algorithm based on an ANN. Instead, it is to provide a tentative proof of concept for this novel idea that hopefully leads to lines of further inquiry.

## Appendix A: Modelling Realistic Perturbations

For clarity we will use a 1D example in the following description, however the results can easily be generalized to 2 or more dimensions. The goal is to generate a “noisy” signal where it is possible to control the amplitude of the noise fluctuations as well as the “smoothness” of the noise features (i.e. how correlated a given part of the signal is with its adjacent values).

We start by creating a discrete ordered sequence  $\nu(x)$  (where  $x = \mathbb{N}$ ), with statistically independent random values. Each value in the sequence is selected from a Gaussian probability distribution function (PDF):

$$P\{\nu(x) = z\} = \frac{1}{\sqrt{2\pi\sigma_\nu^2}} \exp\left(-\frac{z^2}{2\sigma_\nu^2}\right), \quad (A1)$$

i.e. for long sequences the mean value of  $\nu(x)$  will be  $\sim 0$ . This type of uncorrelated sequence or signal is often called “white” Gaussian noise because its power spectral density is constant at all frequencies. Next,  $\nu(x)$  is convoluted with a Gaussian low pass filter and multiplied by constant  $\beta$  to create the filtered and scaled sequence  $\eta(x)$ , i.e.:

$$\eta(x) = \frac{\beta}{\sqrt{2\pi\sigma_G^2}} \sum_{m=-\infty}^{\infty} \exp\left(-\frac{m^2}{2\sigma_G^2}\right) \nu(x - m). \quad (A2)$$

Varying the width of the Gaussian kernel  $\sigma_G$  allows us to control the smoothness of  $\eta(x)$ . The autocorrelation function can be used to obtain a quantifiable measure of this smoothness in terms of the characteristic size of the features in  $\eta(x)$ . For a signal generated using equations A1 and A2, it can be shown that the mean autocorrelation function of  $\eta(x)$  can be approximated by the following analytical expression:

$$\overline{R_\eta(l)} \approx \exp\left(-\frac{l^2}{4\sigma_G^2}\right). \quad (\text{A3})$$

Using equation A3, we can borrow a measure of the mean signal coherence length from turbulence theory in the form of the Eulerian integral macro-time scale, which is given by

$$L_\eta = \int_0^\infty \overline{R_\eta(l)} dl = \sigma_G \sqrt{\pi}. \quad (\text{A4})$$

Furthermore, it can be shown that the variance of the values in  $\eta(x)$  can be approximated by

$$\sigma_\eta^2 \approx \frac{\beta^2 \sigma_\nu^2}{2\sigma_G \sqrt{\pi}}. \quad (\text{A5})$$

Using equations A5 and A4, we can control the mean coherence length and/or amplitude of the features in  $\eta(x)$  by modifying  $\sigma_G$  and/or  $\beta$ .

## Appendix B: Genetic Algorithm Details

We wrote our own custom GA implementation, although the selection and cross breeding functions are identical to those used in the MATLAB<sup>41</sup> GA. As stated, we used a population of 300 individuals where the genomes were initialized with uniformly distributed values between  $-6$  and  $6$ .

When constructing a new generation, the two best performing individuals in the previous generation were included in the new generation unchanged. Of the remaining new individuals to be constructed, 80% were “children” created by selecting two “parents” from the current generation and cross breeding their genomes, and the remaining 20% were “mutants” created by selecting an individual from the current generation and mutating its genome.

Selection of  $P$  parents/mutants from a population containing  $N$  individuals is accomplished as follows:

1. Rank all  $N$  individuals according to their raw fitness scores.
2. Assign each individual a scaled fitness value proportional to its rank. The scaled fitness function employed here is  $F(R_n) = 1/\sqrt{R_n}$ , where  $R_n$  is the rank of the  $n^{\text{th}}$  individual.
3. Create a line of length  $L$  with  $N$  segments, where the length of the  $n^{\text{th}}$  segment is proportional to the scaled fitness of the  $n^{\text{th}}$  individual.

4. Starting from the beginning of the line, take a step of random length  $l_0$  along the line, where  $0 \leq l_0 \leq L/P$ . Select the individual that corresponds to this position on the line as the first selection of the  $P$  individuals that are to be selected.
5. Select the remaining  $P - 1$  individuals by moving along the line with equally spaced steps of length  $l$ , where  $l = L/P$ .

Cross breeding between parent  $A$  and  $B$  is accomplished by generating random binary vectors with lengths equal to the number of genes in the genome. The  $n^{\text{th}}$  gene in the child is then assigned the  $n^{\text{th}}$  gene from parent  $A$  ( $B$ ) when the  $n^{\text{th}}$  value in the vector is 0 (1).

Mutation is accomplished by adding a random Gaussian variable with a mean value of 0 and a standard deviation of  $\sigma$  to each gene. The size of  $\sigma$  used to construct the mutants in the  $n^{\text{th}}$  generation is adaptable (albeit with a maximum value of 0.6), and determined by the maximum fitness at generation  $n - 1$  and  $n - 2$ . If the maximum fitness has not improved between generation  $n - 2$  and  $n - 1$ , the current value of  $\sigma$  is updated by multiplying the previous value by 20. If the fitness between generation  $n - 2$  and  $n - 1$  has increased, the current value of  $\sigma$  is updated by dividing the previous value by 20. Finally, there is also a 1% chance that any gene in a genome that has been selected for mutation will be completely replaced with a new uniformly distributed random value between  $-6$  and  $6$ .

- <sup>1</sup>R. S. Judson and H. Rabitz, *Phys. Rev. Lett.* **68**, 1500 (1992).
- <sup>2</sup>S. A. Rice and M. Zhao, *Optical Control of Molecular Dynamics* (Wiley, New York, 2000).
- <sup>3</sup>M. Shapiro and P. Brumer, *Quantum Control of Molecular Processes* (Wiley, New York, 2012).
- <sup>4</sup>C. Brif, R. Chakrabarti, and H. Rabitz, *New J. Phys.* **12**, 075008 (2010).
- <sup>5</sup>A. K. Tiwari, K. B. Møller, and N. E. Henriksen, *Phys. Rev. A* **78**, 065402 (2008).
- <sup>6</sup>S. Sharma, H. Singh, and G. G. Balint-Kurti, *J. Chem. Phys.* **132**, 064108 (2010).
- <sup>7</sup>A. Assion, T. Baumbert, M. Bergt, T. Brixner, B. Kiefer, V. Seyfried, M. Strehle, and G. Gerber, *Science* **282**, 919 (1998).
- <sup>8</sup>T. C. Weinacht, J. Ahn, and P. H. Bucksbaum, *Nature* **397**, 233 (1999).
- <sup>9</sup>R. J. Levis, G. M. Menkir, and H. Rabitz, *Science* **292**, 709 (2001).
- <sup>10</sup>B. J. Pearson, J. L. White, T. C. Weinacht, and P. H. Bucksbaum, *Phys. Rev. A* **63**, 063412 (2001).
- <sup>11</sup>G. Vogt, G. Krampert, P. Nikaus, P. Nuernberger, and G. Gerber, *Phys. Rev. Lett.* **94**, 068305 (2005).
- <sup>12</sup>J. Savolainen, R. Fanciulli, N. Dijkhuizen, A. L. Moore, J. Hauer, T. Buckup, M. Motzkus, and J. L. Herek, *Proc. Natl. Acad. Sci. USA* **105**, 7641 (2008).
- <sup>13</sup>A. C. Florean, D. Cardoza, J. L. White, J. K. Lanyi, R. J. Sension, and P. H. Bucksbaum, *Proc. Natl. Acad. Sci. USA* **106**, 10896 (2009).
- <sup>14</sup>A. Monmayrant, S. Weber, and B. Chatel, *J. Phys. B* **43**, 103001 (2010).
- <sup>15</sup>M. Mitchell, *An Introduction to Genetic Algorithms*, 5th ed. (The MIT Press, 1999).
- <sup>16</sup>A. M. Weiner, *Rev. Sci. Instrum.* **71**, 1929 (2000).
- <sup>17</sup>R. B. Gerber, V. Buch, and M. A. Ratner, *J. Chem. Phys.* **77**, 3022 (1982).

- 1007 <sup>18</sup>J. Schmidhuber, *Neural Networks* **61**, 85 (2015).  
 1008 <sup>19</sup>R. Selle, G. Vogt, T. Brixner, G. Gerber, R. Metzler, and  
 1009 W. Kinzel, *Phys. Rev. A* **76**, 023810 (2007).  
 1010 <sup>20</sup>R. Selle, T. Brixner, T. Bayer, M. Wollenhaupt, and T. Baumert,  
 1011 *Journal of Physics B: Atomic, Molecular and Optical Physics* **41**,  
 1012 074019 (2008).  
 1013 <sup>21</sup>R. Kosloff, S. A. Rice, P. Gaspard, S. Tersigni, and D. J. Tannor,  
 1014 *Chem. Phys.* **139**, 201 (1989).  
 1015 <sup>22</sup>V. Engel, C. Meier, and D. J. Tannor, “Local control theory:  
 1016 Recent applications to energy and particle transfer processes in  
 1017 molecules,” in *Advances in Chemical Physics*, Vol. 141 (Wiley-  
 1018 Blackwell).  
 1019 <sup>23</sup>A. P. Peirce, M. A. Dahleh, and H. Rabitz, *Phys. Rev. A* **37**,  
 1020 4950 (1988).  
 1021 <sup>24</sup>C. B. Madsen, L. B. Madsen, S. S. Viftrup, M. P. Johansson,  
 1022 T. B. Poulsen, L. Holmegaard, V. Kumarappan, K. A. Jørgensen,  
 1023 and H. Stapelfeldt, *J. Chem. Phys.* **130**, 234310 (2009).  
 1024 <sup>25</sup>E. F. Thomas and N. E. Henriksen, *J. Phys. Chem. Lett.* **8**, 2212  
 1025 (2017).  
 1026 <sup>26</sup>H. Tanji, S. Minemoto, and H. Sakai, *Phys. Rev. A* **72**, 063401  
 1027 (2005).  
 1028 <sup>27</sup>L. Christensen, J. H. Nielsen, C. B. Brandt, C. B. Madsen, L. B.  
 1029 Madsen, C. S. Slater, A. Lauer, M. Brouard, M. P. Johansson,  
 1030 B. Shepperson, and H. Stapelfeldt, *Phys. Rev. Lett.* **113**, 073005  
 1031 (2014).  
 1032 <sup>28</sup>B. J. Sussman, *Am. J. Phys* **79**, 477 (2011).  
 1033 <sup>29</sup>E. F. Thomas and N. E. Henriksen, *J. Chem. Phys.* **144**, 244307  
 1034 (2016).  
 1035 <sup>30</sup>J. Haruyama, C. Hu, and K. Watanabe, *Phys. Rev. A* **85**, 062511  
 1036 (2012).  
 1037 <sup>31</sup>X. Yao, *Int. J. Intell. Syst. Tech. Appl.* **4**, 539 (1993).  
 1038 <sup>32</sup>D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Parallel*  
 1039 *Distributed Processing: Explorations in the Microstructure of*  
 1040 *Cognition, Vol. 1*, edited by D. E. Rumelhart, J. L. McClelland,  
 1041 and C. PDP Research Group (MIT Press, 1986) pp. 318–362.  
 1042 <sup>33</sup>I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT  
 1043 Press, 2016).  
 1044 <sup>34</sup>K. O. Stanley and R. Miikkulainen, *Evol. Comput.* **10**, 99 (2002).  
 1045 <sup>35</sup>R. Kosloff, *J. Phys. Chem.* **92**, 2087 (1988).  
 1046 <sup>36</sup>D. P. Mandic and J. Chambers, *Recurrent Neural Networks*  
 1047 *for Prediction: Learning Algorithms, Architectures and Stability*  
 1048 (John Wiley & Sons, Inc., New York, NY, USA, 2001).  
 1049 <sup>37</sup>C. C. Marston and G. G. BalintKurti, *J. Chem. Phys.* **91**, 3571  
 1050 (1989).  
 1051 <sup>38</sup>S. M. Hankin, D. M. Villeneuve, P. B. Corkum, and D. M.  
 1052 Rayner, *Phys. Rev. Lett.* **84**, 5082 (2000).  
 1053 <sup>39</sup>A. M. Wiener, J. P. Heritage, and E. M. Kirschner, *J. Opt. Soc.*  
 1054 *Am. B* **5**, 1563 (1988).  
 1055 <sup>40</sup>C. W. Siders, J. L. W. Siders, A. J. Taylor, S.-G. Park, and  
 1056 A. M. Weiner, *Appl. Opt.* **37**, 5302 (1998).  
 1057 <sup>41</sup>MATLAB, *version 9.0.0.341360 (R2016a)* (The MathWorks  
 1058 Inc., Natick, Massachusetts, 2016).