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

Esben F. Thomas and Niels E. Henriksen
Department of Chemistry, Technical University of Denmark, Building 206, DK-2800 Kongens Lyngby, Denmark

(Received 2 November 2018; published 19 February 2019)

We propose a method of experimental coherent control that exploits partial and/or 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 one-dimensional 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.

DOI: 10.1103/PhysRevA.99.023422

I. INTRODUCTION

The concept of applying ultrashort laser pulses to control the dynamics of molecular systems has been a topic of interest for some time. A large body of theoretical (see, e.g., Refs. [1–6]) and experimental (see, e.g., Refs. [7–13]) work has been produced in which the feasibility of applying custom-tailored laser pulses to drive various molecular systems into specific target states has been demonstrated.

For simple molecules, where it is possible to make accurate theoretical predictions, a so-called "open-loop" scheme may be employed, where the driving pulse shapes are designed based on knowledge of the system Hamiltonian. However, molecular systems are generally too complicated for this approach to be of much use. Alternatively, a so-called "closed-loop" scheme may be employed [1,14]. Essentially, the closed-loop approach is based on the application of a gradient-free optimization algorithm [15] to the inputs of a pulse shaper [14,16] in a feedback loop where the pulse shaper inputs are updated and optimized "on the fly" based on experimental data generated by the interaction of the molecular system with preceding pulses.

The efficacy of the closed-loop optimization scheme has been proven in a number of experiments, including ionization of gas phase diatomic sodium [10], photoisomerization of organic molecules [11], probing of chemical mechanisms via optimized pulse analysis [12], and manipulation of biological proteins [13]. One reason the closed-loop approach works well in experiments is that it requires little or no prior knowledge of the system it operates on; it is essentially a "black-box" approach to coherent control since the relationship between the problem inputs and outputs is unknown or does not need to be known.

The reality is, of course, that we generally have partial (but not complete) prior knowledge of any given molecular system and/or process that we wish to control. This naturally leads to the question of whether or not it is possible to devise a control scheme that makes use of this partial information in some way to arrive at a solution more quickly and/or efficiently. Such a scheme would be useful in situations where, e.g., it takes a long time to gather the experimental feedback data, since it would allow us to reduce the total number of measurements required to achieve a desired result.

There are numerous ways to implement this idea. One popular approach is to use a closed-loop scheme where the inputs to the optimization problem are parametrized based on prior knowledge of the system. For example, in Ref. [12] it is demonstrated that optimizing the phase of a transform-limited laser pulse with an evolutionary algorithm (EA) can lead to selective control over the branching ratio between two competing energy flow pathways in a bioinspired dyad molecule. The authors use two different strategies to achieve this; initially they perform a "blind" or "naive" optimization, i.e., they allow the EA to search for a solution without placing any restrictions on the way the phase function is constructed, leading to a search space containing 208 parameters. In the second strategy, they parametrize the phase function based on a qualitative analysis of the optimized pulse features from the initial (unconstrained) approach, resulting in a reduced search space containing 40 parameters. It is shown that parametrizing the phase function in this manner leads to significantly faster convergence; however, the optimized pulse doesn't perform as well as the pulse found using the unrestricted approach. This indicates that properly parametrizing the search space of the optimization algorithm can be a challenge, in particular for complicated systems or processes where it may be difficult to gain an intuitive understanding of the underlying control mechanism(s).

Our approach to implementing a control scheme that makes use of previously known information about a given system is based on the primary ansatz that any discrepancies between the theoretical description of an experiment and what happens

in reality can, in principle, be rectified by adding some kind of perturbative term(s) to the theoretical model (in Sec. III we will show that this is loosely analogous to assuming that the quantum molecular dynamics can be described using the time-dependent self-consistent field (TDSCF) approximation [17]).

Based on this premise, we demonstrate theoretically how an artificial neural network [18] (ANN) can be trained to achieve a desired control objective when applied to a theoretical molecular model with randomly perturbed potential and laser-molecule interaction functions. Furthermore, we suggest that an ANN that has been trained in this manner may be able to achieve the same control objective in a real experimental situation. Note that while ANNs have been used in the past to generate predictive models of ultrafast laser-molecule interactions [19,20], to our knowledge they have not been applied to coherent control experiments before.

While the results in this paper are theoretical, we substantiate the general experimental feasibility of our approach by demonstrating that the ANN only requires the measurements of experimentally observable quantities to be able to generate an optimized field that achieves the desired control objective. This is accomplished by allowing the ANN to construct the field directly in the temporal domain in consecutive steps, where the amplitude at each time step is based on measurements of the system at previous time steps. We also demonstrate that the number of required measurements is equal to the dimensionality of the optimization problem (i.e., the number of discrete temporal components that characterize the shape of the field), which is far fewer measurements than would typically be needed if we naively applied a closed-loop optimization scheme to the same problem.

At this point it is relevant to mention local control theory (LCT) [21,22], a qualitatively similar approach that allows for on-the-fly calculation of an electric field based on the dynamics of a theoretical model system at each time step in a way that leads to a monotonic increase (or decrease) in some predefined expectation value. A key difference between our method and LCT is that ours is intended for use on experimental, real-world systems as an alternative to the standard closed-loop approach, whereas LCT is generally used as a more efficient alternative to optimal control theory [23] (OCT) when working, e.g., with theoretical models that are very computationally expensive to simulate.

We would like to underline that the work presented in this paper is intended as a preliminary proof of concept for a general idea: that ANNs or other machine learning techniques can be used to increase the speed and efficiency of determining optimal pulse shapes in coherent control experiments. Consequently, we are not suggesting that the procedure presented here is the best, or only, way to implement this concept. Similarly, the model that we use to outline and theoretically verify our approach is only intended to provide a reasonably plausible example of how our idea might work in a real experiment. For this reason, although we attempt to demonstrate experimental feasibility when possible by citing relevant work, we do not exhaustively consider all of the technical challenges that might be associated with this particular setup.

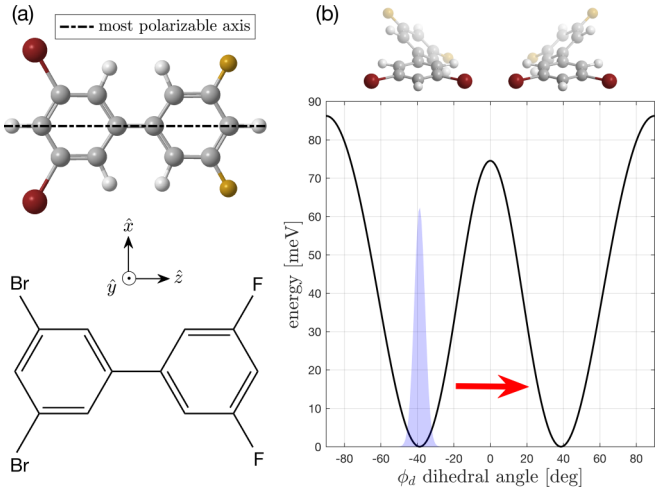

FIG. 1. (a) Molecular structure of the $F_2H_3C_6 - C_6H_3Br_2$ molecule. The most polarizable axis is also shown (dashed black line). In the simulations performed throughout this article, the MPA is always oriented along the laboratory frame $\hat{z}$ axis (b) potential-energy surface as a function of the torsional angle ϕ_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.

II. MODEL SYSTEM

The model we use is based on the torsional potential-energy surface of 3,5-difluoro-3′,5′-dibromobiphenyl (which we will henceforth refer to as $F_2H_3C_6 - C_6H_3Br_2$) in the electronic ground state (see Fig. 1). In Ref. [24] it is shown that the lowest vibrational mode of this molecule corresponds primarily to the torsional motion of the phenyl rings. Furthermore, it is shown that by aligning the most polarizable axis (MPA) of the molecule along the laboratory-frame $\hat{z}$ axis (see Fig. 1) and neglecting all higher frequency modes, the Hamiltonian of the system interacting with a nonresonant laser pulse polarized in the $\hat{x}\hat{y}$ plane can be approximated by (using atomic units)

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

where $\Phi = (\phi_{\mathrm{Br}}I_{\mathrm{Br}} + \phi_{\mathrm{F}}I_{\mathrm{F}})/(I_{\mathrm{Br}} + I_{\mathrm{F}})$ is the weighted azimuthal angle, ϕ_{Br} (ϕ_{F}) and I_{Br} (I_{F}) are, respectively, the rotational angle and inertial moment of the Br (F) substituted ring, $I = I_{\mathrm{Br}} + I_{\mathrm{F}}$ is the total moment of inertia for rotation around the stereogenic axis, $I_{\mathrm{rel}} = I_{\mathrm{Br}}I_{\mathrm{F}}/(I_{\mathrm{Br}} + I_{\mathrm{F}})$ is the relative moment of inertia, $\phi_d = \phi_{\mathrm{Br}} - \phi_{\mathrm{F}}$ is the relative torsional angle between the rings, $V_{\mathrm{tor}}(\phi_d)$ is the torsional potential energy, $\varepsilon(t)$ is the time-dependent electric field of the laser, and $\alpha(\Phi,\phi_d)$ is the molecular polarizability function (the exact forms of the $V_{\mathrm{tor}}(\phi_d)$ and $\alpha(\Phi,\phi_d)$ functions we use, as well as other model details, can be found in Ref. [25]).

Note that the first term on the right side of Eq. (1) describes the overall rotational kinetic energy of the molecule, and the second term describes the "internal" energy of the torsional oscillations. In addition to inducing torsional vibrations in

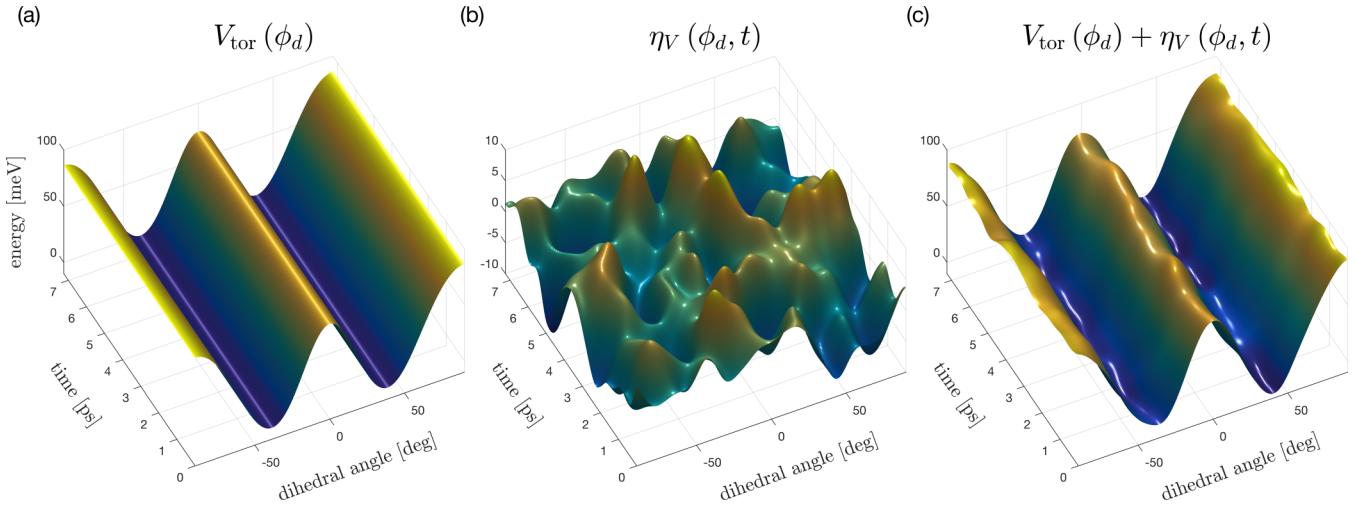

FIG. 2. (a) Theoretical torsional potential-energy function $V_{\rm tor}(\phi_d)$. (b) Example of the perturbing $\eta_V(\phi_d, t)$ function displaying the characteristic size scale of the perturbations in the temporal and spatial domains. (c) When $\eta_V(\phi_d, t)$ is added to $V_{\rm tor}(\phi_d)$,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

the ϕ_d coordinate, driving the system with a time-dependent field will generally lead to rotation in the Φ coordinate as the second most polarizable axis (SMPA; see, e.g., Fig. 2 in Ref. [24]) rotates to align with the field polarization axis. However, if we assume that the SMPA of the molecule is prealigned with the polarization direction of the driving field, there will be very little induced rotation in the Φ coordinate. This type of three-dimensional (3D) orientation and/or alignment of the MPA and SMPA molecular axes can be achieved using an elliptically polarized adiabatic alignment pulse [26,27]. In such a case, the Hamiltonian can be reduced to 1D by considering the dihedral motion at a fixed Φ coordinate, i.e.,

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

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

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

In Ref. [24] it is demonstrated that the description of the $F_2H_3C_6 - C_6H_3Br_2$ system given by Eq. (2) will yield provisionally accurate results based on comparisons with experimental data. However, it is clear that such a drastic simplification will generally not be able to accurately reproduce the real behavior of the system, particularly if the torsional oscillations become very large.

III. CORRECTING FOR DISCREPANCIES

The potential energy $V_{\rm tor}(\phi_d)$ and polarizability $\alpha(\phi_d)$ functions used in Eq. (2) are derived from a series of quantum-chemical calculations performed in Ref. [24], where the torsional angle of the central C − C bond was held fixed at various angles and the remaining structure was allowed to relax into the minimum-energy configuration before calculating the energy and polarizability.

We can identify at least two sources of error that are likely to cause discrepancies in the dynamic behavior of the experimental system compared to the simulated system: the first is simply due to unavoidable inaccuracies associated with any chosen method of quantum-chemical calculation and the second is due to the fact that other modes will undoubtedly become activated as the amplitude of the torsional oscillations become large, which will in turn lead to time-dependent distortions in the potential energy and polarizability surfaces that the simplified Hamiltonian in Eq. (2) does not account for. We will now briefly outline how the TDSCF approximation provides a framework allowing us to formally represent the influence of these other activated modes in the 1D Hamiltonian from Eq. (2).

Assuming a generalized molecular system is evolving on a single electronic potential surface within the Born-Oppenheimer approximation (in our case, this is the ground state), the TDSCF approximation assumes that the total system wave function containing N nuclear degrees of freedom can be written as a single Hartree product:

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

In Ref. [17], it is shown that this allows us to express the Hamiltonian of the ith mode as follows:

$$\hat{H}_i^{\rm TDSCF} = \hat{T}_i + V_i(x_i) + \overline{V}_i(x_i, t), \qquad (4)$$

where $\hat{T}_i$ and $V_i(x_i)$ represent the kinetic and potential energy, respectively, and where $\overline{V}_i(x_i, t)$ represents the time-dependent influence (i.e., energy exchange) from all other activated modes. Inspired by this formal treatment, we now modify the Hamiltonian from Eq. (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 $\alpha(\phi_d)$:

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

Now, the $\eta_V(\phi_d, t)$ term in Eq. (5) is equivalent to $\overline{V}_i(x_i, t)$ in Eq. (4); however, the addition of $\eta_\alpha(\phi_d, t)$ to the polarizability function requires further justification. First of all, the form of Eq. (5) implicitly assumes that the experimental field-molecule interaction is still dominated by the molecular polarizability term, which is reasonable provided the experimental laser pulse remains in the nonresonant regime and does not become too intense. Furthermore, it can be shown that the energy shifts caused by the polarizability interaction can be expressed as an expansion of the molecular dipole moment onto the unperturbed (i.e., field-free) electronic eigenfunctions [28,29]. It is reasonable to assume that these electronic eigenfunctions will be modified when 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 Eq. (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 behavior of the experimental wave packet with perfect accuracy if we insert them into the Hamiltonian in Eq. (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 nonzero probability that any randomly generated set of $\eta_V(\phi_d, t)$ and $\eta_\alpha(\phi_d, t)$ functions will reproduce the behavior 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 Eq. (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 as follows.

(i) 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.

(ii) 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 ϕ_d.

(iii) 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 and/or 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 and/or 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° in the ϕ_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 ϕ_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 Eq. (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 Sec. 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 Sec. III, the goal is to teach a computer to achieve the control task described in Sec. II on the system represented by Eq. (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) [15] using a technique called neuroevolution [31].

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"

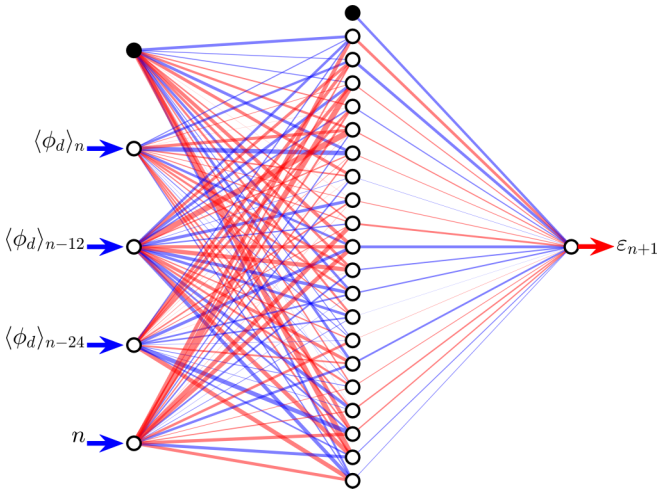

FIG. 3. Sketch of the optimized feedforward network with four inputs and a single hidden layer with 20 nodes. The blue (dark gray) and red (light gray) lines respectively indicate positive and negative connection weights, and the line thicknesses correspond to the absolute weight magnitude (for reference, 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, labeled n, inputs the current time step. All inputs are preprocessed by scaling them to a range between approximately -1 and 1, and the network output is scaled to a value between zero and the (user-defined) laser intensity cutoff limit.

that receives a series of node inputs and generates a node output based on their weighted sum, i.e.,

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

where p_i represents the "raw" value from the ith incoming connection, w_i is the corresponding connection weight, w_b represents the contribution from a constant "bias" input (see, e.g., Fig. 3), and K is the activation function that maps $\sum_i w_i p_i$ to scalar node output f. Each node input p_i comes either from other nodes within the network, or from "outside" the network as part of the network input vector. Likewise, each node output can connect to other nodes within the network and/or to the network output vector. Evaluation of a given network input by the ANN is achieved by propagating the "signal" from the network input vector through the network nodes until it reaches the network output vector.

An ANN can "learn" generalized relationships between the inputs and outputs associated with a given problem or task. This is accomplished by optimizing all the internal connection weights until the ANN consistently produces the "correct" output for any relevant input. In many cases, the connection weights can be optimized by gradient descent using backpropagation [33]; however, this method requires access to a training set of valid input-output pairs. Neuroevolution avoids this issue by using a GA to optimize the network connection weights instead, where each candidate network receives a fitness score based on how successful it is at performing a desired behavior or task. This makes neuroevolution particularly useful for reinforcement learning problems where the correct network outputs for any given set of network inputs may not be known. Note that more advanced neuroevolution algorithms will evolve both the topology of the network and its weights [34]; however, we will not be making use of this approach.

We will now describe our operational approach to using an ANN to generate a field based on dynamic feedback from an arbitrarily perturbed $F_2H_3C_6 - C_6H_3Br_2$ system, and we will demonstrate how we used neuroevolution to optimize the ANN connection weights. The temporal pulse envelope is characterized by a series of N discrete, equally spaced regions or "bins" with width δt and total length $N \times \delta t = T$. The ANN assigns the amplitude of the bin at time step $n+1$ with a constant value based on information about the system behavior from time steps zero to n. In practice, the pulse time window was set to $T = 7.25$ ps, and the number of field components was set to $N = 300$, i.e., each field component had a width of ~24 fs, significantly shorter than the ~1200 fs vibrational period of the system. As we will demonstrate in Sec. V, this step size is small enough to allow the network to tailor fields that perform well on specific sets of perturbations.

We now outline the general procedure. First, let ε_n and $\langle\phi_d\rangle_n$ denote the respective field amplitude and position expectation value at the nth time step, and let $\mathbf{M}_n$ denote the list of positions between time step zero and n, i.e.,

$$\mathbf{M}_n = \{\langle\phi_d\rangle_0, \langle\phi_d\rangle_1, \ldots, \langle\phi_d\rangle_{n-1}, \langle\phi_d\rangle_n\}. \tag{7}$$

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

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

Next, ε_{n+1} is appended to the total field shape, the system wave packet is propagated forward from time step n to time step $n+1$ using split-operator propagation [35], 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 Fig. 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 Fig. 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.,

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

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.

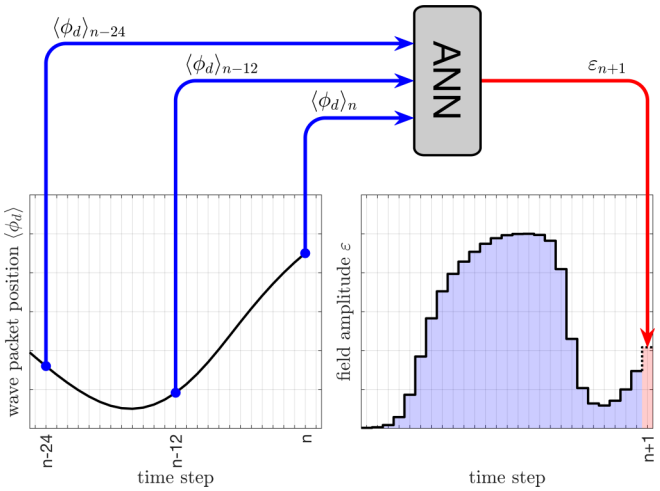

FIG. 4. Schematic showing how information about the system from time steps zero to n is used by the ANN to determine the field amplitude at time step $n+1$. The left and right panels respectively show the wave-packet position $\langle\phi_d\rangle$ and the field amplitude ε at an (arbitrary) interval between time steps $n-15$ and $n+1$. As stated in the article text, once the ANN determines the amplitude of ε_{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.

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 [see, e.g., Eq. (6)] can significantly impact the quality of the results. We found that a simple feedforward configuration [32] 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 Fig. 3 (note that since the ANN uses information from previous time steps it might formally be classified as a *recurrent* neural network [36]; 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 data points going "back in time" from the most recent measurement, equally spaced at interval K, i.e.,

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

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 Fig. 3.

We will now explain how we used a GA to optimize the network connection weights in order to achieve the stated control objective. Note that the feedforward network we wish to train contains a total of 121 connection weights (see Fig. 3). This means that we will essentially be using the GA to solve a 121 parameter optimization problem, i.e., each candidate GA solution is represented by a "genome" consisting of 121 double-precision floating point numbers. Each number in a given genome defines a unique network connection weight within the predefined network topology. For this reason, a given genome can be used to generate its corresponding "phenotype" by mapping its values to the weights of a network, and conversely the "genotype" of a given network can be extracted by mapping its connection weights to the corresponding genome positions. In the following outlined steps it should therefore be understood that the terms "genome" and "network" essentially mean the same thing, and will be used interchangeably depending on context.

(1) Generate S random "training" systems consisting 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 [37] (FGH) method.

(3) For each training system, define a set of 10 target states, $\overline{\chi_s} = [\chi_{s,0}, \chi_{s,1}, \ldots, \chi_{s,9}]$, 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 mth 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, $\Psi_{m,s}(\phi_d, t = T)$.

(6) Assign the mth network a fitness score F_m, defined as the mean overlap of all $\Psi_{m,s}(\phi_d, t = T)$ from step (5) with the target states in $\overline{\chi_s}$, i.e.,

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

(7) Repeat steps (5) and (6) for all M networks, and use the GA to create a population of new genomes by mutating and cross breeding genomes from the current generation with higher fitness scores (see Appendix B for details about our GA implementation).

(8) Repeat steps (5)–(7) until the maximum fitness level of the population converges and/or ceases to significantly improve.

By evaluating the performance of the networks on the same training systems in every generation, we ensure that the convergence is monotonic (this would not be the case if we, e.g., created a new set of training systems for each new generation). The caveat of this approach is that we must include a set of training systems that is large enough to prevent overfitting; i.e., if we use too few training systems, then it is unlikely that a network will be able to learn the general rules

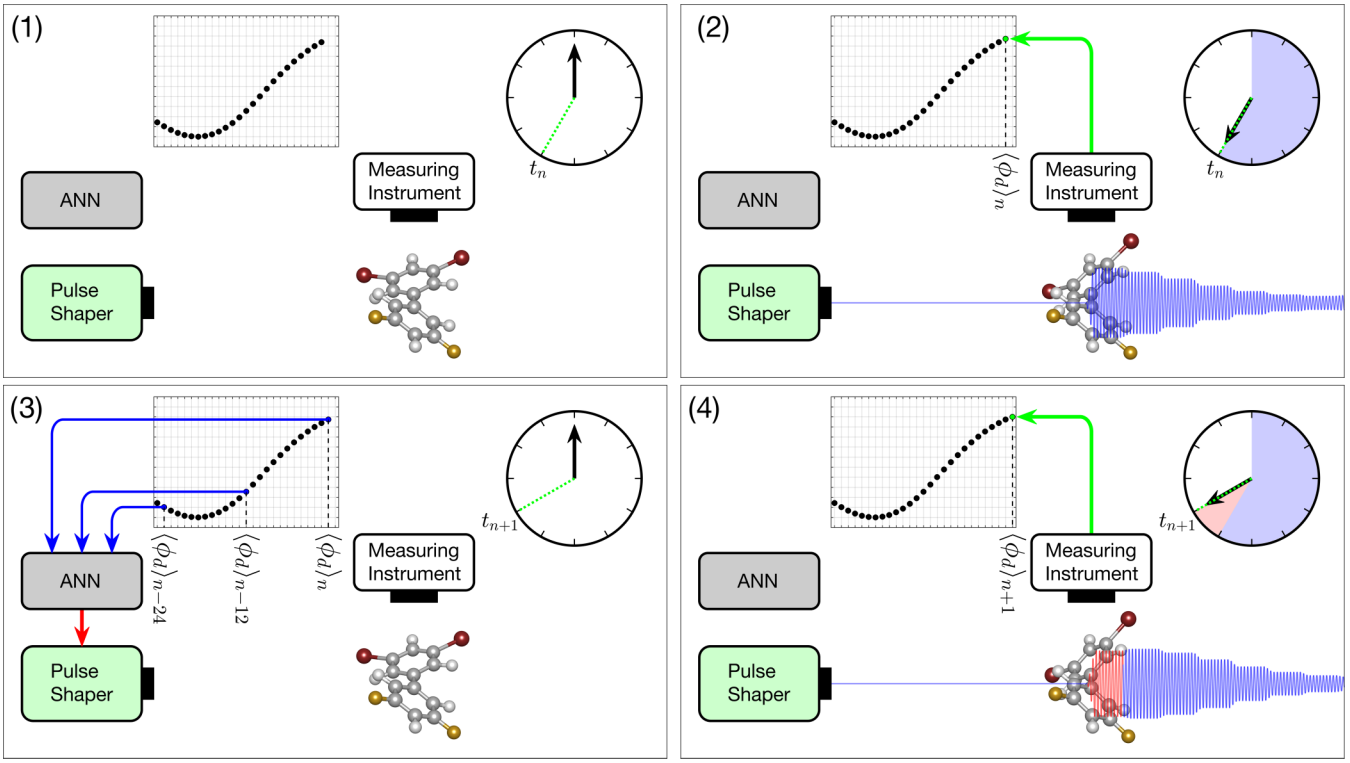

FIG. 5. Illustration of how the trained ANN can be implemented to construct an optimized field in an experimental situation. Note that this figure is meant to show a single intermediate iteration at the nth step of a process that has already been repeated $n-1$ times beforehand. (1) At time step n the 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 zero 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) and (4) are repeated until N time steps have passed.

it needs to know to be able to successfully tackle a system that isn't part of the training set.

In practice, the appropriate number of training systems was estimated through trial and error by cross-validating the performance of the converged network on a series of 10^5 random new systems generated using the same noise parameters as the training set (i.e., the same mean temporal and/or angular coherence lengths and amplitude variances as outlined in Sec. III). We found that using a training set containing $S = 100$ different systems yielded very similar training and cross-validation scores, indicating that this is a reasonable size (another common strategy for avoiding overfitting is to monitor the cross-validation error at every iteration and halt the optimization once this value begins to increase; however, as our initial approach seems to work well enough we did not find it necessary to try other methods).

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

V. RESULTS AND DISCUSSION

Using the methodology outlined in Sec. IV we optimized a network using a GA population of $M = 300$. The network outputs were scaled to a value between zero and a peak pulse intensity of 20 TW/cm^2 (note that experimental evidence suggests that this intensity will not ionize the molecules [27,38]; nevertheless eventual ionization issues may be remedied by increasing the length of the pulse and decreasing the allowable peak intensity).

Figure 3 shows a sketch of this optimized ANN. As stated at the end of Sec. IV, we cross-validated the performance of this network by applying it to 10^5 new random systems generated using the same noise parameters as the training sets, and calculated the resulting overlap of the propagated wave packets with the target wave functions. The red histogram in Fig. 6 shows the distribution of the target occupation levels for all 10^5 cross-validation measurements. It is clear that the network is quite effective at achieving the control objective when dealing with perturbed systems that it hasn't encountered before, as the distribution is strongly peaked with a mean value of 0.95.

Now, it is possible that we have created a network that simply produces generalized pulse shapes that work well across

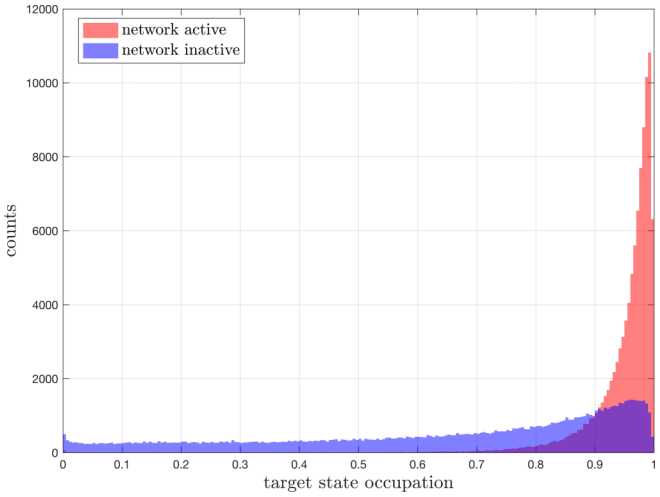

FIG. 6. Strongly peaked red histogram shows the distribution of target state occupations when the optimized network is applied to 10^5 random test systems (note that these systems are not included in the set that was used to train the network). The flatter blue histogram shows the corresponding distribution when the 10^5 pulses generated by the network in the aforementioned analysis are used to drive different systems 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.)

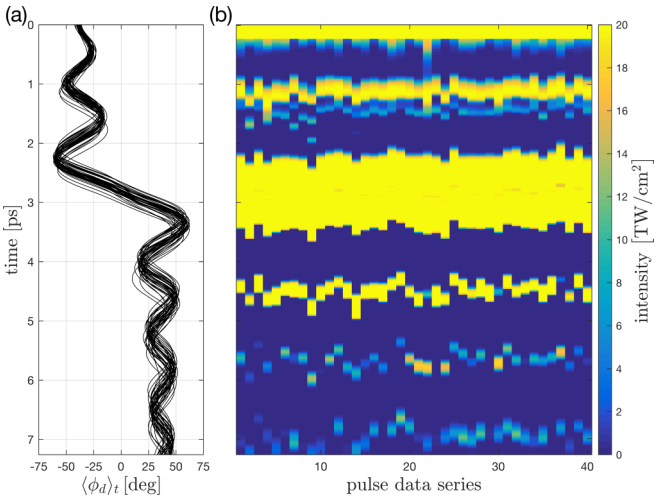

FIG. 7. (a) Overlaid time-dependent position expectation values of wave-packet trajectories generated by applying the optimized ANN from Fig. 3 to 40 different test systems being perturbed by different $\eta_V(\phi_d, t)$ and $\eta_\alpha(\phi_d, t)$ functions created using the parameters outlined in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

all perturbed systems, i.e., the ANN may not actually be making "intelligent" decisions based on the immediate behavior of the wave packet in any given system (in particular, this would be true if the perturbation amplitudes were too small to have a significant impact on the wave-packet dynamics). To test this hypothesis, the 10^5 pulses that the network generated in the previous cross-validation analysis were again applied to the 10^5 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 Fig. 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 behavior 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 10^5 pulses to the unperturbed potential, yielding a mean target state occupation of 0.87. Finally, we used the network to generate 10^5 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 systems 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 Fig. 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, Fig. 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 Fig. 7(b) (or on the right side of Fig. 4) indicate that the rising and falling edges of the pulses are generally "smooth," i.e., each edge consists of multiple intermediate time steps. 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 Sec. I, is equal to the number of free parameters used to characterize the shape of the field. A naive 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 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 Fig. 7 suggest that it would be relatively easy to reparametrize the search space and significantly reduce the dimensionality of the optimization problem, e.g., as discussed in Sec. I. For this reason, a properly parametrized 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 Sec. I, manually determining a suitable parametrization 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 parametrizes the problem for us.

While the complexity of the connections in Fig. 3 makes 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 Sec. 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 Fig. 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 or 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 Sec. 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 ε_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 [39]. Another way around this problem could be to characterize the field as a train of Gaussian pulses generated with a beam splitter [40], 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 [25].

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 "autoparametrize" an optimization problem; as outlined in Sec. I, properly parametrizing 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 Fig. 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 modeling 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 idea that hopefully leads to lines of further inquiry.

APPENDIX A: MODELING REALISTIC PERTURBATIONS

For clarity we will use a 1D example in the following description; however, the results can easily be generalized to two 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):

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

i.e., for long sequences the mean value of $\nu(x)$ will be ~ 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 β to create the filtered and scaled sequence $\eta(x)$, i.e.,

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

Varying the width of the Gaussian kernel σ_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 Eqs. (A1) and (A2), it can be shown that the mean autocorrelation function of $\eta(x)$ can be approximated by the following analytical expression:

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

Using Eq. (A3), we can borrow a measure of the mean signal coherence length from turbulence theory in the form of the Eulerian integral macrotime scale, which is given by

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

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

$$\sigma_\eta^2 \approx \frac{\beta^2\sigma_v^2}{2\sigma_G\sqrt{\pi}}. \tag{A5}$$

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

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 [41] 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 or 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 nth individual.

(3) Create a line of length L with N segments, where the length of the nth segment is proportional to the scaled fitness of the nth individual.

(4) Starting from the beginning of the line, take a step of random length l_0 along the line, where $0 \leqslant l_0 \leqslant L/P$. 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 nth gene in the child is then assigned the nth gene from parent A (B) when the nth value in the vector is zero (1).

Mutation is accomplished by adding a random Gaussian variable with a mean value of zero and a standard deviation of σ to each gene. The size of σ used to construct the mutants in the nth 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 σ 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 σ 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.

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

[2] S. A. Rice and M. Zhao, *Optical Control of Molecular Dynamics* (Wiley, New York, 2000).

[3] M. Shapiro and P. Brumer, *Quantum Control of Molecular Processes* (Wiley, New York, 2012).

[4] C. Brif, R. Chakrabarti, and H. Rabitz, New J. Phys. **12**, 075008 (2010).

[5] A. K. Tiwari, K. B. Møller, and N. E. Henriksen, Phys. Rev. A **78**, 065402 (2008).

[6] S. Sharma, H. Singh, and G. G. Balint-Kurti, J. Chem. Phys. **132**, 064108 (2010).

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

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

[9] R. J. Levis, G. M. Menkir, and H. Rabitz, Science **292**, 709 (2001).

[10] B. J. Pearson, J. L. White, T. C. Weinacht, and P. H. Bucksbaum, Phys. Rev. A **63**, 063412 (2001).

[11] G. Vogt, G. Krampert, P. Niklaus, P. Nuernberger, and G. Gerber, Phys. Rev. Lett. **94**, 068305 (2005).

[12] 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).

[13] 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).

[14] A. Monmayrant, S. Weber, and B. Chatel, J. Phys. B **43**, 103001 (2010).

[15] M. Mitchell, *An Introduction to Genetic Algorithms*, 5th ed. (MIT Press, Cambridge, MA, 1999).

[16] A. M. Weiner, Rev. Sci. Instrum. **71**, 1929 (2000).

[17] R. B. Gerber, V. Buch, and M. A. Ratner, J. Chem. Phys. **77**, 3022 (1982).

[18] J. Schmidhuber, Neural Networks **61**, 85 (2015).

[19] R. Selle, G. Vogt, T. Brixner, G. Gerber, R. Metzler, and W. Kinzel, Phys. Rev. A **76**, 023810 (2007).

[20] R. Selle, T. Brixner, T. Bayer, M. Wollenhaupt, and T. Baumert, J. Phys. B: At., Mol., Opt. Phys. **41**, 074019 (2008).

[21] R. Kosloff, S. A. Rice, P. Gaspard, S. Tersigni, and D. J. Tannor, Chem. Phys. **139**, 201 (1989).
[22] V. Engel, C. Meier, and D. J. Tannor, Local control theory: Recent applications to energy and particle transfer processes in molecules, in *Advances in Chemical Physics* (Wiley-Blackwell, New York, 2009), Vol. 141.
[23] A. P. Peirce, M. A. Dahleh, and H. Rabitz, Phys. Rev. A **37**, 4950 (1988).
[24] C. B. Madsen, L. B. Madsen, S. S. Viftrup, M. P. Johansson, T. B. Poulsen, L. Holmegaard, V. Kumarappan, K. A. Jørgensen, and H. Stapelfeldt, J. Chem. Phys. **130**, 234310 (2009).
[25] E. F. Thomas and N. E. Henriksen, J. Phys. Chem. Lett. **8**, 2212 (2017).
[26] H. Tanji, S. Minemoto, and H. Sakai, Phys. Rev. A **72**, 063401 (2005).
[27] L. Christensen, J. H. Nielsen, C. B. Brandt, C. B. Madsen, L. B. Madsen, C. S. Slater, A. Lauer, M. Brouard, M. P. Johansson, B. Shepperson, and H. Stapelfeldt, Phys. Rev. Lett. **113**, 073005 (2014).
[28] B. J. Sussman, Am. J. Phys. **79**, 477 (2011).
[29] E. F. Thomas and N. E. Henriksen, J. Chem. Phys. **144**, 244307 (2016).
[30] J. Haruyama, C. Hu, and K. Watanabe, Phys. Rev. A **85**, 062511 (2012).
[31] X. Yao, Int. J. Intell. Syst. Tech. Appl. **4**, 539 (1993).
[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, edited by D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group (MIT Press, Cambridge, MA, 1986), pp. 318–362.
[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).
[34] K. O. Stanley and R. Miikkulainen, Evol. Comput. **10**, 99 (2002).
[35] R. Kosloff, J. Phys. Chem. **92**, 2087 (1988).
[36] D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (John Wiley & Sons, Inc., New York, 2001).
[37] C. C. Marston and G. G. Balint-Kurti, J. Chem. Phys. **91**, 3571 (1989).
[38] S. M. Hankin, D. M. Villeneuve, P. B. Corkum, and D. M. Rayner, Phys. Rev. Lett. **84**, 5082 (2000).
[39] A. M. Wiener, J. P. Heritage, and E. M. Kirschner, J. Opt. Soc. Am. B **5**, 1563 (1988).
[40] C. W. Siders, J. L. W. Siders, A. J. Taylor, S.-G. Park, and A. M. Weiner, Appl. Opt. **37**, 5302 (1998).
[41] MATLAB, *version 9.0.0.341360 (R2016a)* (The MathWorks Inc., Natick, MA, 2016).